HAL
open science

Using the blockchain technology for trust improvement of processes in Logistics and Transportation

Adnan Imeri

To cite this version:

Adnan Imeri. Using the blockchain technology for trust improvement of processes in Logistics and Transportation. Data Structures and Algorithms [cs.DS]. Université Paris-Saclay; Université du Luxembourg, 2021. English. NNT : 2021UPASG036 . tel-03859728

HAL Id: tel-03859728

https://theses.hal.science/tel-03859728v1

Submitted on 18 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Utilisation de la technologie blockchain l'amélioration de la confiance dans les processus de logistique et de transport

Using the Blockchain Technology for Trust Improvement of processes in Logistics and Transportation

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Réseaux, information et communications
Unité de recherche : Université Paris-Saclay, Univ Evry, IBISC, 91020, Evry-Courcouronnes, France
Référent : Université d'Evry Val d'Essonne

Thèse présentée et soutenue à Evry, France le 02 Juillet 2021, par

Adnan IMERI

Composition du jury

Hendrik Proper Professeur associé, Université du Luxembourg	Président
Amel Bouzeghoub Professeure, Telecom SudParis (Institut Mines-Télécom)	Rapportrice et Examinatrice
Rafael Tolosana-Calasanz Professeur, Universidad de Zaragoza	Rapporteur et Examinateur
Agnès Lanusse Ingénieure-chercheure, CEA LIST	Examinatrice

Direction de la thèse

Nazim Agoulmine Professeur, Université Paris-Saclay (UEVE)	Directeur de these
Djamel Khadraoui Directeur de recherche, Université du Luxembourg	Codirecteur de these

Membres

Stephane Maag Professeur, Telecom SudParis (Institut Mines-Télécom)	Membre invité
Dimitri Konstantas Professeur, Université de Genève	Membre invité

Thèse de doctorat

NNT : 2021UPASG036

Abstract

This thesis address the general problem of safe and secure transport of dangerous goods (TDG). The TDG is very complicated to manage because of risk for the environment and human life. Currently, it suffers from a lack of efficiency, trust, and transparency. In this thesis, we propose a novel method to specify the workflow aspects of TDG by considering all TDG process stages during its entire lifecycle. This method aims to facilitate the specifications of the TDG workflow management system that is entirely based on existing regulatory frameworks ensuring the compliance, trust, and transparency of all underlying processes. The proposed system design method is based on the so-called model-driven architecture (MDA) approach and enhancing it to consider blockchain properties. The first stage is the formal analysis of the process of TDG and its alignment with the regulatory frameworks. The proposed design method aims, at this stage, to allow the formal definition and verification of the design of the system with regard to the regulatory frameworks. The next stages of the method rely strongly on the model transformation that is a salient aspect of the proposed design method. Model transformation allows to automatically discover peer system components and authorized interactions. The last stage of the whole model transformations is the specification of digital twin profiles for all potential stakeholders. All the interactions in the real world between stakeholders are transformed into interactions in the digital world, while the interactions with the environment are achieved through the use of IoT. The proposed approach enables interactions between components of the systems (digital twins, IoT devices, etc.) only if this is compliant with the regulatory framework. Thanks to blockchain technology, our design method allows improving trust and transparency in the process of TDG from the perspective of stakeholder collaborations. Smart contract technological capabilities are also a cornerstone of the proposed solution. This thesis also contributes to improving the semantic of smart contracts to capture supply chain management specifications as well as dangerous goods specificities in terms of transportation. Dynamic concepts related to the supply chain management of dangerous goods such as time-related and geographic constraints, digital certification, anomaly detection and multi-party smart contract, managing emergencies, and shared responsibility have been addressed at the level of the smart contract. In particular, this thesis proposes applying temporal logic for the formal specification and verification of smart contracts. This thesis proposes an integrated approach for blockchain and IoT to support the dynamic aspects in the supply chain of dangerous goods. Data collected from various IoT devices along the physical supply chain (goods, vehicles, country borders, etc.) are transmitted to the blockchain and further processed by the system following the workflow logic that was specified and automatically triggering related smart contracts and corresponding actions. The last contribution in this thesis is the implementation of a proof-of-concept system to validate the different aspects of the contribution, namely the design method, the trust and transparency assurance, and the automatic triggering of actions and information flows.

Key Words

Blockchain, Logistics, Trust, Workflow, IoT, Smart Contract

Acknowledgements

First of foremost, I would like to express my deep thankfulness to my supervisors, Professor Dr. Nazim Agoulmine and Professor Dr. Djamel Khadraoui, for their supervision, support, collaboration, and encouragement to fulfill this thesis. On any other occasion, this thesis would not be at the same level without their comprehensive supervision.

On this opportunity, I am deeply thankful to the president of the jury of my thesis, Professor Dr. Hendrik Proper.

I want to express my profound gratefulness to the thesis reviewers, Professor Dr. Amel Bouzeghoub and Professor Dr. Rafael Tolosana-Calasanz, for the review, examination, and suggestions to improve my thesis.

Following this, I deeply thank the thesis examiner, Dr. Agnès Lanusse, for the thesis examination and comments and suggestions for future works on top of this thesis.

My sincere thanks to Professor Dr. Stephane Maag and Professor Dr. Dimitri Konstants for agreeing to examine my thesis and attending the thesis defense.

I want to express my gratefulness to my colleagues at the Luxembourg Institute of Science and Technology (LIST) for collaborating with me to achieve excellent thesis results. Also, sincere acknowledgment to LIST for their support in my thesis journey.

Last but not least, my ultimate gratitude to my family for their genuine support shown to me during my entire thesis journey.

Contents

Abstract **iii**

Acknowledgements **iv**

List of Figures **xiv**

List of Tables **xviii**

List of Abbreviations **xx**

1 Introduction **1**

1.1 Supply Chain Management, Logistics and Transportation 1

1.2 The Supply Chain of Dangerous Goods (DG) 3

1.3 Context of the Study and General Research Challenges 3

1.4 Thesis Outline 4

1.5 List of Publications 5

2 Transport and Management of Dangerous Goods (TMDG) **9**

2.1 Introduction 9

2.2 Dangerous Goods (DG) 9

2.2.1 TDG Main Stakeholders, Safety Procedures and Restriction 10

2.2.2 Consequences from Misuse of DG 12

2.2.3 Process of Transport of Dangerous Goods (TDG) 13

2.2.4 Regulatory Framework for TDG 14

2.2.5 Main Components to Consider for TDG 14

2.2.6 Exemption of DG 16

2.2.7 DG Across European Union (EU) 16

2.3 TDG from Scientific and Industry Perspective 16

2.3.1 Decision Support Systems (DSS) as a Management Tool for TDG . . . 17

2.4 Conclusion 21

3 Technologies: Centralized, Distributed, and Decentralized **23**

3.1 Introduction 23

3.2 Centralized: Client-Server Architectures 23

3.3 Distributed Architecture 25
3.3.1 Distributed Cloud Computing Architecture 25
3.3.2 Internet of Things 26

3.4 Distributed Ledger and Blockchain Technology 28
3.4.1 Blockchain Network, Data Structure, and Mining Process 30
3.4.1.1 Incentives: Operations costs in Blockchain Technology 34
3.4.2 Consensus Algorithms 34
3.4.2.1 The Byzantine Fault Tolerance (BFT) 35
3.4.3 Overview: The Current Blockchain Platforms 39
3.4.4 Current Challenges for Blockchain 45
3.4.5 Comparison of Blockchain Platforms 47

3.5 Smart Contract (SC) 48
3.5.1 Smart Contract Definition, Semantics and the Main Characteristics . . 48

3.6 Synthesis: Comparison of the Technological Features for Centralized, Distributed, and Distributed-Decentralized Approaches 50

3.7 Advanced Concepts: Digital Twins 52

3.8 Conclusion 52

4 State-of-the-art: Blockchain Involvement in Supply Chain and Logistics, and Blockchain-based System Design Approaches **55**

4.1 Introduction 55

4.2 Blockchain and Supply Chain Management (SCM) 56
4.2.1 The Value of Blockchain in Supply Chain Management and Logistics . 56
4.2.2 Blockchain and Supply Chain Management 57
4.2.3 Blockchain for Solving Trust Issues in Supply Chain Management . . . 58
4.2.4 Business Process Management (BPM) and Blockchain Integration . . . 61
4.2.5 The Traceability of Goods in Supply Chain Management and Logistics 62
4.2.6 Blockchain Based Startups for Logistics and Supply Chain Management 65
4.2.7 Challenges in Integrating Blockchain in Supply Chain and Logistics . 68

4.3 Integration of Blockchain and IoT 69

4.4 Smart Contract: Related Works Studies 71
4.4.1 Blockchain and Smart Contract Design 72

4.4.2 Role Based Access Control: Related Work Studies 72
4.4.3 Smart Contract Formal Specification and Verification 73
4.4.3.1 The Vulnerabilities of Smart Contract 73
4.4.3.2 Overview of Model Checking and Verification Techniques . . 74
4.4.3.3 Formal Methods 74
4.4.3.4 The Scientific Approach for Model Checking and Verification of Smart Contract 75

4.5 Synthesis: Beyond the Current State-of-the-art 77

4.6 Conclusion 80

5 Context of Study, Problem Statement and Scientific Objectives 81

5.1 Introduction 81

5.2 The Current TDG Organizational Aspects 81
5.2.1 Today's Enhanced Process Organization in TDG 83

5.3 Transportation and Management of Medical Waste (TMMW) 83
5.3.1 Toxic and Infectious Substances 84
5.3.2 The Use Case of the Transport and Management of Medical Waste (TMMW) 84
5.3.3 Regulatory Framework Applied for the Transport and Management of Waste 85
5.3.4 Description of the TMMW Process 87
5.3.5 Workflow Analysis for the TMMW Process 87
5.3.6 An End-to-End Example for TMMW 90

5.4 Background: Definition of the Used Concepts and Terminology 92
5.4.1 Information Security. Secure Information Sharing 92
5.4.2 Trust 92
5.4.3 Transparency 94
5.4.4 Safety 94
5.4.5 Completeness and End-To-End Concept 95

5.5 TDG Challenges and Research Objectives 96
5.5.1 Efficiency Issues 96
5.5.2 Trust Issues and Information Security 96
5.5.2.1 Security Issues with IoT 97
5.5.3 Traceability Issues 98
5.5.4 Interoperability Issues 98
5.5.5 Problem Statement and Scientific Objectives 98
5.5.6 Identified Challenges 98

5.6 Conclusion 99

6 Scientific Approach for Designing the Blockchain-Based System 101

6.1 Introduction 101

6.2 Design Method: Blockchain Based System Model (DG-BCSM) 102
6.2.1 Model-Driven Engineering (MDE) 102
6.2.2 Model-Driven Architecture (MDA) 104
6.2.2.1 MDA Models 104
6.2.2.2 The Benefits of MDA 105
6.2.3 MDA and Model Transformation 105

6.3 Common Information Model for the TDG (CIM-TDG) 107
6.3.1 CIM Overview 107
6.3.2 CIM-TDG Schema and CIM-TDG Profile 108
6.3.3 CIM-TDG General Schema 109
6.3.3.1 CIM-TDG Transport 109
6.3.3.2 CIM-TDG Logistics 110
6.3.3.3 CIM-TDG Stakeholders 112
6.3.3.4 CIM-TDG Activities 114
6.3.3.5 CIM-TDG Dangerous Goods 114
6.3.4 CIM Profile for CIM-TDG Schema 114
6.3.4.1 CIM-TDG Schema Extension 118

6.4 Conclusion 122

7 Platform-Independent Meta-Model, Verification Aspects and User Model 123

7.1 Introduction 123

7.2 L1: Platform-Independent Meta-Model (PIMM) 124
7.2.1 Maintainability: Traceability and Adaptability of Changes in the Regulatory Framework 130

7.3 L2: Platform-Independent Model (PIM): The Definition of BPMN 131
7.3.1 Knowledge Extraction for Composing the BPMN Models 132
7.3.1.1 BPMN: Certification of Stakeholders 132
7.3.1.2 BPMN: Certification for Waste Trader (Broker) 132
7.3.1.3 BPMN: Notification of Authorization or Rejection 134
7.3.1.4 BPMN: The Process of Transport (Movement of DG) 136
7.3.1.5 BPMN: The Process after Transport of the MW 138

7.4 Transformation from PIM Smart Contract Model (PISCM) to PS Smart Contract Model (PSSCM) 140
7.4.1 BPMN to UML Sequence Diagram (USD) Model Transformation . . . 140

7.5 L3: Platform-Independent Smart Contract Model (PISCM) 142

7.5.1 PISCM: Smart Contract Specification Model for the Process Before TMMW 143
7.5.2 PISCM: During Transportation Smart Contract Specification Model . . 149
7.5.3 PISCM: After-transportation Smart Contract Specification Model . . . 152
7.5.4 Managing Time Constraints at the Design Level 154
7.5.5 Role-Based Access Control 155

7.6 L4: Platform Independent System Architecture (PISA) 159
7.6.1 PISA: Deployment Architecture 160

7.7 L5: Platform Specific Model (PSM) 162

7.8 L6: Platform Specific Smart Contract Model (PSSCM) 164

7.9 Agility and Openness of our Design Method 165

7.10 Tools Used for Developing our Design Method 165

7.11 Conclusion 166

8 Smart Contract Improvement in Terms of Semantics to Capture Specification of Supply Chain Management for Dangerous Goods 167

8.1 Introduction 167

8.2 Specification of Smart Contract to Capture the Semantics of SCM for DG . . . 168
8.2.1 Meta-Rules from Regulatory Framework 168

8.3 Time Constraint Smart Contract 170
8.3.1 Applying Time Constraints on the Blockchain and Smart Contract Level 173
8.3.1.1 Ethereum-based Design 173
8.3.1.2 Shared Time-based Variable for Time Management and Constraint 174

8.4 Blockchain and IoT Integrated Approach for a Trusted and Secured TDG Process 178
8.4.1 3-Layers Conceptual Architecture 178

8.5 Geographic Constraints 181

8.6 Digital Certificate for Traceability Management of DG 182

8.7 Managing Emergency Situations 186

8.8 Anomaly Detection in TDG via Blockchain and Smart Contract 187

8.9 Shared Responsibility 189

8.10 Multi-Party Smart Contract and Business Contract Management 189

8.11 Dynamic Smart Contract for Permissioned Blockchain 193

8.11.1 Problem Definition: The Issues of Maintaining the Immutability of the Smart Contract . . . 193
8.11.2 Use Case of Temperature Checking. The Issues of Smart Contract Maintainability in a Dynamic Environment . . . 195
8.11.3 Dynamic Smart Contract for Permissioned Blockchain Based on Dynamic Parameterization . . . 195
8.11.4 Dynamic Parameterization . . . 195
8.11.5 State Machine Representation . . . 196

8.12 Formal Specification of Smart Contract for capturing the semantics in SCM for TDG . . . 197
8.12.1 SC Formalization in Linear Temporal Logic (LTL) . . . 197
8.12.1.1 Formal Specification of Business Contracts Applied in Smart Contract . . . 198
8.12.1.2 Formal Specification of Smart Contract for Governing TDG . 199
8.12.1.3 Time Constraints Specification . . . 203
8.12.1.4 Formal Specification of Geographic Constraint at Smart Contract Level . . . 203
8.12.1.5 Constraint-based Path Determination . . . 204
8.12.2 Model Checking of Smart Contract . . . 206
8.12.2.1 Model Checking . . . 206

8.13 Conclusion . . . 207

9 The Proof of Concept (PoC) Implementation for TDG-control System 209

9.1 Introduction . . . 209

9.2 The TDG-control System Architecture and Related Components . . . 209
9.2.1 Blockchain Environment Configuration . . . 210
9.2.1.1 Network and Channels . . . 211
9.2.2 SC Development and Deployment . . . 213
9.2.3 Template: Generation of Digital Twins Based on the Design Method . 217
9.2.4 The Improvement of Trust and Transparency in SpCDG . . . 219
9.2.5 User Interface and Scenario Implementation . . . 219
9.2.6 Efficiency . . . 226
9.2.7 Interoperability . . . 226
9.2.8 Exploring and Monitoring the Blockchain Components . . . 227
9.2.9 Blockchain and IoT Integration . . . 227
9.2.9.1 The used IoT Devices . . . 228
9.2.10 The Temperature Checking Use Case Implementation . . . 229

9.3 Conclusion . . . 232

10 Conclusions, Perspectives, and Future Research Directions **233**

10.1 Perspectives: Deployment of TDG Blockchain Solution at International Level 235

10.2 Future Research Directions 236

A Appendix: Additional Chapter Components **239**

A.1 Systematic research methods for the composition of the state-of-the-art 239

A.2 Difference between CIM-MDA and CIM-DMTF 240

A.3 Algorithm: Legal Text to BPMN 241

A.4 Presentation of DG Official Signs Based on ADR 241

A.5 Modeling Aspects: Defining the Basic Terminology 243
A.5.1 Unified Modeling Language (UML) 244
A.5.2 Business Process Model and Notation (BPMN) 245
A.5.2.1 Business Process Management (BPM) 245
A.5.2.2 Modeling Language: Business Process Model and Notation (BPMN) 246

A.6 Ontology Web Language (OWL) 247
A.6.1 The OWL classes of CIM Profile for CIM-TDG Schema 248

A.7 Technical Components of Blockchain 250
A.7.1 Hashes 250
A.7.2 Public Key Infrastructure (PKI) 251
A.7.3 Addresses 251
A.7.4 Transaction 252
A.7.5 Wallets 252
A.7.6 Mining Process Based on *Proof of Work* Consensus Algorithm 252

A.8 First-Order and Temporal Logic 253
A.8.1 First-Order Logic (FOL) 253
A.8.2 High Level Logic: Temporal Logic 253
A.8.2.1 Computational Temporal logic 254

A.9 The DT in TDG 257

A.10 Smart Contract Code Examples 259

A.11 Notification and Movement Documents 264

List of Figures

1.1 The SpC network and the interaction schema between stakeholders (Witthaut et al., 2017). 2

2.1 The main stakeholders (entities) involved in the process of TDG, based on (UNECE, 2017). 13
2.2 An example for packing, marking and labeling the DG. 16
2.3 The statistics for total transport of DG in EU and EEA countries. Data source: Eurostat (Eurostat, 2019). 17

3.1 The architecture of the centralized client-server system computing approach. 24
3.2 The high level architecture of cloud computing (Hoy, 2012). 26
3.3 The overview of IoT system architecture. 27
3.4 The blockchain technology components and main characteristics. 33
3.5 The Corda blockchain ledger representation. 41
3.6 The overview of the key concepts of HF. 43
3.7 The schema for DL Tangle. Inspired from (IOTABlog, 2018). 45

5.1 The example of DG transport map in a cross-border context. 82
5.2 The classic "client-server" approach for the TDG process management (Imeri and Khadraoui, 2018). 83
5.3 The interaction and references of legal documents for TMMW. 86
5.4 The UML sequence diagram of different processes involved in TMMW. 89
5.5 The end-to-end concepts for TMDG. 95

6.1 The Blockchain Based System Model (DG-BCSM) design method referential schema. 103
6.2 The MDA approach for the software development lifecycle. 105
6.3 The CIM-TDG general schema for the process of TDG. It presents a global referential CIM model for TDG. 110
6.4 The CIM-TDG Schema for "Transport" and its related components. 111
6.5 The CIM-TDG schema for the "Logistics" and its related components. 111
6.6 The CIM-TDG schema for the "Stakeholders" involved in the process of TDG. This schema may be extended with other involved stakeholders depending on DG specificity and regulatory framework. 112
6.7 The CIM-TDG schema for the business activities performed by "Stakeholders" in relation with TDG. 113

6.8 The CIM-TDG schema for the "Dangerous Goods" component. 113
6.9 The main classes in the OWL model representing the ontology of TDG. . . . 116
6.10 The main axioms used in the TDG and particularly for the use case of TMMW. 117
6.11 The TDG-mission expressed with the help of axioms. 119
6.12 The extended CIM-TDG schema for the "Transport" and its related components. 120
6.13 The changes in the CIM-TDG schema for "Transport" reflected in the OWL model. 121

7.1 The domain-specific meta-model is based on the regulatory framework concepts. 128
7.2 The representation of the concrete syntax for the meta-model (abstract syntax). 129
7.3 An example of a throwing error when the connection between components is not compatible with the meta-model. 130
7.4 The meta-model to support the maintainability and adaptability of changes in regulatory framework into SC. 131
7.5 The flow chart diagram for mapping the BPMN model from legal documents (text). 133
7.6 The broker certification (trader) process. 135
7.7 The Waste Collector (or Transporter) certification process. 135
7.8 The process of notification (authorization) for the transport and management of MW at the international level. 137
7.9 An end-to-end BPMN model for the MW (DG) transport process. 139
7.10 The FED for the stakeholder registration and validation. 144
7.11 The FED for the TMMW (TDG) authorization (notification) process. 147
7.12 The FED for the process during the TMMW (TDG). 151
7.13 The FED for the process after the TMMW (TDG). 153
7.14 The diagram for presenting the relationship between category (role)-treatment (activity) and transactions. 157
7.15 The role-based access control policy (ACP) and the notion of "guard" smart contract. 158
7.16 The platform-independent system architecture for the TDG. 161
7.17 The deployment architecture for the independent system architecture for TDG. 161
7.18 The platform-specific model and related components for the blockchain-based system. 163

8.1 The SC schema for updating the value of alpha (α) based on different events captured with the help of SC. 173
8.2 The conceptual representation of the time-constrain in SC. 174
8.3 The concept of SC for time management at run time level. The conceptual schema propose the time management by using on-chain blockchain capabilities. 176
8.4 The conceptual architecture for blockchain and IoT integration to support the TDG. 179
8.5 The concept of Digital Certificate for digital traceability and management of DG. 184

8.6 The concept of multi-party SC in TDG process. 190
8.7 The schema for presenting the static referential SC addresses issues. 194
8.8 The problem of cross-references for smart contract. 195
8.9 The state machine schema for the dynamic SC for the permissioned blockchain approach. 196
8.10 The FSM model for formal specification of SC. 200
8.11 The NuSMV model-checking for specification of path selection. 206

9.1 The TDG-control system architecture. 210
9.2 The access of blockchain network via web user interface. 212
9.3 The structure of sub-networks (channels) in HF for TDG. 212
9.4 The example of private data collection (PDC) shared in HF channel. 215
9.5 The results for the execution of SC guard access. 216
9.6 The conceptual presentation of real concept (stakeholder) into digital twin. . . 218
9.7 The authorized operation for the DT of "DG Provider" and "Authority". 218
9.8 The user interface for stakeholder registration. 221
9.9 The example of expressing meta-rules for TDG. 221
9.10 The user interface for accessing the TDG control system. 222
9.11 The web form for general information to request authorization to TDG. 223
9.12 The identification of cross-border points for TDG. 224
9.13 The convoy and crew details for the TDG. 224
9.14 The user interface for "authority". 225
9.15 The process (request) identification for TDG. 225
9.16 The exploration of BC components used in the PoC. 226
9.17 The interaction of TDG-control system with stakeholder application (DSS). . . 227
9.18 The exploration blocks and transaction performed by stakeholders (Organizations). 227
9.19 The conceptual model for illustrating the implementation of temperature checking use case. 230
9.20 The web form for supporting the TDG monitoring (Process ID: 97e9da08). . . 231

A.1 The DG official labels according to DG classes. 242
A.2 The example of UML class and relationships. 245
A.3 The basic BPMN Symbols. 246
A.4 The main classes in the OWL model representing the ontology of TDG. 249
A.5 Unwind State Graph to obtain Infinite Tree (Clarke, 1999). 254
A.6 The example of DT in TDG. 258
A.7 The example of SC code expressed in JavaScript. 260
A.8 The simplified SC for stakeholder registration. 261
A.9 The HF commands for creation of channels. 273

List of Tables

2.1 Classification of DG according to ADR. 11
2.2 International regulatory framework and agreements for TDG, specific to the mode of transport. 14
2.3 The laws, directives, and regulations for TDG used in the local and international context. 15
2.4 The summary of scientific approaches and research projects for the TDG. . . . 18
2.5 TDG from the market perspective. 20

3.1 The summary of main characteristics of the blockchain platforms. 47
3.2 Comparison of technological features: centralized, distributed and decentralized. 51

4.1 The results of the survey presented by Deloitte (Pawczuk et al., 2018). 57
4.2 The summary of industrial projects for blockchain in SCM. 67
4.3 Summary of the main approaches related to modeling and verification of SC. 76

5.1 Laws, directives and regulations for transport and management of Waste, MW and DG. 85

7.1 The summary of concepts, relationships and business rules based on the regulatory framework. 126
7.2 The transformation rules (semantics) for BPMN to the UML Sequence Diagram (USD). 142
7.3 Definition of roles, categories and the access-control policy for the TDG. . . . 158
7.4 The transformation rules for FED schema to platform independent system architecture. 160
7.5 The transformation rules from PISCM into code generation. 165

List of Abbreviations

SpC	Supply Chain
SCM	Supply Chain Management
DG	Dangerous Goods
SpCDG	Supply Chain of Dangerous Goods
BC	Blockchain
SC	Smart Contract
CS	Client-Server
CP	Cloud Computing
IoT	Internet of Things
HF	Hyperledger Fabric
ETH	Ethereum
DT	Digital Twin
TDG	Transport of Dangerous Goods
TMDG	Transport and Management of Dangerous Goods
ADR	European Agreement concerning the International Carriage of DG by Road
TMMW	Transport and Management of Medical Waste
MW	Medical Waste
MDE	Model-Driven Engineering
MDA	Model-Driven Architecture
CIM	Common Information Model
OWL	Ontology Web Language
PIMM	Platform-Independent Meta-Model
BP	Business Process

BPM	Business Process Management (BPM)
PIM	Platform-Independent Model
MT	Model Transformation
PISCM	Platform-Independent Smart Contract Model
PISA	Platform Independent System Architecture
PSM	Platform Specific Model
PSSCM	Platform Specific Smart Contract Model

To my parents Sofie and Ahmet,

To my wife and my daughters,

To my whole family,

To my friends,

I dedicate this thesis.

This thesis proposes a novel method and tools to improve the specification of workflow in the process of transport and management of dangerous goods. The proposed method aims to facilitate the management of workflow process and the trustful and secure sharing of information between collaborating stakeholders taking benefit from advanced technologies such as Blockchain, Smart Contracts and the Internet of Things.

Cette thèse propose une méthode et des outils inédits pour améliorer la spécification du workflow dans le processus de transport et de gestion des marchandises dangereuses. La méthode proposée vise à faciliter la gestion du processus de flux de travail et le partage fiable et sécurisé d'informations entre les parties prenantes en tirant parti des technologies avancées telles que Blockchain, Smart Contracts et l'Internet des objets.

Chapter 1

Introduction

The globalization era turned the Supply Chain into a dynamic and distributed environment, with strong dependencies on communication between stakeholders for the purpose of work synchronization. The growth of the markets and the on-demand requests for the goods within developed countries, economies in transition, and the cross-border requests intense the work of the transportation process. The transportation process presents one of the significant challenges to manage due to its intensity and several stakeholders involved in this process.

The development of society and the economy, the large scale of the people, and the different needs for humans to operate here expand the need for developing different goods and substances that are used daily. These goods and substances might expose a high risk for humans and for the environment too. The management of dangerous goods, hazardous and non-hazardous waste raises many challenges for the developed countries and the countries under development.

The emerged challenges will be mitigated and the transport of dangerous goods process well managed with the help of technologies. The technological aspects optimize the way business processes are organized by enhancing efficiency, reducing human involvement, and improving work quality. For overcoming such challenges and improving the Supply Chain of dangerous goods, we researched new possibilities for designing new methods, models and identifying the technological features to improve the workflows in the process of transport and management of dangerous goods.

This chapter presents information about the thesis context of the study and general research challenges. It includes general information about Supply Chain (SpC), Logistics, and Transportation. Further, we present the SpC for dangerous goods as well as the thesis outline.

1.1 Supply Chain Management, Logistics and Transportation

The Supply Chain (SpC) is a complex and massive network of stakeholders composed of "Suppliers", "Manufacturers", "Retailers" and "Logistics", which perform different activities to produce, distribute and manage specific product for the customers (Witthaut et al., 2017; Werner, 2008; Lin et al., 2005). That is possible only with the continuous cooperation between stakeholders. Figure 1.1, illustrates the involvement of stakeholder in SpC. The "Suppliers" are responsible for providing raw materials for the "Manufacturers". The "Manufacturers"

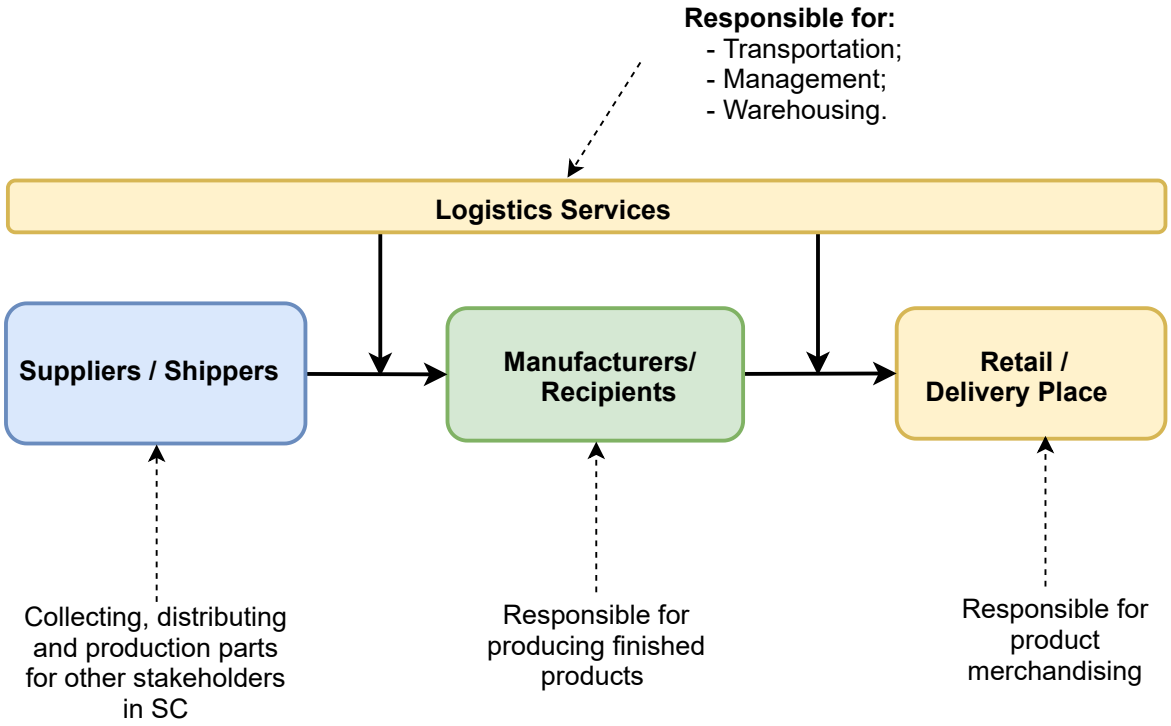

FIGURE 1.1: The SpC network and the interaction schema between stakeholders (Witthaut et al., 2017).

use these materials to produce their finished products (goods). The goods are exposed to the markets through "Retailers". In the SpC network, the "Logistics" covers almost any interaction between stakeholders. It comprises planning, transportation, and management (warehousing) of goods.

The flow of goods, information, and financial transactions need to be appropriately managed to comply with business contracts, legal policies, and national and international standards (Christopher, 2011). For managing SpC activities, the general term "Supply Chain Management (SCM)" is introduced in the literature. The term SCM is defined by research in (Mentzer et al., 2001) as "the systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purposes of improving the long-term performance of the individual companies and the supply chain as a whole". The research from (Lambert, 2008), defines SCM as "the integration of key business processes from end-user through original suppliers, that provides products, services, and information that add value for customers and other stakeholders". The SCM includes planning and managing the flow of goods, services, and financial transactions (Chandra and Grabis, 2007).

Due to the distributed nature of SCM and the fact that the marketplace is growing continuously, logistics and transportation are subject to disturbances (Witthaut et al., 2017). The large number of communications, the centralized storage of information, lack of process auditing, lack of long-life management of goods, traffic congestion during the transportation of goods, and other possible obstacles, are current SCM challenges. Furthermore, a SCM considered concerns are data security, trust among suppliers, and information asymmetry

(on the trading markets).

The logistics services are SCM component and mainly focus on efficient goods transportation of goods (Chandra and Grabis, 2007; Christopher, 2011). The logistic process organization requires strict processing and coordination of tasks and the tolerance of errors tending to zero. Subsequently, these are considered the conditions that should be fulfilled to achieve stakeholders and customer satisfaction. Any delay or error in logistic task processing will impact stakeholders and customer satisfaction. There are many challenges in workflow management in SCM and logistics for business enterprises. For sustainable logistic services, the synchronization and optimization of processes are highly required. For that, logistics stakeholders need to share certain information for process coordination. That constitutes several challenges in sharing information with stakeholders in the business field. This information may contain some business details toward goods capacity, source, destination, current warehousing, and the timestamp of goods and final destination movement.

As one of the most dynamic SCM components, logistics presents fundamental workflow organization challenges during the continuous collaboration among stakeholders and cost reduction. (Márquez, 2010) shows the SCM challenges toward stakeholders collaboration, thus highlighting trust issues, then, in the organization of the transportation of goods. Also, significant SCM challenges are considered transparency over SCM and logistics, traceability of goods in SCM, network flow optimization (non-efficiency), and many more.

1.2 The Supply Chain of Dangerous Goods (DG)

The Supply Chain of DG (SpCDG) consists on a complex network of stakeholders involved in managing and transporting dangerous goods (DG). The involved stakeholders should comply with a wide range of regulatory frameworks and other transport and process management specifications. The process retains a certain level of specification on goods treatments, storage, transport, and processing. For the involved stakeholders, a wide range of requirements such as qualified staff, particular transport vehicles, and mandatory safety procedures to follow. SpCDG eventually differs from the general SpC. In the context of DG, stakeholders must be certified by competent authorities in order to be able to operate in SpCDG. The transportation and processing (treatment) of DG must be a priori granted from the relevant competent authority. In SpCDG, stakeholders that provide DG are obliged to monitor the transport of DG and the lifecycle of DG, and finally, receive a certificate of treatment for that DG. Chapter 2 presents an extensive study for the transport of dangerous goods (TDG).

1.3 Context of the Study and General Research Challenges

The context of the study is the general problem of safe and secure transportation of dangerous goods (TDG), such as gases, explosives, nuclear materials, waste, medical waste, pharmaceutic products, fuel, acids, etc., that are treated and used by a public (or private) enterprise, and in some cases, by the military. The study also intends to cover the dual-use (civil-military) of the DG. In the context of TDG, many stakeholders cooperate to ensure the

successful fulfillment of the transport process. This process is very delicate because of the risks for the environment and human life. The process reaches a certain level of complication due to the many standards and strict local and international policies that govern it. There are distinguished constraints that apply during each stage of the process (before, during, and after), which entirely characterize the process of TDG. During this process, many sensitive information needs to be securely shared among the stakeholders. Examples are contractual terms, types of goods to be transported, the timestamp of movement of goods, warehousing, transportation mode, and many others. The current way the process of TDG is operating suffers from the lack of efficiency due to manual and administrative works required for this process, the issues of cooperation of stakeholders due to lack of trust and transparency over the process of TDG, and potential external audit. This enormously limits the workflow process in TDG and increases its implementation cost.

Several challenges emerged in the context of this study in terms of setting up a **trustable** process for the TDG. The first of these challenges is **"Maintaining safety and security"**, which intends to keep these goods secure and under the authorities' surveillance and to avoid cases where such goods are used for other purposes, e.g., benefiting from selling them to unauthorized parties or being stolen and used for destructive (harmful) purposes. The other challenges are **"Managing compliance with strict and specific regulations (national and international level, e.g., ADR, ADN, and many others, described in section 2.2.4)"** and **"Increasing traceability and transparency along the supply chain of TDG, including in the cross-border context"**. This would help to increase **transparency** and **trust** in the movement of DG and improve the cooperation of stakeholders operating at the national and international level. From a business perspective, it is necessary to **"secure information sharing between all the stakeholders (actors) (businesses, security responders, citizens, government authorities)"**, in order to have a reliable end-to-end system, which enables a secured and trustable environment. The **"Crisis management in the event of damage related to DG"** presents a specific challenge due to the high risk for people, the environment, business, and even politics. In the event of a crisis, **fast dissemination of information to the authorities and first responders is necessary** with a high degree of accuracy of information about the goods and the context.

1.4 Thesis Outline

The outline of this thesis is as follows:

Chapter 2 shows an extensive study concerning the Transport of Dangerous Goods (TDG). It presents the definition of Dangerous Goods (DG), the main stakeholders involved in the TDG process, and DG classification based on their substantial specificity and regulatory framework involvement in the TDG. Further, we show a study over the current scientific and industrial research.

Chapter 3 aims to synthesize the existing standardized technologies, including centralized approach, distributed (cloud computing and IoT), and distributed-decentralized ledgers

(blockchain) used to support business processes. Furthermore, we present a technological features comparison of these technologies.

Chapter 4 is characterized by state-of-the-art studies of blockchain and SCM, the integration of blockchain and IoT, designing smart contracts, and formal specification and verification of the smart contract. In this chapter, we present a synthesis and show our research objectives beyond the current state-of-the-art.

Chapter 5 presents the context of the thesis, the use case and its current organization, and the use case workflow analysis. Furthermore, it introduces the general problem statements and highlights the specific scientific objectives.

Chapter 6 presents our proposed scientific method to design a blockchain-based TDG management system. Also, it introduces the common information model for the TDG (CIM-TDG), which allows us to express a knowledge representation model.

Chapter 7 outlines six layers of our design method L1 to L6. The L1 layer defines the platform-independent meta-model for the TDG, representing the static aspects of the Computational Independent Model (CIM*). It maps the business rules into a meta-model for the TDG. The L2 layer presents the dynamic aspect of the platform-independent model (PIM). Furthermore, the L3 layer is mainly characterized by model transformation. We define a platform-independent system architecture in L4, which collects and digitally presents the targeted system component functionalities and their interaction with systems. We describe the technology-related and code-generation model in layers L5 and L6, respectively.

Chapter 8 present some advanced concepts in the SCM of DG supported with the help of SC. We specify dynamic aspects for obtaining the semantics and improving the process of TDG. The semantics captured concerning TDG are implicitly associated with the SC. Furthermore, we formally illustrate the SC semantics in terms of formal specification and verification of business rules for the TDG with the help of temporal logic.

Chapter 9 shows details of implementing our design method in a Proof of Concepts (PoC). We present the technical architecture, implementation schema, user interface and applications to interact with the blockchain. Furthermore, we present the technical components of a blockchain and IoT integration.

Chapter 10 outlines some discussions, the potential remarks, future research perspectives, and research areas that were opened by this thesis.

1.5 List of Publications

The research work activities for this thesis have produced several publications. These publications are distributed over thesis chapters, and they compose an indispensable part of this thesis. The list of publications is as follows:

1. Imeri, Adnan, Abdelaziz Khadraoui, and Djamel Khadraoui. **"A Conceptual and Technical Approach for Transportation of Dangerous Goods in Compliance with**

Regulatory Framework". In: Journal of Software 12.9, p. 14. ISSN: 1796-217X. DOI: 10.1016/j.ssci.2016.09.008. (2017). (Imeri et al., 2017).

2. Imeri, Adnan and Djamel Khadraoui. **"The Security and Traceability of Shared Information in the Process of Transportation of Dangerous Goods"**. In: 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS). Paris: IEEE, pp. 1–5. ISBN: 978-1-5386-3662-6. DOI: 10.1109/NTMS.2018.8328751. (2018). (Imeri and Khadraoui, 2018).

3. Imeri, Adnan, Christophe Feltus, Djamel Khadraoui, Nazim Agoulmine, and Damien Nicolas. **"Solving the trust issues in the process of transportation of dangerous goods by using blockchain technology"**. In: Proceedings of the 11th International Conference on Security of Information and Networks. Cardiff, United Kingdom: ACM Press, pp. 1–2. ISBN: 978-1-4503-6608-3. DOI:10.1145/3264437.3264470. (2018). (Imeri et al., 2018).

4. Imeri, Adnan, Djamel Khadraoui, and Nazim Agoulmine. **"Blockchain Technology for the Improvement of SCM and Logistics Services: A Survey"**. In: Industrial Engineering in the Big Data Era. Springer International Publishing, pp. 349–361. ISBN: 978-3-030-03317-0. (2019). (Imeri et al., 2019b).

5. Imeri, Adnan, Nazim Agoulmine, Christophe Feltus, and Djamel Khadraoui. **"Block chain: Analysis of the New Technological Components as Opportunity to Solve the Trust Issues in Supply Chain Management"**. In: Intelligent Computing. Vol. 998. Series Title: Advances in Intelligent Systems and Computing. Springer International Publishing, pp. 474–493. ISBN: 978-3-030-22867-5 978-3-030-22868-2. DOI: 10.17706/jsw.12.9.708-721. (2019). (Imeri et al., 2019c).

6. Imeri, Adnan, Nazim Agoulmine, and Djamel Khadraoui. **"A secure and smart environment for the transportation of dangerous goods by using Blockchain and IoT devices"**. In: 7th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2019). Cape Verde, pp. 1–8. (2019). (Imeri et al., 2019a).

7. Imeri, Adnan, Jonathan Lamont, Nazim Agoulmine, and Djamel Khadraoui. **"Model of dynamic smart contract for permissioned blockchains"**. In: Practice of Enterprise Modelling Conference Forum (PoEM 2019 Forum). Luxembroug, pp. 1–16. (2019). (Imeri et al., 2019e).

8. Imeri, Adnan, Nazim Agoulmine, and Djamel Khadraoui. **"Smart Contract modeling and verification techniques: A survey"**. In: 8th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2020), pp. 1-8. (2020). (Imeri et al., 2020a).

9. Imeri, Adnan, Nazim Agoulmine, and Djamel Khadraoui. **"Blockchain and IoT integrated approach for a trusted and secured process to manage the transportation of dangerous goods"**. In: Computing and System Journal, v. 10, n. 3, p. 17. (2020). (Imeri et al., 2020b).

10. Imeri, Adnan, Nazim Agoulmine, Djamel Khadraoui and Abdelaziz Khadraoui. **"Block chain-based multi-party smart contract for service digitalization and automation"**. In: Future Technologies Conference. Series Title: Lecture Notes in Networks and Systems. Springer International Publishing (2021).

Chapter 2

Transport and Management of Dangerous Goods (TMDG)

2.1 Introduction

In this chapter, we present an extensive study concerning the Transport of Dangerous Goods (TDG). Initially, we present the definition of Dangerous Goods (DG), the main stakeholders[1]' involved in the process, and DG classification based on their substantial specificity. The SpC of DG is complex and belongs to the regulated domains. The regulatory framework governs TDG entirely. We then introduce the TDG regulatory framework applied at the national[2] and international level. The TDG requires strict procedures for preparing transportation and its specificity, documentation, and DG treatment. The DG storage, treatment, re-use, or processing (for industrial purposes) refers to the management part of DG. Finally, in this first part, we present several specific procedures (safety procedures, loading, unloading procedures, and transportation procedures) in TDG, which need to be considered when operation with DG.

The transport and management of DG (TMDG) has been a subject of study in academia and industry. Most of these studies are focused on addressing risk mitigation in TDG, transport scheduling issues, and process monitoring. Therefore, in the second part of this chapter, we present a research synthesis from the academic and industry perspectives and their efforts to design and develop a software-oriented solution for the TDG. For describing the process of TMDG, this thesis focuses on road transport of DG since it is one of the most used TDG mode, and our use case is focused on this type of transportation.

2.2 Dangerous Goods (DG)

Dangerous goods (DG) are considered as any material or substance or a mixture of substances (gases, liquid, or solids), which exposes potential risks (identified as hazardous) for harming humans, animals, property, and the environment (UNECE, 2017). In our daily life, we are surrounded by DG. In case a DG is not managed correctly, they represent a high level of

[1]Stakeholder (Actor): A *stakeholder* is either an individual, organization or group of organizations that are affected or that affect the enterprise (or organization) (Frooman, 1999).

[2]In the national context, we refer the regulatory framework applied in Luxembourg.

threats to the safety and security of human beings, the environment, and properties. The daily activities of businesses and human needs (domestics products, e.g., household) are requiring DG. It is the elemental source of activities for different industrial sectors, for example, oils, waste treatments, and other liquids.

DG are classified based on their physical and chemical effects (UNECE, 2017). The DG identification and classification play a significant role in establishing an action for safety and security for an appropriate way of transporting DG. That helps in indicating the DG involvement, therefore taking the necessary precautions (GuideADR, 2018). The DG classification is provided by ADR (Part 2). It defines specific criteria for classifying DG. Table 2.1 shows all the classes of DG, based on ADR criteria, while Appendix A.4 shows corresponding symbols for DG classes.

The national legislation for governing the TDG by using road (land) mode of transport is based on the ADR (UNECE, 2017). There are currently around fifty countries that have already adapted their legislation based on ADR (GuideADR, 2018). The responsibilities of national authorities involved in the monitoring and implementing general practical duties, obligations, and penalties are divided based on nature (hazardous) of DG. For example, for the Explosives (Class 1), the responsible authority is usually the Ministry of Justice (Justice, 2020). For the TDG with radioactive range or infectious potentials, the competent authorities are usually the Environment Agencies (GuideADR, 2018). For example, in Luxembourg, for class 6.1 and 6.2, the Minister of the Environment[3] (Environment, 2020) is the responsible, while for radioactive materials, the responsible is the Ministry of Health (Health, 2020). The responsibility of TDG management incorporates other competent authorities that are part of the process with their specifies such as Road Safety Authority (i.e., SNCA[4] in Luxembourg) that is responsible for technical examination of the vehicle annually. The National Roads Administration is responsible for monitoring the national road network and imposing restrictions on DG transport through tunnels (GuideADR, 2018).

2.2.1 TDG Main Stakeholders, Safety Procedures and Restriction

ADR recommends that any participant engaged in the transport and management of DG shall take prior measures based on the nature and reaction of the DG, to avoid any injury or damage (ADR 1.4.1). In case of any potential risk foreseen, and the public safety is exposed, the participant should inform the emergency services immediately and provide them with the required information to take appropriate action (ADR 1.4.2)

For operation with the DG, various participants are involved, such are "consignors", "carrier", "driver and vehicle crew", "filler", "loader", "unloader", "consignee", "tank-operator/portable tank operator" and "DG Safety Advisor". ADR specifies their role and legal responsibilities in Section 4 (subsections 3.2 and 3.10). Following, we present the main participants and more precisely their duties and obligation as defined in ADR.

[3]The determination of responsible authority to manage DG is subject to change based on the country political decisions.

[4]SNCA: Transport international des marchandises dangereuses par route.

TABLE 2.1: Classification of DG according to ADR.

Nr.	Dangerous goods class	Sub-class	Description
1	Explosives	//	Substances and articles that expose high mass explosion potentials. Any substance or article that has intensive (medium or minor) fire or blast
2	Gases	*2.1: Flammable gases* *2.2: Non-flammable gases* *2.3: Toxic gases*	Any substance that has a vapour pressure of 300kPa or greater than 50°C or which are completely gaseous at 20°C
3	Flammable Liquids	//	This class presents any liquid or mixture of liquids that contain solids that have a flesh point in temperature 60-65°C
4	Flammable Solids	*4.1: Flammable solids* *4.2: Substances liable to spontaneous combustion* *4.3: Substances which, in contact with water, emit flammable gases*	Any material that under specific conditions emitted from the transport process might cause or contribute to fire. In this class of DG, there are include all the self-reaction substances that might react in contact with heating, air or water
5	Oxidizing Substances and Organic Peroxides	*5.1: Oxidizing substances* *5.2: Organic peroxides*	A substance (not necessarily combustible themselves) that, by yielding oxygen, might cause or contribute directly to the combustion of any other material around
6	Toxic and Infectious Substances	*6.1: Toxic substances* *6.2: Infectious substances*	This class of DG, covers any substance or material, that are liable to cause death or serious damages to human life, on a single action or short-duration action
7	Radioactive Material	//	This class of DG covers any material containing a radio activity above a certain degree
8	Corrosive Substances	*8.1: Acid substances* *8.2: Alkalis*	All the substances that by contract may damage or destroys other materials (or skin or human parts)
9	Miscellaneous Dangerous Substances	//	This class present all DG and its hazardous that are not covered from other classes

- *Consignor (or DG Provider)* presents any enterprise that provides its own DG or on behalf of the third-parties. The safety measures and duties are defined in the ADR (1.4.1).
- *Carrier (or Transporter)* is any enterprise that carries DG in accordance with ARD recommendation (1.4.2.2)
- *Consignee (or DG Receiver))* presents any individual or business who is responsible for taking charge of DG when they are delivered.
- *Driver and vehicle crew* is any participant that has control of the vehicle and fulfills the driving function. The crew member should be a participant that has appropriate training for the TDG duties and responsibilities.
- *Loader* is any individual or business who is responsible for loading the DG (1.4.3.1)
- *Packer* is any individual or business whose responsibility is the final packing of the DG (1.4.3.2)
- *Filler* is any individual or business whose responsibility is to fill the tanks of containers with DG (1.4.3.3).
- *Tank-operator/portable tank operator* is any individual or business whose responsibility is for the operation of a tank-operator/portable tank operator (1.4.3.4).
- *Unloader* is any individual or business whose responsibility is for removal (unload) DG from a vehicle (1.4.3.7).
- *DG Safety Advisor* a certified and competent person who can advise on the safe transport and management of DG at the national and international levels.

There are enormous *restrictions* in the process of TDG. Different countries might impose these restrictions that may adapt their national regulatory framework for additional safety reasons (Article 4, paragraph 1 in ADR) (UNECE, 2017). Further, the TDG has a significant restriction when they have to pass through the tunnels. In such a case, a special authorization should be required from different competent authorities hosted in different countries (ADR, scope 1.5.9.1)

2.2.2 Consequences from Misuse of DG

DG is considered a source of contamination in case they are used for another purpose other than they are intended. ADR recommends keeping DG secured by providing "security measures". The security aspects aim to minimize theft or misuse of DG (1.10.1). The threats and possible damages by DG are at a high level. The reactive nature of the DG is challenging for the involved stakeholder and also the competent authorities for maintaining them under surveillance. For example, Class 7 of DG is powerful to destroy entire society, and if they are used for malicious purposes, the consequences might be enormous (1.10.3).

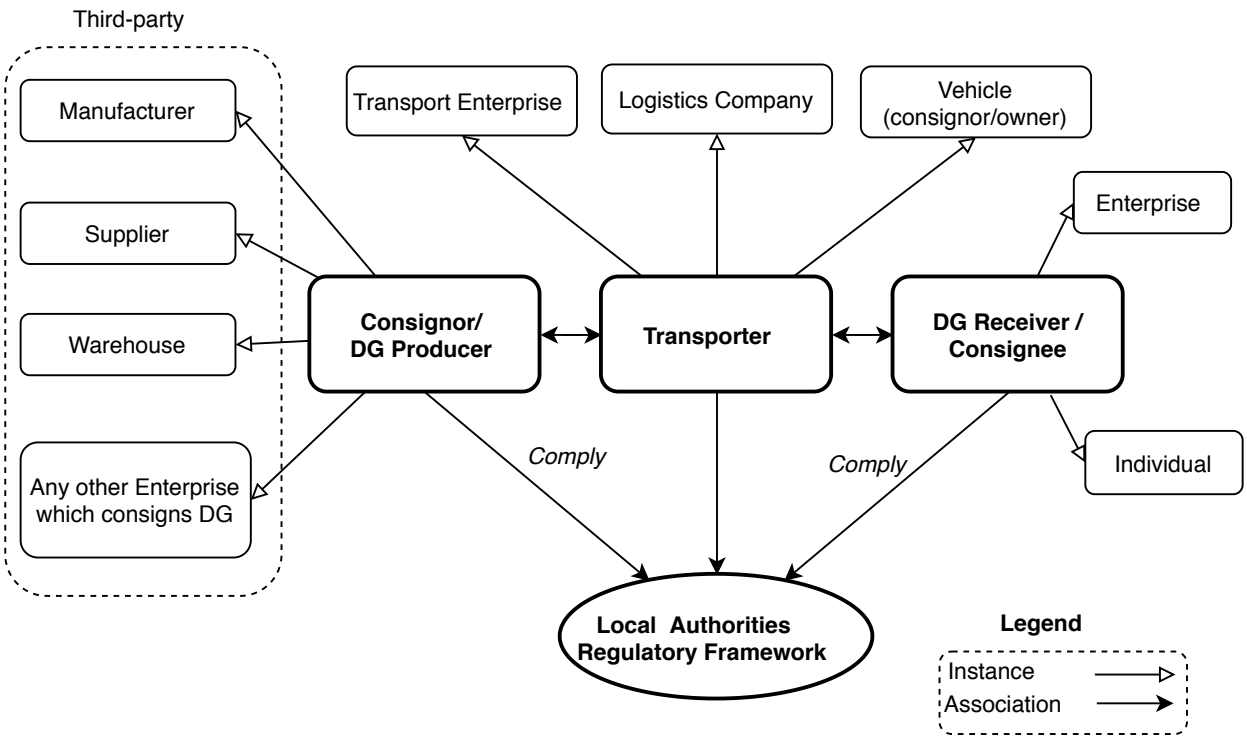

FIGURE 2.1: The main stakeholders (entities) involved in the process of TDG, based on (UNECE, 2017).

2.2.3 Process of Transport of Dangerous Goods (TDG)

In the context of TDG, many stakeholders cooperate to fulfill the process of transportation. This process is very delicate because of the risks to human life and the environment. The process is complex due to the many standards and strict local and international policies (Imeri et al., 2019b). For correct TDG process management, stakeholders are obliged to appoint a dangerous goods safety adviser (DGSA) if they need to manage a large amount of DG (GuideADR, 2018).

Many processes must be followed for the TDG from the point of origin to the point of destination. These process generate sensitive information to be shared among the stakeholders. Examples of this information are contractual issues, types of goods to be transported, the timestamps of movement of goods, warehousing, and the mode of transportation" (Imeri et al., 2018). Figure 2.1 presents the main stakeholder (entities) involved in the process of TDG and their fundamental interactions.

The TDG requires a high level of preparation for the involved stakeholders. For example, "TDG Providers" are required to follow specific regulations on packing, labeling, loading, and use special vehicles for TDG. For the " Carriers (Transporters) ", it is required to have trained drivers, special vehicles, and prior routing plans for performing the safe process of TDG (UNECE, 2017). When an intermediate stop is required (for a certain time), the warehouse providers should comply entirely with the regulatory framework for TDG at the local level (depend on country) and international level.

TABLE 2.2: International regulatory framework and agreements for TDG, specific to the mode of transport.

Mode of Transport	Regulatory Framework	International Organization	Abbrev.
Road (Land)	European Agreement on the International Carriage of Dangerous Goods	UNECE (UNECE, 2017)	ADR
Inland Waterway	European agreement on the international carriage of Dangerous Goods by Inland Waterway Navigation	UNECE (ADN, 2016)	ADN
Rail	Regulation for the International carriage of Dangerous Goods by Rail	OTIF (RID, 2019)	RID
Sea	International Maritime Dangerous Goods Code	IOM/CCC (OTIF, 2019)	IMDG Code
Air	a) Dangerous Goods Regulations b) Technical Instructions For The Safe Transport of Dangerous Goods by Air	ICAO (IATA, 2017)	DGR IACI IT

Besides TDG by the road, different transport modes are used for the TDG (Croneri, 2020; Serpac, 2020), as showed in Table 2.2 (in the first left-column):

2.2.4 Regulatory Framework for TDG

The TDG is performed by using different modes of transport. The research from (Torretta et al., 2017) indicates that around 60% of TDG is covered by road transport, motivated by its lower cost of operation and feasible way of organizing the TDG. For the different modes of transport, specific regulatory frameworks govern TDG. Table 2.2 shows a different regulatory framework for the process of TDG based on transport mode.

For the TDG, beyond the international regulatory framework (2.2), it should also fulfill all the legal requirements imposed by the country where a stakeholder of TDG is operating. Each country reserves its rights to organize their own process of TDG by adapting or newly providing regulatory frameworks. Furthermore, several bilateral or multilateral agreements are signed and applied between countries as a legal arrangement for TDG. For example, Table 2.2 presents several directives, regulations, and laws applicable to the TDG in several countries, including Luxembourg.

2.2.5 Main Components to Consider for TDG

The DG should be classified, marked, and labeled as indicated in the regulatory framework. The DG producer (i.e., consignor) is legally obliged to classify, packing, marking, and labeling the DG. According to ADR classification, all DG that are subject to transport should have an appropriate label, that for the transporter to have prepared the safety procedures in case of accidents (UNECE, 2017; GuideADR, 2018).

Packaging: For TDG, packaging manifests the process of packing the DG, according to the DG that are subject to transport. Depending on the type of DG, appropriate packaging is suggested by ADR.

TABLE 2.3: The laws, directives, and regulations for TDG used in the local and international context.

Directive/Regulation	Description
Directive 2008/68/EC	Inland transport of dangerous goods
Regulation EC/1907/2006	Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH), establishing European Chemicals Agency
Regulation EU/649/2012	Export and import of hazardous chemicals
Law of 24 December 1999	Safety advisers for the transport by road, rail and inland waterway of dangerous goods
Regulation of 1st March 2007	(a) Road transport of dangerous goods; (b) Duties and training certificate of the Safety Adviser for the carriage of dangerous goods by road, rail or inland waterway
Law of 16 December 2011	Registration, Evaluation, Authorization and Restriction of Chemical Substances and the Classification, Labeling and Packaging of Chemical Substances and Mixtures
Regulation of 23 February 2008	Carriage of dangerous goods by rail
Regulation of 23 February 2008	Carriage of dangerous goods by road
Regulation of 30 July 2013	Restriction of the use of certain hazardous substances in electrical and electronic equipment (RoHS)
Regulation (EC) No 1013/2006	Shipments of Waste

Marking: ADR provides a unique way of marking DG. It provides general UN[5] codes (UN Number) (Notion, 2015). During the TDG, the marking indicates the "UN" packing symbol. Following, it indicates the type of package used (e.g., plastic, tank, or iron) by providing a specific code, then further it indicates the packing group (PG I: "high danger", PG II: "medium danger", PG III: "low danger"), by corresponding capital letters "X, "Y", and "Z". An example of the marking of a DG is as follows "**UN**21HA/Y/0006/DP/..." (GuideADR, 2018).

Labeling: ADR specifies a clear requirement on labeling of DG package. The labeling gives a visual sign on the DG and aims to raise the warning for everyone. ADR provides a label for any DG, as presented in Appendix A.4. Figure 2.2 shows an example of a DG prepared for transport. Besides the DG package (pallet, container, tank, or others) labeling, the vehicle that carries the DG is obliged to carry an orange plate placed on it.

Documentation: The process of TDG requires strict documentation. It presents an essential aspect of TDG. Among the main documents required are related certificates for DG that are carried, emergency response information, and driver's qualifications. ADR presents an extensive list of documents that should be considered for the TDG. These documents are regarding "current DG in transport", "large container (or vehicle) package certificate", "vehicle certificate", "driver identification certificate", "driver training certificate" and many others (GuideADR, 2018).

[5]United Nations.

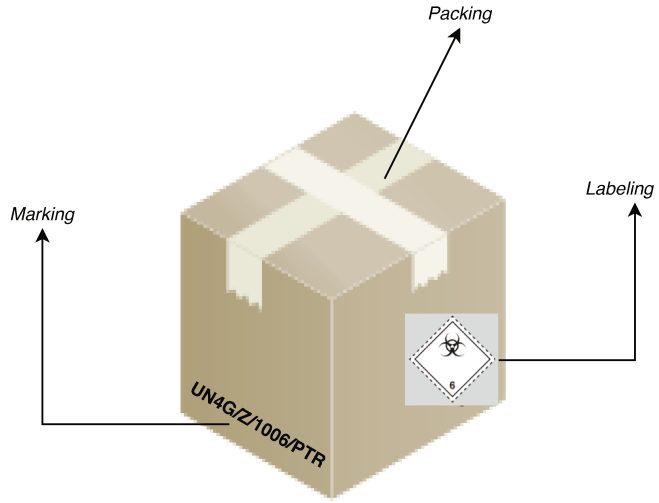

FIGURE 2.2: An example for packing, marking and labeling the DG.

Vehicles and Safety Equipment: For some specific DG, there is required to have specialized vehicles, e.g., transport of explosives, tankers for liquid materials (highly flammable). For the other DG, vehicles might be standardized vehicles, e.g., trucks, vans, and many others (GuideADR, 2018). Vehicle safety equipment (personal protection pieces of equipment) should be carried out during any DG activities (ADR, part 8).

2.2.6 Exemption of DG

ADR defines several DGs that are subject to an exemption under specific conditions. The exemption means that there is not required strict labeling of DG while transporting. The exemption criteria (presented on ADR 1.1.3, ADR 1.1.3.6) are focused on small loads, where the DG that are subject to transport are divided into small pieces, and they do not present any risk (UNECE, 2017; GuideADR, 2018).

2.2.7 DG Across European Union (EU)

The TDG has enormous activity across Europe. DG is used as the first material (raw materials) in many industries. The need for gases, oils, and medicine related products increases the activity with DG. Figure 2.3 show the activity[6] of EU (and EEA) countries with DG. This statistic presents the entire activity (any class of DG) with DG. The left axis presents years, from 2009 to 2018. The right axis presents the quantity expressed in *"thousand tonnes"* of DG for each country.

[6]Data source: Eurostat, available at the following link: https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=road_go_ta_tott&lang=en

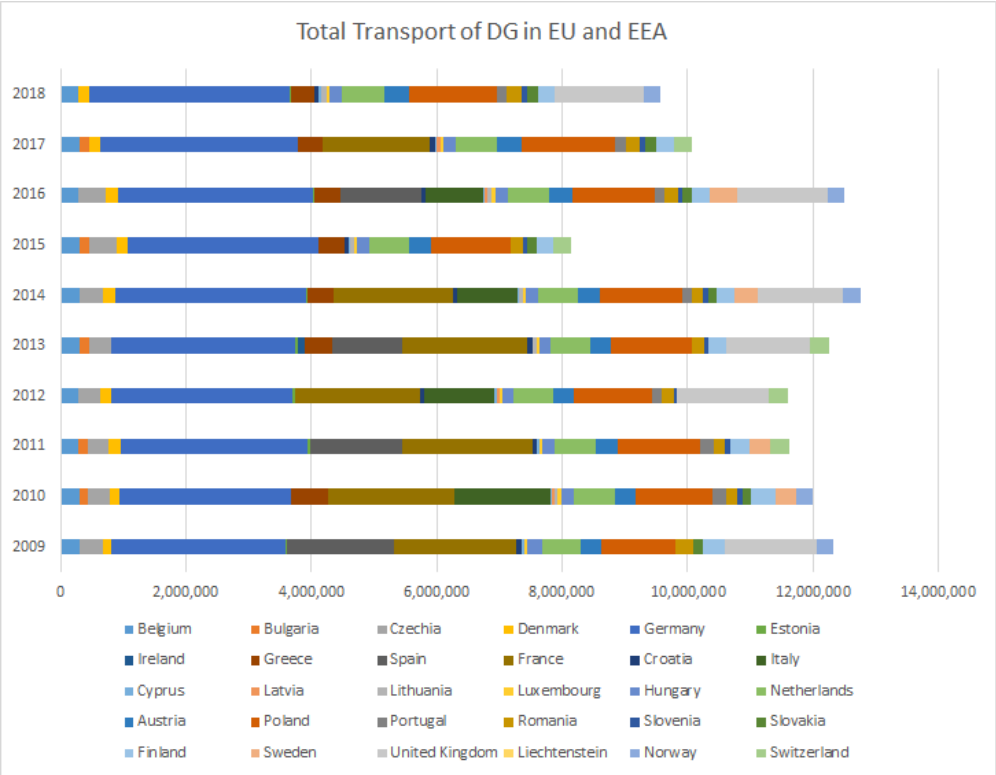

FIGURE 2.3: The statistics for total transport of DG in EU and EEA countries. Data source: Eurostat (Eurostat, 2019).

2.3 TDG from Scientific and Industry Perspective

The scientific community has studied the transport and management of DG. Their efforts concentrate on risk mitigation, monitoring the process of TDG, and highlighting the best practices for the management of the TDG. The scientific approaches for risk estimation intend to decrease the negative impact in case of accidents with DG. Table 2.4, we summarize some of the research papers that address risk, surveillance, and TDG process best practices.

2.3.1 Decision Support Systems (DSS) as a Management Tool for TDG

The risk involved in TDG comes from the nature of goods that are subject to transport. For the risk estimation and management of the processes of TDG, there are designed and develop decision support systems (DSS) as a computer-based solution. Several projects in the state of the art have addressed issues related to risk estimation, management, and decision-making support (Decision Support Systems) in TDG. The list of these projects is reviewed by research on (Torretta et al., 2017). The basic idea behind DSS is to help stakeholders to measure the risk for TDG, to save time on the critical decisions, monitoring the process of transportation (Torretta et al., 2017), decrease the negative impact in case of accidents with DG (Zografos and Androutsopoulos, 2008), scheduling, planning, and resource allocation (Ocalir-Akunal, 2016; Frank et al., 2000; Torretta et al., 2017).

TABLE 2.4: The summary of scientific approaches and research projects for the TDG.

Scientific approach (project) for TDG	Key advancements
The framework presented in this research uses the Internet of Things (IoT) approach for managing DG. As a framework, it provides the functionalities for: "Container Information Forecast" which allows the owners to have the information for containers in advance; "Gate-in and Gate-out" functionalities which include information for entrance and exit of containers; "Environment Monitoring" which allows monitoring of the environment where the DG is hosted to avoid any risk of explosion, firing, etc.; "Fire Fighting" which presets different method (water, sand, carbon dioxide) in case fire on dangerous goods (Ding et al., 2016).	IoT based approach; Information monitoring; Forecast; Entrance/Exit; Environment; Risk mitigation.
The traffic flow and the frequency of accidents are the primary analysis in this research paper. This research analyzes the road accidents where DG are involved. Performing such analysis leads to the reason for the leakage of hazardous materials. This research proposes a methodology for routing solutions by identifying hazardous accidents and by performing a risk assessment for the road transportation of DG. In general, this research identifies the components, which are in correlation with the risk of TDG by roads (Conca et al., 2016).	Risk assessment; Identification of components with risk potentials; Routing proposition.
This research highlights the need for an appropriate design for inland terminals for containers with DG. It studies and implements the specific criteria for designing inland terminals to avoid risks. It applies the multi-decision theory, and it includes several criteria simultaneously. Among the new criteria discovered for designing inland terminals, the criteria related to safety and security, environment maintenance, and information and communication technology (ICT) are part of this study. This study considers concurrently all these criteria for extracting the best solution. The method here is composed of "Analysis," which defines the problem and the criteria to be considered, then by "Synthesis," which proposes a development model, and finally, the "Evaluation," which shows results and conclusion (Molero et al., 2017).	Design of inland terminals; Criteria of design; Risk mitigation.
This research proposes to use the theory of D numbers for the route selection for TDG. This approach takes into account the "cost" of transport when selecting the specific route, and then it considers the "risk" exposes by using road transport route and "response-ability" in the case of accidents. These are the main criteria this approach uses to evaluate the routes, and it uses a specific algorithm to find the optimal routing. A real case study is shown in this research by presenting the results of the selection of the optimal route in specific (Wang et al., 2016).	Cost measurements; Risk analysis; Responsibility.
This research paper treats the risks related to the process of TDG. It presents an extensive study on the existing decision-making systems that are developed for the management of the process of TDG. This review made for decision-making systems: HAMER; HAMER PATH Spatial DSS; TrHaM; TrHazGis; TRAT- GIS 4.1; DESTINATION Project 2014 – SIIG (Torretta et al., 2017).	Risk analysis; Review of existing TDG related ICT systems.

Scientific approach (project) for TDG	Key advancements
This research shows an overview of the risk-based methods, for the case of measuring the risk acceptance for TDG by using road tunnels. Initially, a risk threshold is formulated. Further, the road tunnels might allow using for TDG in the risk acceptance criteria is achieved, i.e., the risk threshold is not reached or passed. Three different risk assessment methods are proposed for the TDG such as "qualitative", "semi-quantitative", and "quantitative approaches". In general, these methods tend to identify the hazards (danger), performing risk analysis, and evaluation of general risks in the report with specific risk acceptance criteria (Benekos and Diamantidis, 2017).	Risk assessment; Risk acceptance; Risk threshold; Methods: - Qualitative; - Semi-Qualitative; - Quantitative.
This research presents a new system that follows the movement of DG and human mobility data. The system called DGeye uses the risks pattern for TDG and then it redefines the causal network among these patterns by identifying the possible risky points on this network (Wang et al., 2017).	Risk assessment; Risk pattern; Map with risky areas.
This research evaluates the risk of an accident with DG by considering the temporal aspects such as the volume of traffic, weather conditions. The proposed approach follows the multi-agent simulation to calculate the risk in the TDG (Kanj, 2016).	Risk calculation; Multi-Agent simulation.
This research uses the Dempster–Shafer method for estimation of the probability of accident occurrence on the road for TDG. Several risk factors are taken into accounts, such as long-route accident rate, road type, and traffic conditions. This method proposes a new way of combining these factors to estimate the probability of accidents on roads. This method allows a combination of estimation from different sources and reaches the degree of satisfaction. Compare to the Bayesian method, which requires the estimation of prior and conditional probability, the Dempster–Shafer method takes into account the uncertainty (Leung et al., 2017).	Probability estimation; Manages uncertainty; Risk factors: - Accident rate; - Road type; - Traffic condition.
EU funded project MITRA developed an operating system for the monitoring of the TDG in Europe and took regional responsibility into account. In real-time, the position and content of vehicles should be known in the area of responsibility. In case of emergency, warnings and alerts will be created, and this crisis management intervention team will be informed (Planas et al., 2008).	Monitoring; Intervention; Real-time; Crisis management.
This research proposes a middleware for real-time monitoring of the transport of dangerous goods, i.e., the shipment of oil trucks along in Europe and USA (Laarabi et al., 2014).	Tracking; Real-time; Telemetry; Event-data.

TABLE 2.5: TDG from the market perspective.

Project name	Description	Key advancements
HAMER (Hazardous Material Emergency Response System)	The basic idea behind this DSS is to minimize the risk in case of accidents with TDG by minimizing the duration of and the impact of accidents (Zografos et al., 2000)	Minimize risk in case of accidents
ORISIS	A software simulator that helps to organize training for the stuff in case of fire, explosive, gas release, toxic release, and many other simulation (Tixier et al., 2002)	Simulation of accidents; Training.
WISER (Wireless Information System for Emergency Responders)	The purpose of this system is to respond to an emergency situation when transporting dangerous goods. WISER provides mechanisms for the identification of substances, information about hazardous substances and their physical characteristics, and the impact on human health (WISER, n.d.)	Information on substances; Respond to emergency.
HAZMAT PATH Spatial DSS	The objective of this DSS is to provide possible restriction and recommendation for route selection in process of TDG (Frank et al., 2000)	Route Recommendation; Provide restrictions.
TrHaM (Transport of Hazardous material)	TeRaM is a DSS that quantifies the risk for TDG for a different mode of transport (water, pipeline, roads, and railways), and the basic idea is to help decision-makers to plan their transport activities (Torretta et al., 2017)	Risk analysis; Decision support on planning.
TRAT-GIS 4.1 (Transport of Hazardous GIS)	The core intention for this DSS is performing risk analysis for TDG by road, rail, and pipeline. It quantifies the risk for these modes of TDG. The riks is calculated for people, the environment, and also the "total" risk which includes environment-people (Torretta et al., 2017)	Risk analysis
HAMER (Hazardous Material Emergency Response System)	HAMER is a DSS that intends to minimize the risk in case of accidents with DG. The key improvements for this DSS are its focus on minimizing the duration and the impact of accidents (Frank et al., 2000)	Decision support; Risk minimization of accidents: - Duration - Impact
DESTINATION project (Global Integrated Information System)	Is a DSS for assessing the risks for the environment, accident prevention and management and monitoring of DG (Torretta et al., 2017)	Risk analysis; Decision support; Monitoring; Management.
DG-ASSIST	Creates IATA, IMO-IMDG, ADR, and 49 CFR transport documents (shipping declaration and checklist). This DSS intends to manage almost any task related to TDG, hazardous chemicals, and waste (DGAssistant, n.d.)	Transport documents creation; Module for waste management.
SAP Dangerous Goods Management	Assess unpacked and packaged dangerous goods. Classification of DG according to regulation (SAP-DG, 2020)	Assessment DG; Classify DG.

Project name	Description	Key advancements
XENVIS	This DSS supports risk assessment and management of tasks related to DG, by acting in compliance with the European regulatory framework. It is composed of geographical information systems (GIS) at the national level. It provides information about substances, simulation of models for industrial air pollution, and performs routing recommendation (Xenvis, n.d.).	Risk assessment; Regulation assessment; Geographic Information System (GIS); Simulation model; Route Recommendation.
RAGISARD	A Tool for GIS Based Risk Analysis for TDG on Road. This DSS performs risk analysis in order to assess the environmental risk for the TDG (Resigard, n.d.)	Risk assessment; Decision support; Regulation assessment; Geographic Information System (GIS).

In general, the architecture of these systems is composed of several other systems. The embedded systems for "Sensors", "GPS tracker", "RFID", "GIS for moving objects", and other related ones, which are integrated into the main architecture of DSS, provide information for the process of TDG. The risk analysis, monitoring of the process of TDG, and other related tasks are depended on the current state of information that should be provided by these systems. Table 2.5, presents the information systems i.e., DSS and ICT[7] platforms designed for the management of the process of TDG, including project from market perspectives.

2.4 Conclusion

This chapter presents the general information about DG, their classification, and the specificity of their transportation. We presented the TDG process organization, involved stakeholders, and main procedures. Further, it highlights the regulatory frameworks that govern TDG based on transport mode. The risk involved in TDG is strongly related to the DG nature. For the risk estimation and managing the process of TDG, decision support systems (DSS) as computer-based solutions have been developed. The basic idea behind DSS is to help stakeholders measure the risk involved in TDG, save time on the critical decision, monitor transport, decrease the negative impact in case of accidents with DG, scheduling, planning, and resource allocation. The information generated in the TDG is mainly stored by DSS in a local databases of the stakeholders and further analyzed. This approach raises several concerns in terms of security of the information, its reliability, TDG process efficiency, and trust issues regarding the sharing of information about the TDG process between involved stakeholders, including authorities in the process of TDG.

[7]ICT: https://en.wikipedia.org/wiki/Information_and_communications_technology

The objective of this research intends to propose new approaches to manage the TDG process workflow, to improve the TDG process, and achieve a better security and transparency of the processes and the information storing and sharing. Our research targets novel technologies that enable a compliant, trusted, and transparent TDG process.

Chapter 3

Technologies: Centralized, Distributed, and Decentralized

3.1 Introduction

This chapter aims to synthesize the existing technologies used to support business processes. These technologies are used daily and are dedicated to improving the quality of services and facilitating human activities in different domains. The technological aspects optimize the way business processes are organized by enhancing efficiency, reducing human involvement, and improving the work quality. Beyond that, specific cases supported by technology are not at the expected level. That is mainly because of a lack of information security aspects, missing interoperability between different technological platforms, and technology maintenance issues. In this chapter, we discuss different standard technologies and their main characteristics. Initially, we present the centralized (client-server) architecture; then, we present distributed systems characteristics, i.e., Cloud Computing, including the Internet of Things (IoT). Furthermore, we present the decentralized distributed systems, i.e., distributed ledger technology (blockchain), and their main characteristics. We show an extensive study over the blockchain technology since it is the considered technology behind this thesis research. We deliver a comparative table (3.2) which summarizes the technological features of each technology studied in this chapter.

3.2 Centralized: Client-Server Architectures

Client-server (CS) architecture presents one of the earliest and most used types of technological architecture. CS architecture describes a configuration of an information system involving a broad range of technologies and incorporating various business cases (McFarland and Nicholson, 2007). CS architecture is primarily composed of two software processing sides that cooperate in providing the desired functionalities. CS architecture is mainly composed of the server-side and client-side. The server-side stores and manages user data, processing application data, and user queries. The client-side allows users to query formalization, send a query in server-side, and receive a response (query result) from the server-side. Figure 3.1 shows the general schema of CS architecture, for example, any time we request information

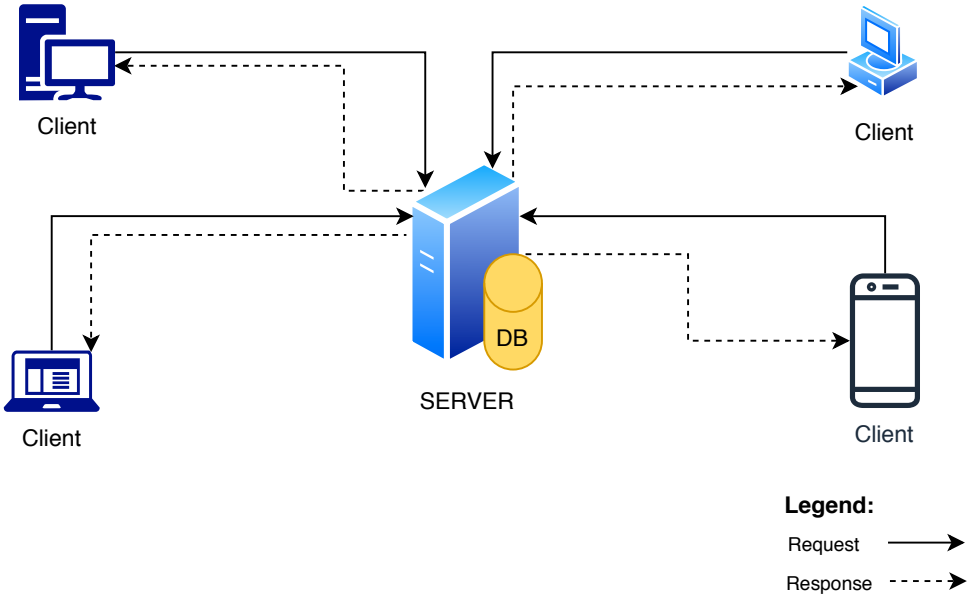

FIGURE 3.1: The architecture of the centralized client-server system computing approach.

through a web page (client-application) by using an Internet web browser[1] in a specific application (web-server) we are using CS architecture. CS physical architecture is mainly composed of the client-side, e.g., a personal computer or mobile device, and the server- side, which is a centralized, powerful computer node, ready to respond to client-side requests. The communication between client and server is enabled by using a network infrastructure (LAN[2] or WAN[3]). The software architecture in CS is seen as a partition of an application that supports a specific side in CS, i.e., the software that processes client requests and sends them to server software that further processes that request and responds according to its programmed features (McFarland and Nicholson, 2007; Borrie, 2004; Kanter, 1998; Lewandowski, 1998; Hanson, 2000; Berson, 1996). Even today, CS remains one of the most used types of software architecture, and many financial institutions, such as banks and other organizations, use CS to develop their application, maintain client information, and perform daily activities (tasks).

CS has some enormous issues that need to be addressed. In CS, there is no guarantee regarding data loss, data altering, and data integrity. Also, CS is expensive to deploy since specific IT infrastructure is required, and the maintenance costs are very high (McFarland and Nicholson, 2007). Table 3.2 shows a summary of the general issues of CS architecture.

3.3 Distributed Architecture

This section aims to present the distributed architecture models, such as Cloud Computing and the Internet of Things.

[1] https://en.wikipedia.org/wiki/Web_browser
[2] https://www.webopedia.com/definitions/local-area-network-lan/
[3] https://www.fortinet.com/resources/cyberglossary/wan

3.3.1 Distributed Cloud Computing Architecture

Cloud Computing (CP) is a distributed computing model that provides virtual computing resources (shared pool of configurable computing, e.g., server, storage, network, application, data and services) and software solutions that exist in a distributed cloud computing infrastructure (Hoy, 2012). These computing resources are accessible through a computer network. There are various definitions of the term CP. For example, the National Institute of Standards and Technology (NIST) defines CP as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell and Grance, 2011). NIST also defines major CP entities, including a) "Cloud Consumer (user)", a person or organization that uses the CP resources; b) "Cloud Provider", an organization or person that provides CP services for the cloud consumer; c) "Cloud Auditor", a responsible, independent organization in charge of the evaluation of cloud services and determining effectiveness and security; d) "Cloud Broker", a third-party intermediary organization, legal entity, or person positioned between the cloud provider and cloud consumer in the negotiation of terms and condition for contracted cloud services; and e) "Cloud Carrier", any organization or person that enables the connectivity of cloud services from the provider to the cloud consumer (Mell and Grance, 2011; Odun-Ayo et al., 2018).

CP made it possible to move from traditional client/server architecture into the distributed architecture of interconnected data centers located in different geographic locations. CP allows accessing the computing resources is performed over web (or other accessing tools) (Hoy, 2012). Figure 3.2 presents the high-level schema of CP architecture. CP may be set up for a variety of purposes (e-mail, document management and sharing, storage, computing resource sharing, and many others), and is accessible from various devices depending on the needs of the cloud consumers (Velte et al., 2009). Beyond that, in CP, users can use cloud provider infrastructure to configure and run their computing resource without a human help (Mell and Grance, 2011). The access into CP resources is available over internet (Mell and Grance, 2011; Velte et al., 2009). CP mainly do not dedicate specific hardware for each user, and multiple users use the same hardware and resource may be distributed several data centers (Mell and Grance, 2011; Velte et al., 2009). CP user is permitted to configure their dedicated computing resources, to adapt their needs for scalability in their application (Mell and Grance, 2011; Velte et al., 2009; Hoy, 2012). For the computing resources that the users "consume", they pay them based on the payment model offered by the cloud provider, and one of them is "pay per use (pay-as-you-go)" supported by many cloud providers (Mell and Grance, 2011; Gong et al., 2010; Patidar et al., 2012; Odun-Ayo et al., 2018).

There are different types of CP: "Software as a Service (SaaS)", "Platform as a Service (PaaS)", and "Infrastructure as a Service (IaaS)".

In Software as a Service (SaaS), cloud users launch their application on the cloud using a web browser (e.g., web-based email access, Google Docs, and so on), instead of installing them on their local computers (Srinivasan, 2014).

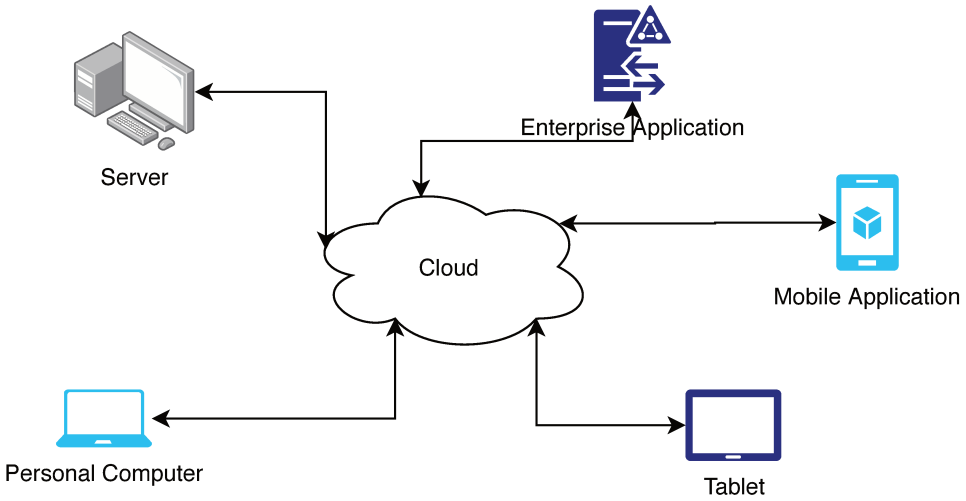

FIGURE 3.2: The high level architecture of cloud computing (Hoy, 2012).

The "Platform as a Service (PaaS)" allows users to access specific platform toolsets (programming environment, tools, configuration management, and many other platform-related components) to build and deploy cloud-based applications. Examples of PaaS[4] are SAP Cloud[5], and Microsoft Azure[6] (Odun-Ayo et al., 2018; Srinivas et al., 2012; Mell and Grance, 2011; Srinivasan, 2014).

The "Infrastructure as a Service (IaaS)" enables cloud consumers to directly use computing resources, such as storage, processing, memory, network, and other IT infrastructure related components. IaaS uses cloud virtualization (virtual machines) as an isolated environment to dedicate computing resources for cloud consumers (Odun-Ayo et al., 2018; Srinivas et al., 2012; Mell and Grance, 2011; Srinivasan, 2014).

The research in (Dillon et al., 2010; Singh and Chana, 2016; Popović and Hocenski, 2010) studies the CP issues. Besides opportunities offered by CP, there are enormous drawbacks to consider before deploying any cloud based solution. Major issues include information security, performance, availability, hard to integrate, costing models, and many others. We highlight elements of the drawbacks to CP in Table 3.2.

3.3.2 Internet of Things

In today's trendy technologies, the Internet of Things (IoT) is recognized as one of the novel technologies that has made a major impact in the transformation of business processes. By adding the ability for physical objects to communicate with business applications, IoT technology has enhanced different vertical domains by improving the quality of service (QoS). IoT describes a set of devices that are able to collect, exchange, and share information. IoT devices sense data from objects and their context, perform computing and establish communication between devices and data transmission channels and actuation (Voas, 2016;

[4] https://www.webhostingsecretrevealed.net/blog/web-business-ideas/paas-examples/
[5] https://www.sap.com/products/cloud-platform.html
[6] https://azure.microsoft.com/en-us/

Rafique et al., 2020; Ammar et al., 2018). The IoT is considered a distributed processing model since IoT devices may perform some processing on data before sending it to the backend servers Yasumoto et al., 2016. Current IoT architectures are primarily composed of three different layers (Khan et al., 2012), "Sensing/Perception Layer", "Network Layer", and "Application Layer". Figure 3.3 illustrates the three layers of IoT architecture.

The *IoT devices layer (sensing)*, which is composed of different devices (sensors, RFID, actuators, IP cameras, GPS, thermostats, and many more), is responsible for gathering (sensing, measuring, identifying) specific data for a given use case.

The *network layer* enables the transmission of data to the application layer. For the deployed sensors, different network topologies are established (point-to-point, star, and mash) (Yaqoob et al., 2017). Short-range protocol infrastructure (ZigBee, Bluetooth Low Energy (LTE), WiFi, LTE-A, Z-Wave (Çorak et al., 2018)) is used to enable communication for IoT devices. Wide-range communication protocols such as SigFox and cellular networks (3G,4G) (Al-Sarawi et al., 2017) are used to transmit information to the application layer.

The *application layer* stores the data captured by the IoT sensing layer, enabling end-users to explore data collected from IoT devices (supported by standard protocols HTTP, MQTT, XMP, AMQP, DDS (Sultana and Wahid, 2019; Çorak et al., 2018)), and also transmitting instructions for the sensing layer (talk to IoT devices) (Qian et al., 2018; Ammar et al., 2018).

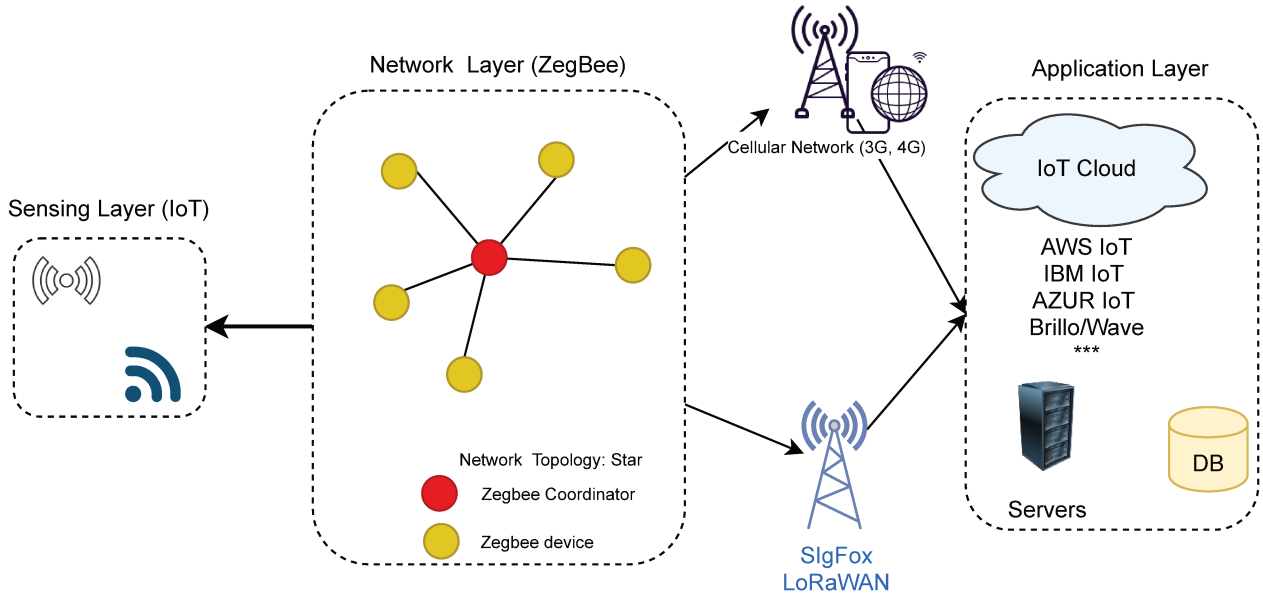

FIGURE 3.3: The overview of IoT system architecture.

There are already well-known concerns regarding IoT systems, such as privacy and security of information generated by IoT devices (Medaglia and Serbanati, 2010) (Yang et al., 2017) (Ammar et al., 2018), but they are out of the scope of our research. In the context of our study, we consider no security issue with the data transmitted by IoT devices since a proper security mechanism is implemented.

3.4 Distributed Ledger and Blockchain Technology

Ledgers have existed since ancient times and have served as record-keeping of transactions. At a particular time, a pen and paper ledger has been used to keep a track of trade and exchanging goods and services. With the emergence of technology, ledgers are now stored digitally in a large database maintained by a central authority. Distributed ledger technology (DLT) presents a distributed and decentralized database, shared among multiples parties, known as network participants. DLT is contrary to the centralized technology, meaning that the database in DLT is decentralized, synchronized, and shared among network participants. In DLT, information is stored based on consensus ("witnessing") and shared among multiples parties. Adding a new transaction record (A.7.4) in the ledger is possible only after the "witnessing" by the majority of network participants. BC technology is an instance of the distributed ledger, with the significant difference that in BC, the grouped transaction data ("block") are chained together with the previous block, thus forming the BC.

Blockchain (BC) technology[7] allows storing immutable cryptographically signed transaction data in a distributed decentralized database that is shared between multiple parties. The transactions executed from different BC user addresses (organization, person, government, entity, and many more (A.7.3)) are gathered into blocks, and each block is cryptographically linked to the previous block, thus making BC tamper-evident (Yaga et al., 2018). While BC is growing, older blocks become more resistant to changes through adding a new block in the chain (temper proof). The added (verified) block is distributed across network participants, thus keeping the digital ledger decentralized (Nakamoto, 2009; Yaga et al., 2018). Figure 3.4 illustrates the common components and characteristics of BC technology, and we explain some of them as follows:

- *Decentralized:* BC *network* is composed of several nodes that share the same ledger. The information added in the distributed ledger is done by agreement on the shared state on data from network nodes. The BC nodes are geographically distributed and rely on a peer-to-peer mode of communication. That ultimately leads to the unnecessary central authority to validate the information in the network (Yaga et al., 2018; Xu et al., 2016; Xu et al., 2017; El Ioini and Pahl, 2018).

- *Public (permissionless) BC:* In public BC, the network (BC) is accessed without any pre-required permission from any party. The public BCs are considered fully decentralized and rely on the independent-distributed nodes that maintain the network. Any user can join the public BC network, execute transactions, propose new blocks, or explore the block of the transaction conducted by other end users (Wang et al., 2018a; Xu et al., 2017; Cae-Imt-Inria, 2021).

- *Permissioned (consortium) BC:* In permissioned BC, users that intend to join the network must be certified (known by the consortium members). The network is maintained

[7] BC technology varies based on the implementation, and currently, there is an extensive list of BC platforms intended for a different purpose. This section presents generic terms and describes BC functionalities.

only by the certified members, and only they are allowed to execute a transaction and publish a new block. Permissioned BC is not entirely decentralized since an authority certifies network members, and a consortium of nodes maintains the network (Dib et al., 2018; Cae-Imt-Inria, 2021). Permissioned BC uses almost the same characteristics (consensus, distributed, semi-decentralized, data structures, storage, and many others) as permissionless BC. This type of BC is mainly used by organizations (public, private, or business organizations). The organization may use permissioned BC and invite their business partners or members to transact on the shared ledger. Certain types of permissioned BCs allow a particular level of privacy, enabling users to exchange transactions privately (Wang et al., 2018a; Yaga et al., 2018).

- *Fully Private BC:* In a fully private BC, different levels of access and read and write permissions are presented. These types of BC are considered almost centralized. Accessing the private BC network is intended only for a specific (basically invited) set of users, and permission must be granted by the host of the BC-based application to join the network (Wang et al., 2018a; Xu et al., 2017; Cae-Imt-Inria, 2021).

- *Consensus Algorithm:* In BC technology, the consensus refers to the agreement of nodes in shared content (El Ioini and Pahl, 2018). Different consensus algorithms are used, such as Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine fault tolerance (PBFT), and many more as we extensively present in section 3.4.2.

- *Data Security:* BC technology utilizes established and well-known cryptographic and computer science concepts, such as cryptographic hashing functions (Dang, 2015), and asymmetric cryptography (digital signatures) [8]. The data stored in BC are cryptographically checked. BC technology uses digital signatures (public key cryptography) for signing and verifying transactions (Xu et al., 2017; El Ioini and Pahl, 2018; Xu et al., 2016; Nakamoto, 2009).

- *Data Immutability:* BC is considered immutable. The data recorded on BC are cryptographically checked and distributed over all nodes in the network. To alter or rewrite the existing transaction in BC, the user should change all transactions simultaneously in all nodes in the network. That is almost impossible; meanwhile, the consensus algorithm compares the hash root of the transaction and denies these changes. Therefore, the BC transaction cannot be altered or deleted (Wang et al., 2018a; Xu et al., 2017; El Ioini and Pahl, 2018; Xu et al., 2016; Nakamoto, 2009). Even in BC, we may consider immutability conditioned since there are situations where BC may not stand as immutable. For public BC, the issues on BC immutability arise when 51% (BitcoinWiki, 2018) of processing power may rewrite blocks and makes the longest chain to be followed by new proceeding blocks. In permissioned BC, since there is an owner (consortium) of the network, this attack can be mitigated. Consequently, the control over the network may give some privileges to the network owner to replace any transaction based on a legitimate method (Yaga et al., 2018).

[8] In section A.7 we presents details about cryptographic components utilized in BC.

- *Auditability:* The timestamp and immutable storage of the validation transaction enables any user to trace the previous transaction executed by any specific user. That is possible by having access to the BC from any node in the network (Xu et al., 2017).
- *Smart Contracts (SC):* A computer code deployed in BC, which is executed to performing specific tasks after some predefined conditions are fulfilled. Section 3.5 shows the main characteristics of SC.
- *Low-cost maintenance:* BC technology does not use any central authority to exchange messages and validate transactions. That enables low-cost operations when using BC since there is no need to develop server infrastructures to validate transactions. That is in contrast to traditional systems, which use central servers for messages exchange and validation, and which usually have high database maintenance costs (upgrade, backup) (Wang et al., 2018a; El Ioini and Pahl, 2018).
- *Sustainability:* If several nodes fail or are disconnected, BC is still available and works properly on the remaining nodes. When the "offline" nodes come back into the "online" mode, they receive the latest state of the ledger (Wang et al., 2018a; Xu et al., 2017; El Ioini and Pahl, 2018).
- *Data management:* On-Chain vs. Off-Chain. In the context of data management, the common practice in BC is to store high volume (raw data) off-chain, and the hashes of these data are stored on-chain in BC. The on-chain data includes BC transactions with the hash of the "raw data" (Nakamoto, 2009; Poon and Dryja, 2016; Xu et al., 2016).

3.4.1 Blockchain Network, Data Structure, and Mining Process

A BC network (label: *Blockchain Network* in Figure 3.4) is composed of several nodes that are distributed geographically. A BC node is any computer or server that contains the updated ledger of blocks chained together. There are two types of nodes in the BC network: the nodes that only read transactions, and the nodes that read and write transactions. Nodes in the BC network communicate in a peer-to-peer fashion. The communication, exchange of information by executing transactions from the participants on the BC network rely on the interaction between nodes instead of any central storage of information. In the public BC, i.e., permissionless, any node can join the network and receive the full copy of the ledger.

The nodes that participate in forming the new block are called **miners**. A miner is a powerful BC node that initially collects transactions that are executed from different nodes in the network. There are different mining pools that hold the transaction collections. This collection of transactions presents the new potential "block" to be appended on the ledger (Xu et al., 2016; Xu et al., 2017; Wang et al., 2018a).

The BC ledger is composed of multiple blocks chained together. The blocks serve as "containers" of transactions executed from the network participants (BC users). A BC user submits a transaction by using digital software wallets, personal computers, smartphones, and other devices, and sends these transactions to the BC node or group of nodes, potentially forming a candidate block to appear on the BC ledger (Yaga et al., 2018; Paulavičius et al.,

2019; Fournier and Petrillo, 2020). Figure 3.4 illustrates the structure of the block (label: *Block Structure*), in turn illustrates the BC data structure[9] (Anceaume et al., 2019). The block is composed of the block "header," which stores metadata for the block, and the block "body", which contains a long list of validated transactions. The blocks are identified by the block cryptographic hash, which is generated by hashing the block header (Antonopoulos, 2017). The block header metadata indicates the *hash of previous block, the timestamp, difficulty target, Nonce, and the Merkle Tree Root*. The *timestamp*, indicates the time the block was created (appended into the BC ledger). The *hash of previous block* presents the hash value (digest) from the previous block, e.g., the block *n* contains the hash from the header of block *n-1*. The difficulty target presents the difficulty for block mining adjusted by the consensus algorithm ("Proof of Work"). *Nonce* is a numerical value generated by the consensus algorithm, e.g., Proof of Work (Antonopoulos, 2017; Wang et al., 2018a; El Ioini and Pahl, 2018; Yaga et al., 2018).

The *Transaction Root* is the root of all transactions received from the network users in a determined time. These transactions are organized in the tree by using the Merkle Tree[10]. In the content of block [11], we can see only the Merkle tree root, which is the root of all transactions on this block. The BC current block stores the hash of the block header of the previous block, thus chaining blocks and continuously grooving BC (Antonopoulos, 2017; Yaga et al., 2018). We present additional details about hashing, public-key cryptography, and wallets in Appendix A.7.

The process of forming and proposing a new block is called the "mining" process. The mining process requires high computing power, and the miners should prove that they have spent an enormous amount of computational power to propose a new block (in case of Proof of Work) or fulfill other conditions imposed by a different consensus algorithm (3.4.2). The computational power is adjusted and increased continuously (Kraft, 2016; Antonopoulos, 2017). The miner is rewarded for the work performed on mining the new block. The miners mainly carry any operation in the BC, and it requires incentives. For example, Ethereum uses gas as an incentive for the miners that carry operations (send ether, receiver ether, and queries). The miner rewards (incentives) are intended to motivate network participants to join and maintain the BC network. In general, public BCs are strongly dependent on the number of participants on the network. If this number is small, then the "honesty" of the network is questioned since the processing hash power might rewrite transactions (Tang et al., 2020; Yaga et al., 2018; Antonopoulos, 2017). This presents a conflict in the ledger since there may be inconsistency in the ledger (from the point of view of several nodes). That occurs when the block (n+1) mined from miner A contains transaction Tx_a, Tx_b, and Tx_c and block (n+1) mined from miner B transaction Tx_a, Tx_b, and Tx_d.

In the competition for solving the puzzle, two or more miners may solve the puzzle simultaneously, thus proposing a new block simultaneously. In such a situation, the BC forks into two or more branches, and the current (new) blocks point to the same parent block. In

[9] This inclusion is for the public BC, e.g., Bitcoin and (or) Ethereum.
[10] Description of the Merkle Tree is given in the following link Merkle Tree.
[11] An example of the content of the block is presented for this block.

Figure 3.4 (label: *Forks*, we present the forking visually. The "A" and "B" presents the conflict blocks, where for both of them, block *n* is the parent block in the ledger.

The block (n+1) is distributed to some nodes and will not be the same across the network. That creates a conflict since transaction Tx_c (but not transaction Tx_d) is in the block (n+1), distributed by node A, and transaction Tx_d (but not transaction Tx_c) is in the block (n+1) distributed by B. From the perspective of other nodes, the current BC states are not the same (accurate) since in some nodes, transaction Tx_c and Tx_d are not visible. BC intends to solve conflicts quickly to have a consistent ledger. Most of the BC networks wait until the next block is mined, and then it follows it as an official chain, thus adapting (following) the "longest" chain. Suppose the new block is mined on top of node "A"; thus, the "A" is the official block mined in the BC. Then any transaction that was presented in "B" but not in "A" will be returned to the mining pool and remain active to be mined after a particular time (Pîrlea and Sergey, 2018; Yaga et al., 2018; Wang et al., 2018a; Atzei et al., 2017; Antonopoulos, 2017).

The term *forking* is also considered when major changes are made on the level of BC protocol and data structure (Yaga et al., 2018). The soft fork is allowed nodes that have not implemented changes to transact with updated nodes. While in hard forks, all the participating nodes, should at a certain time (block number), implement technological changes on the BC in order to be able to transact with other nodes (Yaga et al., 2018).

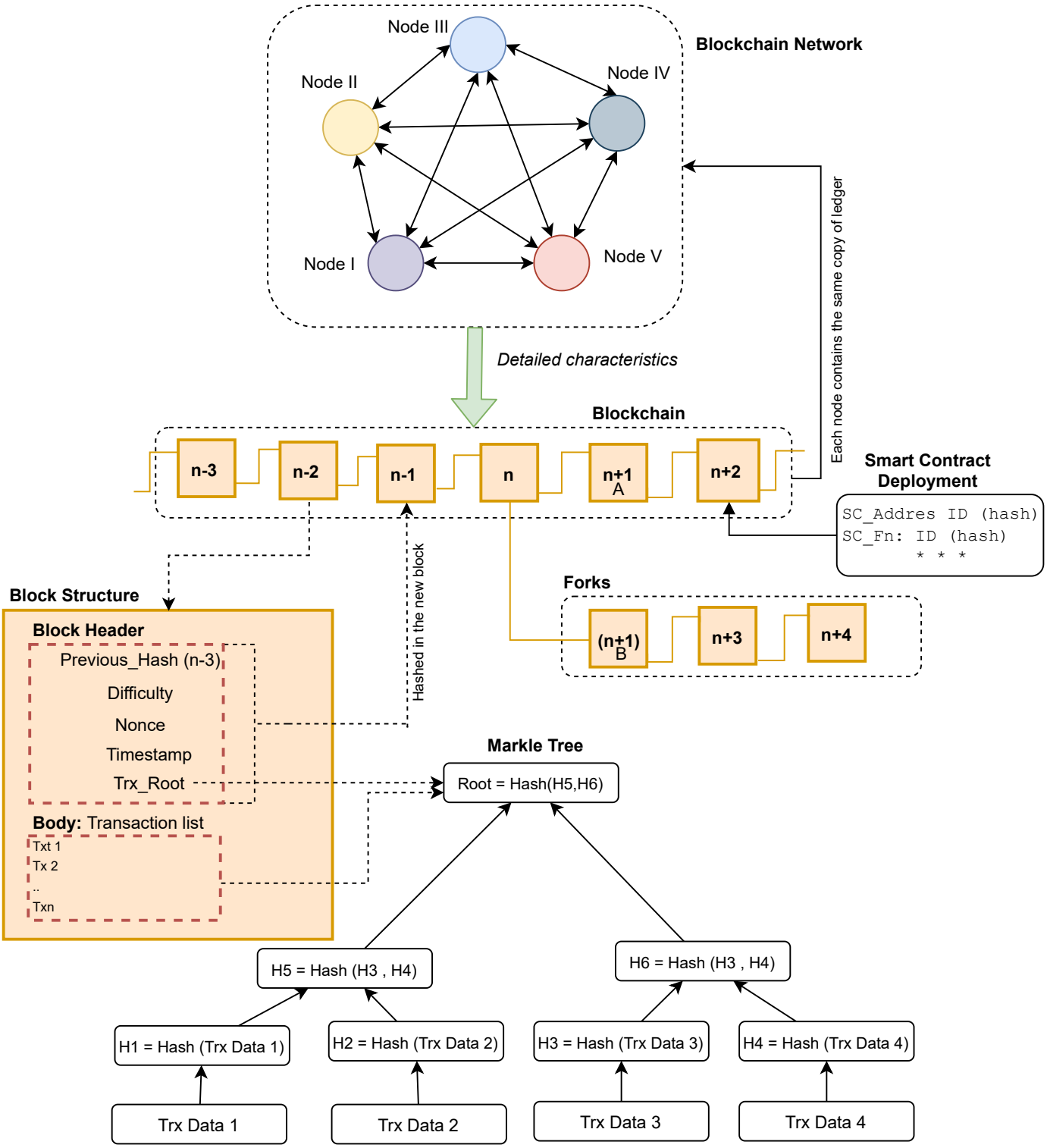

FIGURE 3.4: The blockchain technology components and main characteristics.

3.4.1.1 Incentives: Operations costs in Blockchain Technology

The mining process requires high computing power, and the miners should prove that they have spent an enormous amount of computation power to propose a new block. Bitcoin BC uses transaction fees as incentives for the miners that propose new block. Similarly, Ethereum uses mining incentives (Ether) for the miners that maintain the ledger state updated. There are other BC protocols that do not rely on incentive modes. For example, in Hyperledger Fabric (3.4.3), it is possible to deploy the BC network and to maintain it without relaying at the incentive model.

Gas[12] is considered as a unit of measurement of computational operation in Ethereum[13]. For any operation such as a transaction, SC deployment, or execution, some amount of gas is required. The EVM network executes in a decentralized way all SC that are deployed. To limit the resources an SC can use (e.g., exponential blowup and infinite loop code), Ethereum presents a mechanism by adding special parameters in the transactions. Besides standard parameters in the transaction (receiver of a transaction, the signature of the sender, the amount in Ether, and data to send), there are another two crucial variables: *startgas* and *gasprice* (Buterin, 2017). In the transaction execution, the *startgas* represents the maximum number of operations required, and the *gasprice* represents the amount that represents the transaction fee that the sender pays for performing computational steps. The *gasprice* represents the value in Ether that the sender should pay when executing the transaction. Sending a transaction with 3000 gas and 0.001 Ether *gasprice*, the total cost is 3000 * 0.001 = 3 Ether[14], which is considerably costly if many or daily operations are running on Ethereum. If the total of gas is not spent during the operation execution, it will be returned to the transaction sender (Wood, 2014; Buterin, 2017; Albert et al., 2020). If more gas is required to perform the operation, the execution will be in "out of gas" mode, which stops the operation and reverts all state changes, but not the transaction payment. Calculation of computational costs in Ethereum (in gas), are dependent and they vary on the cost of the computation or the data stored as part of the transaction execution (Bashir, 2017; Wood, 2014; Buterin, 2017; Albert et al., 2020).

3.4.2 Consensus Algorithms

The consensus mechanism is one of the essential components of BC technology. This mechanism ensures that the nodes agree on the current state of data and maintain BC network safety and security (Mingxiao et al., 2017). To add a new block to the BC, all nodes should reach a common agreement (Antonopoulos, 2017). The consensus algorithms are categorized as "proof-based algorithms", also known as probabilistic consensus algorithms, and "voting based algorithms" also known as deterministic consensus algorithms (Nijsse and Litchfield, 2020; Nguyen and Kim, 2018; Wang et al., 2018b).

[12] https://ethgasstation.info/
[13] https://www.luno.com/blog/en/post/understanding-ethereum-fees-how-gas-works
[14] The price of Ether varies continuously: https://coinmarketcap.com/currencies/ethereum/

3.4.2.1 The Byzantine Fault Tolerance (BFT)

Globally, the distributed computing issue is the overall reliability of a system where some nodes (agents) do not behave according to the global rules. These nodes might be large in number, but they may not comprises the majority of nodes in the computer system. "Byzantine Fault Tolerance (BPF)" states that a system tolerates a certain level of failure (Fault Tolerance) events (classes, nodes) on the system. In practice, some "evil" nodes might not cause the entire system to fail entirely. In distributed decentralized systems, the **fault tolerance** refers to the measurement of reliability to which a computer system might fail[15] (Castro and Liskov, 1999). The genesis of BFT is from "Two Generals problem"[16] and further it has been extended to "Byzantine General Problem" (Lamport et al., 1982). In the permissioned BC, several consensus protocols are proposed based on the BFT model, such as the Practical Byzantine Fault Tolerance (PBFT) used by Hyperledger Fabric (Sousa et al., 2018).

Several consensus algorithms are used by different BC platforms. In order to agree on appending of a new block on the BC, there must be an agreement between nodes in the BC network (Wang et al., 2018b). There is an extensive list of consensus protocols[17]. Following we list some of the most commonly used consensus algorithms by the current BC platforms.

Following, we present some of the **most common consensus algorithms** used in different **BC platforms.**

- **Proof of Work (PoW):**

  The *Proof of Work (PoW)* is the most commonly used consensus algorithm by BC technology, and it stands behind *Bitcoin* BC. In the PoW, a miner is the first among all miners to publish a new block as "proof" of work performed on solving the cryptographic puzzle (Nakamoto, 2009; Wang et al., 2018b). The design principles of PoW ensure that solving the puzzle is considerably difficult, but proving the puzzle solution is quite simple (by adding the nonce number to the block header hash). Once the miner solves the puzzle, it distributes the solution, and the other nodes verify the solved puzzle quickly, thus accepting the new block by updating their local ledger. The difficulty of finding the puzzle is adapted so that miners try to find a hash digest of the block header to be less than a "target" value. For example, hash ("block header" + nonce) = "00000xxxx", a hash digest that start with "00000". The "00000" is called the target, and the goal is to find a hash that is numerically less than the "target", so hash ("block header" + nonce) < "target". The nodes that intend to solve puzzle continuously make small changes on the block header mainly by manipulating nonce and continuously checking if the resulting hash is less than the "target". The process of hashing the block header makes it computationally difficult, and if the target changes, e.g., by adding a new "0", that makes it even more challenging for the nodes to solve the puzzle (see Appendix A.7.6, for detailed PoW algorithm for mining block). Miners need to spend time and allocate resources and computational power to solve the puzzle. Adjusting the difficulty (target)

[15] For example, a specific distributed computer system may tolerate up to 1/3 of "evil" computer nodes. The 1/3 indicated the **fault tolerance**.

[16] https://geeks.co.uk/2020/03/two-generals-problem/

[17] https://101blockchains.com/consensus-algorithms-blockchain/

also determines how often new block will appear on the BC; e.g., in the Bitcoin BC network, the new block appears approximately every 10 minutes. The advantage of PoW is that it remains fully decentralized and stable while the honest nodes (51% of processing power) guarantees the longest chain since this prevents the malicious nodes from rewriting blocks in the BC. The significant issues with PoW are energy inefficiency, less throughput, a long time for block creation (latency), and hardware dependency. Also, a considered potential issue is the "tragedy of commons" (Hardin, 2009) where miners will work only on their interests and enable "denial-of-service" attacks (Yaga et al., 2018; Bamakan et al., 2020; Antonopoulos, 2017; Yang and Shen, 2019; Gervais et al., 2016; Wang et al., 2018b; Leonardos et al., 2020).

- **Proof of Stake (PoS):**

  In the *Proof of Stake (PoS)*[18] consensus algorithm, nodes that intend to propose a new block must have stored some "stake", before being chosen to perform that work. The stake is usually stored in terms of cryptocurrency investment by sending a significant amount to a specific address. In PoS, the next block's creator is chosen in a quasi-random manner based on his stake and age, which can provide good scalability (Bamakan et al., 2020). In contrast to PoW, PoS offers high computational power, and miners in PoS should have a high stack of "cryptocurrency" to propose a block. The miner that proposes the block is rewarded by receiving the transaction fee. In the PoS consensus algorithm, the more stake the miner holds, the more chance to be selected for the block proposal. The advantages of PoS are faster block creation, which results in high throughput, scalability, and energy efficiency. There are two considered issues with PoS, since the more stake the miner has, the more probability to be selected for proposing a new block, which leads to centralization of BC. Also, the nodes that do not have any stake (known as "nothing at stake") (Bach et al., 2018), can freely misbehave and still be part of the network (Bamakan et al., 2020; Yang and Shen, 2019; Mingxiao et al., 2017; Leonardos et al., 2020; Bartoletti et al., 2017).

- **Delegated Proof of Stake (DPoS):**

  Delegated Proof of Stake (DPoS) is proposed as an improvement to PoS selection criteria (Bashir, 2017). An analogy for DPoS is the selection of board members in an enterprise. In the DPoS, the nodes vote on selecting a limited number of network representatives (delegate) to validate new blocks. These representatives should achieve consensus when proposing a new block. It is considered more democratic than PoS since each node votes on electing the representative for a new block proposal. The network nodes can vote on the replacement of the representatives if any of them is misbehaving. The limited number of representatives for block proposals improves scalability, low-cost transaction, and energy efficiency. However, it is considered semi-centralized (Bashir, 2017; Bach et al., 2018; Yang and Shen, 2019; Bamakan et al., 2020).

[18] https://en.bitcoin.it/wiki/Proof_of_Stake

- **Liquid Proof of Stake (LPoS):**

  The liquid proof of stake (LPoS) is considered a democratic version of DPoS, where delegation is not mandatory. For example, a miner (also known as a baker) needs to have a certain amount of tokens, e.g., currently, it must have 8k tokens to be qualified as a miner. Suppose the miner does not want to participate in consensus (not willing to spend for computing power or does not have a specified number of tokens, e.g., 8k, but still it has T tokens). In that case, the token holder (miner) is allowed to delegate the validation right to the other token holder randomly selected without transferring the ownership (so still keeping the T tokens). In LPoS, only the "right" to perform mining is delegated, not the ownership of tokens. The selected (delegated) miners will perform mining on their behalf, and received incentives can be shared between them (Arluck, 2018).

- **Proof of Elapsed Time (PoET):**

  In the Proof of Elapsed Time (PoET)[19], each participant on the BC network waits a random amount of time. In PoET, a secure hardware time source (Intel hardware called Secure Guard Execution) will randomly generate the waiting time. The network participants take the waiting time and go into idle mode (Ali et al., 2019). Once the network participant finishes the idle mode, it becomes a leader to propose the new block by solving a computational puzzle, similar to PoW. The main disadvantage is that PoET is proposed by Intel, and it depends on Intel devices; thus, it is classified as semi-decentralized (Ali et al., 2019; Wang et al., 2018a; Bamakan et al., 2020).

- **Proof of Burn (PoB):**

  The idea behind the PoB[20] consensus algorithm is to burn coins instead of using high computational power (spending energy and time) when mining the new block. Burning coins intend to prove that something difficult is done. Miners have to burn some of their own cryptocurrencies (coins), and the design principles of PoB indicate that the burned coins are sent to the "burning" address. That "burning" address does not have any private key, meaning that no one can use them, so these coins are permanently locked and therefore considered "burnt" or out of circulation. The miners that "burn" coins win the right to propose a new block in proportion to the burned coins. In exchange, they receive incentives in the native cryptocurrency (Tian, 2014). The advantage of PoB is that miners that spend some amount will probably stay in the network to gain profits later on, thus increasing the reliability and decentralization of the network. The disadvantage of PoB is that it makes it possible for the richest miners to get more frequent chances to mine the new block (Tian, 2014; Bamakan et al., 2020).

[19] https://www.investopedia.com/terms/p/proof-elapsed-time-cryptocurrency.asp
[20] https://github.com/slimcoin-project/slimcoin-project.github.io/blob/master/whitepaperSLM.pdf

- **Practical Byzantine Fault Tolerance (PBFT):**

  The PBFT consensus protocol aims to propose a consensus mechanism that reaches agreement with the system's nodes by tolerating byzantine fault (Castro and Liskov, 1999). It works by assuming that 1/3 of nodes are dishonest (faulty (F)), meaning that the network of nodes should consist on total N = 3F + 1, in order to tolerate F (thus F = (n-1)/3 faulty nodes (Sukhwani et al., 2017). In PBFT, all participants are considered validation nodes (VN), and one of them is chosen to create a block. When the transaction is received and validated by the VN, it is broadcasted to the other VN. After a particular time (or a specific number of transactions), the chosen leader node forms the block by strictly maintaining the transaction order based on the timestamp of receiving the transactions. Then for achieving the consensus, the leader node broadcasts the formed block to the other VNs. It requires 2f+1 nodes to agree on the new block of the transaction, and each VP individually executes all transaction, and add the new block in their private ledger (Sukhwani et al., 2017; Sousa et al., 2018; Sukhwani et al., 2018; Androulaki et al., 2018; Mingxiao et al., 2017).

- **Raft[21]:**

  Raft is an algorithm that is used to manage replicated state machines and logs. This type of consensus algorithm is suitable for building consortium BC, where members are known. In Raft, a leader is chosen to propose a new block. For allowing F nodes to be faulty, Raft needs 2F+1 nodes to be active in the network (Raft, 2012; Ongaro and Ousterhout, 2014; Baliga et al., 2018).

- **Proof of Authority (PoA):**

  The Proof of Authority (PoA) relies on a set of N nodes, known as "authorities". Based on the real-world documents, these "authorities" are identified within a BC, and an identification (id) is assigned to them. At N/2 +1, they are expected to be honest "authorities". The consensus in PoA is based on the mining rotation schema that intends to fair distribution of responsibility among the N authorities for block creation. The node's reputation is directly affected by its behavior. Suppose they act against the rules set up for the BC network (consortium), then their reputation decreases. Otherwise, as long as they act fairly, their reputation increases. That indicates the higher the reputation, the higher the chance of publishing a new block (Bentov et al., 2014; De Angelis et al., 2018; Yang and Shen, 2019).

- **Round Robin (RR):**

  The Round Robin (RR) consensus model is designed to support a permissioned BC network. In RR nodes, "wait" in turn to create a new block. Once the node has published the new block, it must wait for several blocks to be created before it can be chosen again for proposing a new block. The algorithms include a time limit, and in case the publishing node is not available to publish a block in its turn (not proposing a new

[21] https://raft.github.io/

block in a certain time), the other available node might continue proposing a new block. In such a case, the algorithm avoids any stop in the production of new blocks. In the RR consensus algorithm, none of the nodes are able to create the majority of blocks, and it avoids any computational puzzle. (Ahmed-Rengers and Kostiainen, 2020; Yaga et al., 2018; Yu et al., 2018)

- **Tendermint[22]:**

  In Tendermint, the consensus is achieved based on the voting process. For being able to propose and validate a new block, a committee of nodes is selected. From that community, a particular node is selected to propose a new block. The block is validated (mined) with two-thirds of the votes from the committee (Natoli et al., 2019). To avoid any adverse situations such as double voting from community nodes, it uses a locking mechanism (Natoli et al., 2019; Amoussou-Guenou et al., 2019).

3.4.3 Overview: The Current Blockchain Platforms

Currently, there are many BC platforms designed and developed for different purposes. In this section, we present some of the most influential BC platforms. These BC platforms are among the most prominent, however many other BC platforms are emerging based on their philosophy, e.g., new cryptocurrency or many use cases from different domains are being developed on top of these BC platforms.

- **Bitcoin:**

  Bitcoin is the first and most prominent cryptocurrency-based BC technology. (Nakamoto, 2009) proposed BC as a peer-to-peer payment method based on cryptocurrency named Bitcoin[23]. Bitcoin is a decentralized digital currency that allows transferring assets (cryptocurrency) directly through the internet instead of trusted third-party platforms such as banks. The Bitcoin mechanism works in a decentralized way, meaning that no single user (user, organization, or governmental office) controls the digital currency. The decentralized mechanism of Bitcoin does not allow a single point of failure since Bitcoin nodes (miners) are distributed geographically, and each contains the full copy of the ledger. The Bitcoin users share the same ledger, and transactions are validated and executed directly between users. Bitcoin allows anonymous users, it supports one type of accounts (user accounts), and accounts are not anonymous. Transactions are publicly visible (accessible) and organized based on the Merkle Tree approach (Yaga et al., 2018; Antonopoulos, 2017). The ledger is shared, append-only, and maintained by distributed-decentralized miners that collect transactions and propose a new block. The agreement on the ledger's shared state is achieved based on the Proof-of-Work (PoW) consensus protocol (see 3.4.2.1), and proposing a new block by a miner is possible only when solving the PoW computational puzzle. Bitcoin uses cryptographic hashes for securing ledger from tampering. In early 2009, Bitcoin mined the first (genesis) block[24],

[22] https://github.com/tendermint/tendermint
[23] https://bitcoin.org/en/
[24] The first mined block in the BC is shown here.

and since then, the BC has continued to grow. Today there are thousands of developed cryptocurrencies based on the Bitcoin technology[25].

- **Ethereum:**

  Ethereum is a BC that supports the definition and execution of the SC. Ethereum is built on top of Ethereum Virtual Machines (EVM), enabling the SC (SC) execution. SC is written in Turing-complete EVM bytecode (Wood, 2014; Buterin, 2017; Atzei et al., 2017). Ethereum offers a programming language called Solidity[26] for writing SC. SC can transfer/receive assets (Ether[27]) to a user or other SC. The possibility of writing SC allows users to determine their own rules of ownership and to transfer and manage their assets (Ether) (Buterin, 2017). Two types of accounts are present on Ethereum: user account and SC account. In Ethereum, transactions are executed to send *Ether* to a user or SC, create new SC (deploy new SC to Ethereum BC), or invoke a function of another SC (Wood, 2014; Atzei et al., 2017). The Ethereum network is composed of decentralized EVM that share the same ledger. After successfully executing the SC and confirmed transaction (mining the new block), the state of the ledger changes (Buterin, 2017). With the emergence of the Ethereum BC, it is considered the second generation of BC as new decentralized applications (Dapps) emerge to solve different domain problems, e.g., finance, insurance, healthcare, and notary.

- **Quorum:**

  Quorum[28] BC enables building business application for enterprises. It is based on Ethereum and is intended to allow enterprises to develop an Enterprise Ethereum Client and use BC benefits. It offers additional features for the enterprises such as transaction privacy, pluggable consensus (Raft, PoA or Istanbul BFT) (Baliga et al., 2018) based on the use case, and "access control" for participants and network nodes. At the highest level, Quorum is a public Ethereum BC, and it uses an advanced component called private transaction manager which enhances privacy by enabling off-chain communication of nodes based on HTTPS protocol (Baliga et al., 2018).

- **Corda:**

  Corda[29] is a permissioned BC that allows financial sector businesses to exchange transactions in a strict privacy maintained environment. The participants in the Corda ecosystem comprise the network. The network is presented as a fully connected graph of nodes, and communication occurs only on a point-to-point basis. The connected nodes can possibly communicate with each other based on the "need" to share facts[30]. The ledger presents evidence on how facts are shared between nodes. It can be visualized as the intersection between sets, as displayed in Figure 3.5. In the ledger presented

[25]The list of current cryptocurrencies: `https://coinmarketcap.com/1/`
[26]Solidity programming language: `https://docs.soliditylang.org/en/v0.7.5/`
[27]Ether is the digital currency of the Ethereum: `https://ethereum.org/en/eth/`
[28]https://consensys.net/quorum/
[29]Corda key concepts: `https://docs.corda.net/docs/corda-os/4.6/key-concepts.html`
[30]"Fact" may be a payment request.

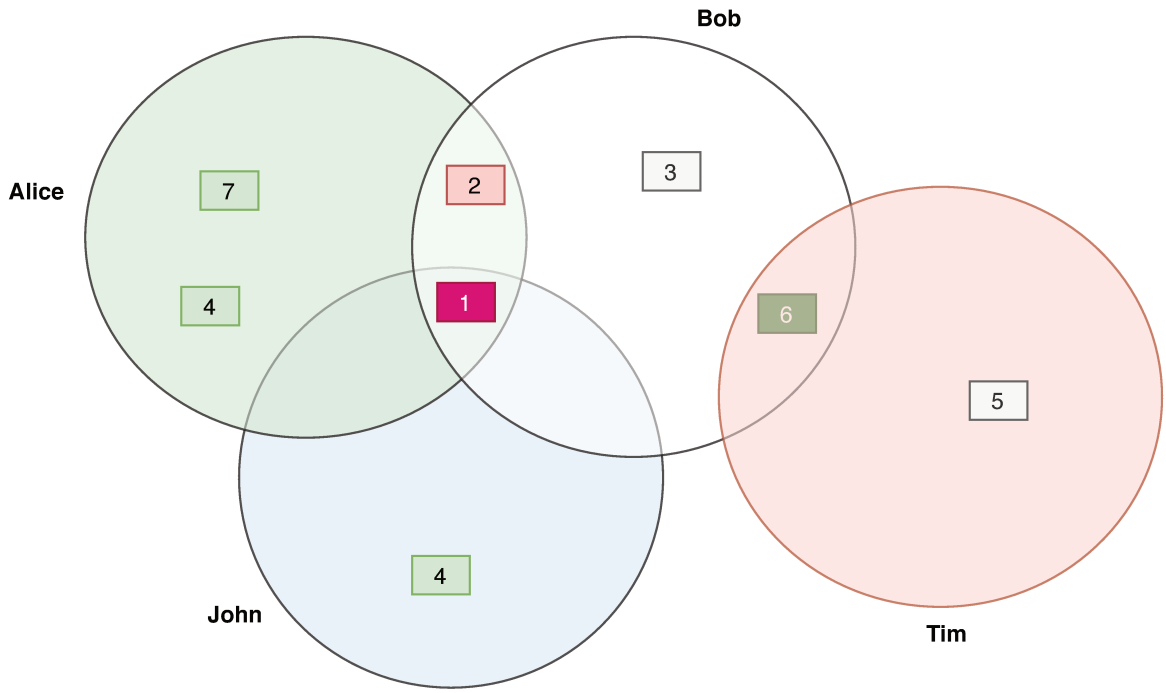

FIGURE 3.5: The Corda blockchain ledger representation.

in Figure 3.5, Alice and Bob share the fact "2", and Alice, Bob, and John share the fact "1". Bob and Tim share fact "6", while John and Tim and Tim and Alice do not share any fact. Only the intersected facts are shared since the nodes keep the other facts on their ledger, e.g., Alice does not share facts "7" and "4" with other network participants. Corda uses "states" to maintain immutable information about facts. Once the state is created, it cannot change, but it can be replaced (and the old state remains a historical state) when the facts are changing over time, e.g., the fact may be paid. In Corda, transactions are not globally broadcast, and it does not use a gossip protocol for the network; thus, the recipient of the transaction should be specified. The node decides with whom it will communicate. The transaction is known as the proposal for changing the ledger, and they are committed if they do not contain double-spending, are contractually valid, and are signed by the involved parties. The agreement on the ledger's shared state is achieved based on a consensus mechanism, applied in two phases: *Validity consensus* and *Uniqueness consensus*. The *Validity consensus* is the process of verification whether the current conditions (transaction is digitally signed by all parties and transaction is contractually valid) holds for the current and the entire transaction chain, which generates the input for the proposed transaction. The *Uniqueness consensus* intends to avoid the double-spending issues by verifying that the input in the transaction is not consumed already in other transactions (Brown et al., 2016; Hearn and Brown, 2019).

- **Tezos:**

  Tezos is a public open-source BC that allows the deployment of decentralized applications. Among the main features of the Tezos is the "on-chain" governance (known as

self-amending), which enables upgrading the protocol and adapting smoothly newly proposed technological features, thus avoiding hard forks in the network. The "on-chain" governance enables proposing, selecting, testing, and activating the protocol upgrades. The network stakeholders (known as bakers) may participate in proposing or voting for upgradable changes in the Tezos protocol. It requires four different stages to accept amendment changes on the protocol: proposal, exploration, testing, and promotion periods. This type of governance ensures long-term upgradability and stability for the Tezos protocol. For the development of the SC, Tezos proposes a native language, i.e., Michelson, which enables formal verification of SC, thus avoiding costly bugs. In terms of consensus, Tezos use LPoS, thus allowing stakeholders to validate (Goodman, 2014; WikiTezos, 2021).

- **Ripple:**

  The Ripple protocol is a BC-based solution for "real-time gross settlement, currency exchange and remittance network"[31]. Ripple is proposed as a multi-party transaction settlement solution, enabling cheaper currency exchange, and is intended for use by banks and payment providers. The Ripple network is called *Ripple Net*, and it consists of distributed decentralized nodes known as validators. Validators maintain the shared ledger. Transactions are executed based on the *Ripple Transaction Protocol (RTXP)*, and validators check and verify if the executed transaction follows the RTXP rules. The *Ripple Net* is accessed through a gateway (considered as entry points), and any payment company or bank can run a validator. The Ripple consensus mechanism indicates that each node maintains a unique node list (UNL) of the identities of subset trusted nodes in the network. Several rounds are performed to achieve consensus, thus, round *i)* transaction collection and making them public in the form of a list, called "candidate set"; *ii)* broadcast "candidate set" transaction in UNL; *iii)* each validator validates transactions and votes on the truthfulness of these transactions; *iv)* transactions that receive a minimum percentage of "yes" (otherwise discarded or set in the first consensus round) are passed in the next round; *v)* and finally, it requires a minimum 80% of UNL agreement on the transaction, and all transactions that fulfill this criterion are added to the ledger (Chase and MacBrough, 2018; Kuo et al., 2019).

- **MultiChain[32]:**

  MultiChain allows for the rapid deployment of BC-based solutions. It belongs to the permissioned BC platforms, and is intended to ensure BC activity is visible only to specific participants. It provides ledger features by improving user permission (Kuo et al., 2019). It allows configuration and concurrently running of multiple blockchains in the same node. The node connectivity is achieved by fulfilling the "hand-shaking" protocol. Nodes are identified and an associated list of permissions is assigned to each node. MultiChain propose features to control the transaction by determining which transaction are permitted. MultiChain proposes a mining solution that avoids

[31] https://ripple.com/
[32] https://www.multichain.com/download/MultiChain-White-Paper.pdf

manipulation of the mining process. It restricts mining by constraining the number of blocks that the same miner may create within a given window (Kuo et al., 2019).

- **Polkadot:**

  Polkadot presents a heterogeneous BC-related solution that intends to allows users to operate their own BC. The initiative is on researching, definition, and developing the network of BCs. The core idea is to establish a protocol that enables the interoperability of BCs. It presents a scalable heterogeneous multi-chain network protocol that operates in a permissionless environment. In-depth, Polkadot operates two types of BCs, the main networks where transactions are permanently stored and a user-created network called parachain. The parachain is customized from users and laterally maintained on the main chains, thus benefiting the same security as the main chain (Team, 2017; Polkadot, 2017).

- **Hyperledger Fabric (HF):**

  Hyperledger[33] Fabric (HF) is a BC-based framework that provides the technological features for developing a consortium or private BC. HF is an open-source framework implemented in GoLang programming language, and it is supported by several tools such as Hyperledger Explorer and Hyperledger Composer[34], simplifying the business logic over HF. HF has a modular and configurable architecture that allows users to adopt BC technology for their use case. Furthermore, it allows smart contracts (SC) to be written in general-purpose programming languages, e.g., Go, Java, Node.js and Python, which is beyond the domain-specific language provided by other SC-enabled BC platforms (Media, 2019; Blog, 2020; Androulaki et al., 2018; Cachin, 2016).

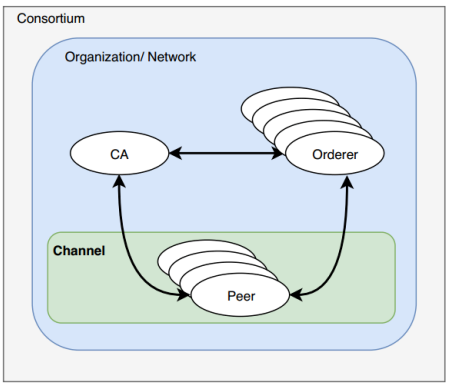

FIGURE 3.6: The overview of the key concepts of HF.

The main components of Hyperledger Fabric (HF)

[33]Hyperledger is a consortium of different research and development communities that are gathered (under the Linux Foundations) to contribute to many projects related to BC. Hyperledger provides open-source BC frameworks, tools, documentation, practical experiments, with a specific focus on business-oriented use cases (Hyperledger.org, 2016).

[34]Hyperledger Composer has been deprecated. A similar-intention tool to Hyperledger Composer called Hyperledger Convector is currently provided.

The HF network is composed of nodes that are connected together in a peer-to-peer fashion. HF has different types of nodes, such as *Peer, Orderer, Certification Authority, Smart Contract, Applications*.

- **Peer** node (peers) is one of the HF nodes. The BC network primary is comprised of a set of peers. Peer hosts one or more instances of the ledger and SCs (Media, 2019).
- **Orderer** node is used to ensure the consensus of the HF. The order's role is to keep the peer's ledgers consistent (Media, 2019; Blog, 2020).
- **Certification Authority (CA)** nodes ensure identity delivery via digital certificates, typically required by each organization to enroll new members (Media, 2019; Blog, 2020).
- **Smart Contract (SC)** is a piece of code written in a specific programming language, e.g., GO, JavaScript, whose purpose is to query and (or) update ledger (Media, 2019; Blog, 2020).
- **Applications (Client)** nodes can connect to and interact with peers deployed over the network (Media, 2019; Blog, 2020) .

HF can be managed by several organizations, that constitute a consortium. Thus, each organization is responsible for managing its own nodes, and it is mandatory to have at least one Certification Authority (CA) node (Orderer node). Figure 3.6, shows the interactions between HF components (Peer, Ordered, CA) in an organized consortium.

Channels: Private Sub-Networks

Channels provide a private communication link between peers. That is a way to separate the network into a private sub-network composed of a subset of members/peers. Communications onto each channel are ciphered and controlled by Orderer nodes and CA nodes. Because the network is private and permissioned, every action applied by organizations over the network must be done through a specific channel with the right permissions and credentials. The SC must be installed over a channel, which leads to installing it on each peer belonging to that channel (Media, 2019; Blog, 2020).

Performance analysis for HF

From a performance point of view, in general, BC technology is not the most suitable technology, especially when public BCs such as Ethereum or Bitcoin are applied for a particular use case. There are several gaps in transaction throughput (number of transactions per second (tps) and latency in confirming a new block on the BC). Unlike public BCs, private and consortium BCs are much better in terms of performance. For example, HF allows for adding some basic configuration, such as choosing the block size (or block time), which impacts the transaction throughput and latency (Sukhwani et al., 2018). For example, depending on the block size (e.g., 2 MB), Local Area Network (LAN) properties, and storage (SSD vs. HDD), HF can support high transaction throughput on the order of several thousand (approx. 3000 tps) with latency in milliseconds (Androulaki et al., 2018).

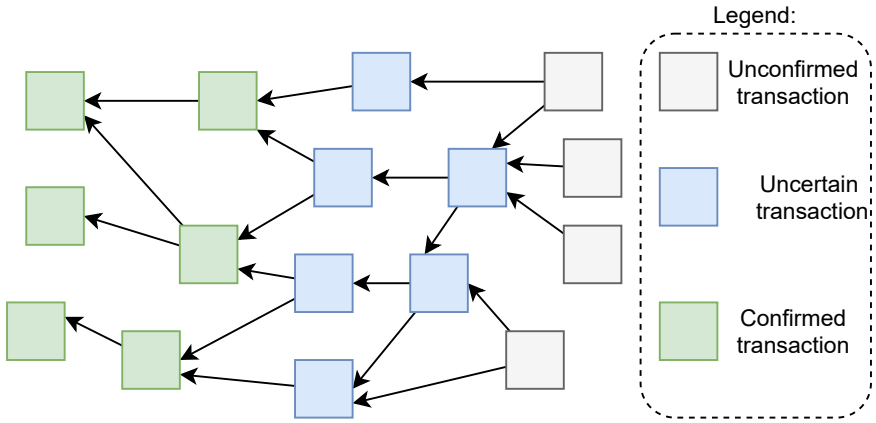

FIGURE 3.7: The schema for DL Tangle. Inspired from (IOTABlog, 2018).

- **IOTA:**

  The ledger, named Tangle (IOTA), records transactions performed by nodes in the network. The network of IOTA uses directed acyclic graph (DAG) (VanderWeele and Robins, 2007) to store transactions (IOTABlog, 2018). The directed means all the data are attached in the same direction, meaning that they avoid forming any loop since they follow "acyclic" reference on the "graph", e.g., tree (IOTABlog, 2018). In IOTA, nodes store only one single transaction. The transaction is signed with the private keys. Then the transaction is deployed as unconfirmed, which further needs verification. For verifying (appending) that transaction, the node needs to verify two previous (unconfirmed) transactions. This selection is seen as attestation, which is selected by using the "random walk" (Markov Chain Monte Carlo) algorithm. So, with the transaction, attest directly (blue nodes towards green nodes) or indirectly (white nodes towards green nodes), as showed in Figure 3.7. The nodes that are attempting to verify (append) a transaction must spend some computational power (cryptographic puzzle) in order to validate the transaction (and to avoid Sybil attack and spamming)[35].

3.4.4 Current Challenges for Blockchain

BC's advanced technological features disrupt the way business processes are designed, organized, and executed. However, there are considerable challenges for BC at the technological and design levels to integrate it into different domains. In general, permissionless BCs suffer from privacy and scalability issues. Permissionless BC does not offer any privileged users based on roles and permission. Further, anyone can join the network and receive a full copy of the ledger, and they can explore any transaction, giving any related transaction (balances, transaction value) related to the user public key. Furthermore, in (Biryukov et al., 2014) marks the possibility of exposing the IP address of the anonymous sender or receivers of the transaction, thus compromising privacy (Xu et al., 2017; Beck et al., 2016; Atzori, 2017).

[35] What is needed to issue a transaction? (IOTABlog, 2018).

The scalability issues refer to the number of transactions processed in seconds (TPS) by the BC framework. The technological capability of Bitcoin and Ethereum BC enables a limited number of transactions per second, e.g., up to 7 TPS for Bitcoin and up to 20 TPS for Ethereum (Xu et al., 2017; Beck et al., 2016). Scalability in the permissionless BC is impacted by the size of data in BC, the transaction processing rate, and the latency in transaction transmission (up to 10 minutes in Bitcoin) (Xu et al., 2017; Scherer, 2017).

On the other hand, permissioned BCs offer better management of privacy and scalability issues. Consequently, they lose some decentralization level since an administration (user or group of users) imposes rules for such BC. Each permissioned BC proposes its own way of managing privacy. For example, Hyperledger Fabric, besides membership service provider (MSP)[36] for network participant certification, proposes channels for managing private communication between nodes in the network.

In terms of security, BC is not a silver bullet that solves any issues. The transaction stored in Bitcoin and Ethereum is considered stable (immutable) as long as the processing power of honest nodes is higher than 51%. That prevents malicious nodes from rewriting transactions (Ye et al., 2018; Lin and Liao, 2017). If the processing power of the malicious nodes is higher than 51%, then the transaction immutability is not guaranteed since the chain of the transaction might be rewritten by a malicious user (Ye et al., 2018; Lin and Liao, 2017).

Another threat is the Sybil attack[37]. To avoid Sybil attacks, the Bitcoin BC requires nodes (miners) to prove the "work" they have done before proposing a new block. In the permissioned BC, based on the PBFT consensus algorithm, the nodes are certified before joining the network and proposing a new block. That also helps to prevent Sybil attacks.

[36] https://hyperledger-fabric.readthedocs.io/en/release-2.2/msp.html

[37] A Sybil attack presents a security threat in large scale peer-to-peer system, where a user presents many identities in order to take control of the system or fraction of system (Douceur, 2002).

Blockchain Platform	**Description of Platform**	**Crypto Coin**	**Mining Incentives**	**Consensus Mechanism**	**Privacy**	**Smart Contracts**	**Network**	**Latency (block time)**	**Throughput (Tx. per second(tps))**	**Fault Tolerated**
Bitcoin	Crypto platform (P2P) money	Bitcoin	Yes	Proof of Work	Open BC; Privacy issues	Miniscript, Script	Open	approx. 10 min/block	7 tps	51% processing power
Ethereum	Platform for developing decentralized application	Ether	Yes	Proof of Work	Open BC; Privacy issues	Solidity-based SC platform	Open	approx. 20 seconds/block	approx. 20 tps	51% processing power
Hyperledger Fabric	General purpose BC platform which enables development of enterprise BC solution	No	No	Enables pugable consensus algorithm: PBFT	Channels	Different programming language Go, Java, JavaScript	Consortium	less than 1 sec/block	approx. 3000 tps	1/3 nodes
Tezos	Crypto Platform	XTZ	Yes	LPoS	Open BC	Michelson, Smartpy	Open	approx. 15 sec/block	approx. 40 tps	1/3 of nodes
Ripple	Finance Industry	XRP	No	PBFT, Raft	Open BC	No	Open	approx. 4 sec/block	approx. 1500 tps	20% of nodes
Corda	Finance Industry	No	No	PBFT-based, Notaries nodes	Private ledger	Cotlin, Java	Consortium	//	//	1/3 nodes
Quorum	To develop an enterprise ethereum client	No	No	Raft-based	Transaction manager	Solidity based SC	Consortium Private	milliseconds	approx. 750 tps based on network configuration	1/3 nodes

TABLE 3.1: The summary of main characteristics of the blockchain platforms.

3.4.5 Comparison of Blockchain Platforms

In this section, we present a comparison of the main BC platforms. This detailed schema includes an extensive comparison of existing BC platforms, inspired by research in (Kuo et al., 2019; Clincy and Shahriar, 2019; Polge et al., 2020; Macdonald et al., 2017; Atzori, 2017). Table 3.1 shows the technological features of the BC platforms.

3.5 Smart Contract (SC)

This section presents the main characteristics and formal definition of SC. The emerging of BC, e.g., Bitcoin, the primary focus was on the digital currency, i.e., cryptocurrencies, which was also considered the first epoch of the BC (Nakamoto, 2009). The second epoch of BC technology is considered with the emerging a decentralized autonomous computer program described as "smart contract (SC)" that runs on the BC. With the SC emerging as a decentralized software agent, academia and industry researchers had the opportunity to apply BC in several cryptocurrency-free use cases. Since then, based on the technological characteristics, BC and SC have been widely used to solve well-known research and industries problems in SpC (Tribis et al., 2018), healthcare (Hölbl et al., 2018), insurance (Raikwar et al., 2018), finance (Treleaven et al., 2017) and many more.

3.5.1 Smart Contract Definition, Semantics and the Main Characteristics

Smart contract (SC) is an autonomous computer programming code that runs on the BC and is executed when a specific event happens, based on specified parameters (Buterin, 2017). A SC deployed on the BC is assigned a unique address that identifies it. BC users can invoke the SC by sending a transaction to the SC address (Buterin, 2017; Luu et al., 2016). SC logic is mainly based on domain-specific. It encodes any set of rules emerging from the source of the SC into the programming language (Mik, 2017). The SC source can be a natural language law, scope of any agreement between parties, and other possible sources depending on the business process requirements (Mik, 2017; Kolluri et al., 2018; Luu et al., 2016). For the transaction that is accepted on the BC, and if it contains the SC address as a message received, the miners will execute the SC code and react according to the SC's specific tasks. A SC is a self-executed program; moreover, it can invoke another SC, call external service (oracles[38]), fulfill given tasks, and implement and automate a wide range of domain-specific applications (Luu et al., 2016; Buterin, 2017). The main design principles of the SC[39] includes (Luu et al., 2016):

- *SC address (ID)* - a hash value that identifies the SC on the BC.
- *Owner ID* - is a *160 bit* hash value, which indicates the owner of the SC.
- *Immutable* - once deployed into BC, the SC remains immutable.
- *Internal storage* - the SC has its own private storage, holds its execution code with pre-defined parameters, and some amount of virtual currency (own balance of the SC).
- *Execution costs* - for the BC platforms that need digital coins to perform transaction mining (incentives), the execution of the SC has its cost.
- *Enforcement* - the contractual obligation expressed in terms of SC, e.g., transfer assets once the goods are received, are automatically performed as contractual obligations and enforced by SC.

[38]Oracle for modern DApps: `https://provable.xyz/`

[39]Ethereum-based SC design principles. Similarly, the other BC platforms follow the same principles with a slight difference in executing SC and maintenance SC lifecycle.

- *Invoke another SC* - by sending a transaction to the SC address.
- *Autonomous* - the pre-defined parameter on the SC code allows changing the BC state if executed successfully.
- *Self-Execution* - the SC is a set of autonomous executable agents that are triggered by pre-defined parameters in the SC code or executed from environmental parameters.
- *Event/Method* - SC functions (methods) are a set of instructions that are executed in the SC for completing the intended task.

Users invoke SC by sending the transaction to the SC address, e.g., a transaction involving an amount and parameters to execute the targeted SC (Kolluri et al., 2018; Mik, 2017; Luu et al., 2016). Furthermore, one SC can call another SC synchronously (on-chain) as well as asynchronously (off-chain) (Kolluri et al., 2018). SC supports an ordered logic that follows the ordering rules *if this then that (IFTTT)*. This semantic is called event ordering logic (or order-execute), and SC events (function invocation) are executed as they are allocated in SC. Following this logic, if the current order passes, the SC continues processing the next order; otherwise, the SC throws an exception (Kolluri et al., 2018). This logic is present in almost all blockchains that enable deploying SC. For example, Ethereum implements SC using Solidity, a domain-specific programming language. Contrary to that, Hyperledger Fabric (HF) follows execute-order-validate [40] logic for developing SC. The execute allows execution and checks for its correctness of the transaction, further the transactions are ordered (by consensus algorithm) and validate transaction based on the endorsement policy of the application (Fabric-v.2.2, 2021).

From a technical point of view, BC can be seen as a "state transition system". Generally, the "state" presents the current information regarding the account and the SC on the BC, and the "transition" function applies a transaction on the current state, which results in a "new" state as presented in 3.1. The new state becomes the BC's currents state only after a successful transaction (without any error).

If the BC state is noted by γ, for any successful transaction executed by the SC, the BC state will be updated into γ' (Buterin, 2017; Luu et al., 2016):

$$\gamma \xrightarrow{Tx} \gamma' \tag{3.1}$$

The new state γ' may impact many user accounts or other SC, impacting the BC's empirical data. The transaction Tx in the BC network, for example, in Ethereum, is performed to create the SC, invoking other SC by calling its functions, or transferring "Ether" or other assets (Atzei et al., 2017). The BC network then agrees on the new state[41] of the BC (γ'), impacted by the SC. Otherwise, the transaction might be refused as an unaccepted transaction by rolling back the BC state (Buterin, 2017; Luu et al., 2016; Kolluri et al., 2018).

[40] Hyperledger Fabric Documentation: https://buildmedia.readthedocs.org/media/pdf/hyperledger-fabric/latest/hyperledger-fabric.pdf

[41] Besides the operations performed by SC, the BC state might be updated from other operations, e.g., directly sending digital currency, updating particular variables on the BC, and other possible operations.

If ρ presents the SC address on the BC, the global state of the SC is $Q = \gamma\ [\rho]$. The SC state presents its private storage and the balance of the SC (in Ether, or any other cryptocurrency). If ϕ presents the user account on the BC, the invocation of the SC is performed by sending transaction (Tx) on the ρ, thus,

$$\phi \xrightarrow{Tx} \rho$$

In such a situation, the transaction (Tx) represents an input event (Σ) for the SC (ρ), and it is mainly composed of:

Tx = $(sender \rightarrow id, receiver \rightarrow id, value \rightarrow N, function\_name \rightarrow name, other\_data)$ (Kolluri et al., 2018)

In general an SC instance is composed of three main elements $SC =< \rho, CODE, Q >$. For a publicly available SC, meaning that the SC's function is accessible from the other SC, the SC invocation is performed by an invocation of the *id* of the function of the SC (Kolluri et al., 2018; Buterin, 2017; Luu et al., 2016).

The mathematical model for SC presents a state-transition system composed of a quintuple set of elements (Bai et al., 2018; Luu et al., 2016)[42]:

$$M = (Q, \Sigma, \Delta, S_0, F), \text{ where,}$$

- Q is the finite set of all possible states of the SC;
- Σ is the set of all input events on the SC;
- Δ is the set of transition-function of the SC, Δ: Q x $\Sigma \rightarrow Q$;
- S_0 is the initial state of SC, $S_0 \in$ Q;
- F the final state of the SC, $F \in$ Q;

3.6 Synthesis: Comparison of the Technological Features for Centralized, Distributed, and Distributed-Decentralized Approaches

We compare the BC technological features versus the centralized and distributed approach. Table 3.2, shows the result of this comparison based on the state-of-the-art contributions (McFarland and Nicholson, 2007; Yaga et al., 2018; Ali et al., 2019; Wang et al., 2018a; Antonopoulos, 2017; Dillon et al., 2010).

[42]Formal Definition of a Finite Automaton: https://www3.cs.stonybrook.edu/~cse350/slides/automata2.pdf

TABLE 3.2: Comparison of technological features: centralized, distributed and decentralized.

Centralized Approach	**Distributed Cloud Computing/IoT frameworks**	**Distributed-Decentralized (Blockchain)**
Data Lost. The centralized approach user should trust the centralized authority that their data are properly backup and safely stored	Data Lost and Destroy. Cloud providers usually commit to providing backup data centers, but still, the user needs to trust that the data is backup properly and safely stored	Distributed by design, the same copy of the ledger is distributed on many nodes, and any user can maintain its copy of the ledger. The new nodes that join the network receive the full copy of the ledger
Deployed in specific geographic locations, in a specific country, organization's premises, e.g., bank	Located in several geographic locations. Distributed data centers for storing information	BC nodes are geographically distributed around the world and communicate (information exchange) using a peer-to-peer protocol
Single point of failure. The centralized approach is prone to failure as long as it relies on a single point (central database)	"Availability" is one of the main issues. CP provides backup points that may support the user when a specific node (data center) fails. There is a Service Level Agreement (SLA) between the cloud provider and cloud consumer to guarantee cloud services availability. The specification of the SLA itself imposes several issues as well	The decentralization-distribution and peer-to-peer mode of communication enable BC network availability and resilience. Even if a single BC node remains available, the new nodes joining the network will receive the full copy of the ledger, thus forming a sustainable network
Technology adaptation (homogeneous network) based on a defined centralized solution. Any new potential user that intends to join the "solution" provided by centralized authority needs to adapt its technology, software, network, and hardware components	Depending on the needs of the cloud consumer, it may rely on the technology and software infrastructure provided by cloud providers. CP offers a range of technologies (PaaS), IT- infrastructure (IaaS), and software services (SaaS)	BC supports various types of network infrastructure, software, and hardware. BC is accessible from different devices such as servers, workstations, PCs, laptops, and mobile phones. These devices might have different software platforms,different network configurations, and hardware specifications
Transaction validation. In a centralized approach, the user must trust the centralized authority that all the transactions are valid and malicious users are denied	Depends on applications purpose and SLA agreements for transactions validation and processing	For transaction validation, BC technology uses the current state of the BC to verify the transaction, and if malicious transactions are detected, they are immediately ignored
Data Integrity - Altering. In the centralized approach, the user needs to trust the central authority that the data integrity is guaranteed. Also, data might be altered in the centralized approach, and the user only needs to trust the central authority that the previous data are not altered	There is no formal proof that guarantee data integrity on the cloud. CP provides usefully offers, sort of SLA to fulfill, based on which CP guarantees data integrity. Still, at the formal and practical level, it is possible to alter data stored in the cloud	BC uses advanced computer science mechanisms such as hash function and digital signatures to provide data integrity and authenticity. The transaction data stored on the BC are considered tamper-proof
In a centralized approach, users must trust that the best security practices and standards are implemented	CP is considered to have major issues with security and privacy. In CP, the user needs to trust (based on SLA) that the cloud providers have implemented and maintained security standards	BC distributed nature increases its strength against possible attacks since there is no central point that an attacker may target. Based on that and the resistance of the honest nodes, BC is not affected by security attacks. Any security attack on a single node may affect that particular node but not the entire BC

3.7 Advanced Concepts: Digital Twins

The current SpC are dynamic and intense in the sense that they generate many interactions between the involved stakeholders. The complexity rises when the number of operations increases, which occurs when more stakeholders are involved. The current SpC operation systems have limited capacities. This decreases the value of the SpC since efficiency, trust, reliability, and transparency are continuously threatened (Christopher, 2011).

To avoid such a scenario, prior studies of the SpC systems recommend verifying these drawback scenarios at an early stage and continuously improving them. One promising direction is to simulate such a system in a virtual world that completely models real-world system artifacts workflow before development in order to achieve this objective. The current studies highlight the need to use simulation to extract, test, and improve the operation in SpC.

The concept of "Digital Twin (DG)" has been introduced to allow the design of such virtual systems as a reflection (twin) on real-world systems. Initially presented in 2003 at the University of Michigan by Grieves (Grieves, 2014), the concept of DT has been explored since, and several definitions are presented by (Hochhalter et al., 2014; Glaessgen and Stargel, 2012; Reifsnider and Majumdar, 2013). A DT presents a probabilistic, multi-physics, and multi-scale simulation for a complex product or processes and uses the specific connection, e.g., Internet of Things (IoT) devices and communication channels, to mirror its real-world twin in the virtual world (Glaessgen and Stargel, 2012; Tao et al., 2018). In its depth, DT is composed of three main parts, i) physical part, ii) virtual part, and iii) connection, which provides data and information that ties physical and virtual parts. This indicates there are two spaces in DT, i.e., physical space, which presents the real-world process, product, or any "targeted object", and virtual space, which reflects the real-world (Grieves, 2014; Tao et al., 2018; Tao et al., 2019). This reflection provides a system digital equivalent to the real-world system (Grieves, 2014). The research in (Putz et al., 2021) shows a blockchain-based solution for secure information management. It uses DG for managing information for Industry 4.0. The solution offers information confidentiality, access control, and availability based on blockchain.

3.8 Conclusion

This chapter presents an extensive study of existing technologies that could leverage SpC. To better understand the existing technologies, we examined their main characteristics and further highlighted the opportunities and challenges in supporting future use cases. Since this thesis mainly focuses on BC technology, we studied it in-depth and showed its technological features as an opportunity to support the specific use cases. We studied and described the consensus mechanism, its importance, and the principal characteristics of numerous consensus algorithms currently most commonly used in different BC platforms. We presented a definition, a detailed description, and a formal representation of the SC.

In general, BC enables distributed ownership (distributed ledger and geographically distribution of physical nodes) of the ledger by network participants, thus avoiding complete ownership by a single authority (centralized approach). There is an extensive list of BC platforms, and essentially, the use case specification determines the BC platform's selection. Accordingly, the use case determines the level of security and privacy required, technical requirements (network deployment, efficiency, scalability) to support the use case. For this, we have deeply investigated options and the possibility to facilitate selecting the right BC based on the use case requirements. We designed and developed a *method for improving BC application* showed in (Imeri et al., 2019d).

BC is considered a disruptive technology, and many domains such as SpC and logistics, healthcare, insurance, finance are researching to apply BC to solve many research problems. The following chapter presents an extensive study of BC applicability in SCM, its integration with other technologies such as the Internet of Things, and SC formal specification and verification.

Chapter 4

State-of-the-art: Blockchain Involvement in Supply Chain and Logistics, and Blockchain-based System Design Approaches

4.1 Introduction

Since the advent, blockchain (BC) has found its application in numerous industries, including SpC and Logistics. BC is disrupting the current design and development of the new software system applications for SpC and Logistics. Nowadays, for business providers, the efficiency of the service they provide is crucial for the long-term sustainability and improvement of their operations. This efficiency depends on consumer satisfaction. That includes service delivery and reliable information related to goods, correct delivery, and timeliness. These providers are strongly dependent on the application and models they use to plan and manage their daily activities. In this context, information sharing is crucial to ensure a reliable and efficient way of collaborating with SpC stakeholders.

This chapter aims to survey the current range of academic literature and industrial applications from the business perspective related to BC technology's applicability in SCM and logistics. BC has been the subject of study for possible integration with other technologies such as the Internet of Things (IoT). We investigate the existing approaches related to BC and IoT. BC technology comes with the advanced features of self-execution computer programs, i.e., smart contracts (SC); therefore, several kinds of research have investigated designing system services based on BPMN (A.5.2.2) and deploying SC for blockchain-based applications. We investigated these researches and highlighting possible advantages. Designing a SC is challenging and, at a certain level, some design patterns are proposed when alterations are required on the existing SC. We present related works on design patterns for SC. In general, the SC implements some business logic, holds enormous amounts of money in terms of cryptocurrency, and intends to fulfill specific tasks required from the SC owner. For being sure that the SC behaves as intended, some researches address formal specification and verification of the smart contract. For composing this chapter, we define research method (A.1), which allowed us a systematic literature review.

4.2 Blockchain and Supply Chain Management (SCM)

This section explores the state of the art research activities from scholars and industries related to BC technology as a potential technology to improve SCM and Logistics operation.

4.2.1 The Value of Blockchain in Supply Chain Management and Logistics

BC technology enables a new way of designing and developing decentralized applications, which will eventually improve the quality of services in the SCM and Logistics. Information security, record keeping properties, and avoiding third-party involvement show significant potential for overcoming SCM and Logistics's main issues. The considered values brought by BC in SCM and Logistics are in the field of compliance, transparency, consumer satisfaction, trust and real-time response (Feuchtwanger, 2017).

Compliance issues: The logistics and transportation processes are governed by national, international, and internal business agreements. For a regular logistics and transport process, the compliance standards' enforcement is established by the usability of the SC. The combination of the BC and SC provides real-time visibility (immutable transaction and transparency) in SpC. That gives the potential to ensure that all contract conditions are fulfilled. Further, it enforces organization to work in compliance with the regulatory framework (compliance source) since the immutable information stored in BC are available for auditors (Anjum et al., 2017a; Chang et al., 2020).

Consumer satisfaction: The record-keeping properties of BC allows the customer to access certain information related to the products they are consuming or selling to other parties. BC technology presents a suitable mechanism for tracking and tracing products in SCM (Staples et al., 2017). The consumer may access this information by scanning a barcode or ID number in the product, and the information will show up from the origin of the product up to destination (costumer shelf) (Staples et al., 2017).

Trust: The cryptographic algorithms, no single point of failure, and operation without intermediaries (relying on BC network and consensus algorithm instead of third-parties), provides a trust mechanism for the stakeholders, which operate in the network of the SpC (Jeppsson and Olsson, 2017). The context of sharing information in a distributed decentralized ledger is a valuable asset of BC technology since this information is immutable (Jeppsson and Olsson, 2017).

Real-time response: The monitoring (surveillance) of SCM operations is one of the most highlighted issues from the stakeholders in SpC. The BC technology combined with SC allows automation of processes by producing real-time events on transportation, warehousing, and management of goods (Xu et al., 2017; Staples et al., 2017).

Digitization: The global SCM is complex involving multiple parties in each cycle. It is associated with enormous paper works and documents that have high cost and incorporate inefficiencies (data inconsistency, redundancies, document loss, lack of transparency) of SpC (Heutger and Kückelhaus, 2018; Chang et al., 2020). BC enables defining and exchanging verifiable and immutable digital information. In SCM, information exchanged plays a significant role in the sustainability of the SCM (Chang et al., 2020).

TABLE 4.1: The results of the survey presented by Deloitte (Pawczuk et al., 2018).

Percentage (%)	Shared opinion
55	In competitive disadvantages, in case they fail to adopt the technology
45	Little, to no knowledge of BC technology and believed it would disrupt their industry
33	Over-hyped
25	Top five priorities
28	Already invested up to $5 million
25	To invest up to $5 million in 2017
10	To invest up to $10 million

Treacability: The BC ability to record verifiable, immutable, and transparent transaction records present a potential mechanism for improving traceability of goods from origin to destination (Chang et al., 2020).

(Pawczuk et al., 2018) presents the *Deloitte Survey Results on Blockchain Across Industries*, surveying 308 US-based senior executives in BC technologies, which represent companies with a revenue over $500 million. These surveys intend to show the blockchain's impact on business and government, how the blockchain will work cross-industries (Massey et al., 2019; Deloitte, 2019), and the insight of enterprise to use the blockchain in the near future. Some of the survey results are shown in Table 4.1.

4.2.2 Blockchain and Supply Chain Management

(Hackius and Petersen, 2017), realized an online survey with logistic professionals for better understanding the role of BC in logistics and the possible use cases it might solve, enlisting the possible barriers and facilitators, and also the general prospect. In terms of Logistics and SpC, four use cases are present "Ease Paper Work Processing", "Identify Counterfeit Products", "Facilitate the Origin Tracking", and "Integration of BC and IoT" (Cole et al., 2019; Tijan et al., 2019). The conducted survey results indicate BC's positive evaluation to offer potential benefit to solve or improve the current issues in the presented use cases in logistics and SpC (Abeyratne and Monfared, 2016). In (Tribis et al., 2018), a literature review confirms such trends in BC and logistics, especially in "Traceability in Supply Chain", "Supply Chain Finance", and "Information Security of SCM System". Also, in (Juma, 2020), the BC features are highlighted for improving the SCM in cross-border by overcoming issues on inefficient procedures, regulations, and infrastructure service, reduces the operational cost in SCM. (Perboli et al., 2018) analysis the real-world use cases for applying BC in SpC, and it considers BC a promising enhancement, suitable to provide benefits for the involved stakeholders. In (Wang et al., 2019a), examines the proprieties of BC technology from the perspective of applicability in the future SpC. It purposes to find the possible influences of BC in SpC practices and policies. Four-axis a BC impacts SpC are identified: "data security improvement", "visibility and traceability", "supply chain digitalization", and usability of "smart contracts".

4.2.3 Blockchain for Solving Trust Issues in Supply Chain Management

This section contains a set of articles considered and studied related to BC and trust in SCM, Logistics, and Transportation. We select these articles as related works in our study since we treat SCM, Logistics, and Transportation in this thesis study.

Existing software systems, such as ERP[1], nowadays used for SCM, are centralized services. The concern is the "trust", considering the information is stored in a centralized database and "owned" by the host of the service, e.g., a third-party IT firm or stakeholder. This way of storing information does not guarantee immutability and data integrity as it allows the database administrators to alter, delete, or insert other related data records. This model and its technological components enable the sharing of information among stakeholders (Verwijmeren, 2004; Somers and Nelson, 2001). Updating and maintaining the centralized database is a crucial problem since parties are dependent on each other in the SpC, and the database requires a high level of work organization (Sreenivasaiah and Kim, 2010; Coy, 2008). For example, for upgrading systems, some of the processes will remain on hold for the other parties in SC, which may cause delays and inconsistencies in SpC processes, e.g., in the transport of goods.

The current challenges in the SCM come from the management of the workflows of SpC, related to the organization of processes by current systems (Márquez, 2010). Given that these systems are mainly centralized, several issues become evident:

- The lack of trust among stakeholders who cooperate in the SpC (Staples et al., 2017).
- Sharing and disclosing sensitive information among all stakeholders, for example, the substance (goods) for transportation, contractual business terms, capacity, departure point, current warehousing, the timestamp of the movement of goods, and final destination (Márquez, 2010; Imeri et al., 2017).
- Interoperability issues: The new stakeholders should adapt their systems to exchange information with other parties in the SpC (Márquez, 2010; Staples et al., 2017).
- Competitive disadvantages: The possibility to share the information with other business competitors (*EU Note* n.d.; Ellram, 1991).

(Xia and Yongjun, 2017), present a model for the evaluation of trust among enterprises. This model will evaluate the trust between enterprises in the SpC management (SCM), in a BC environment. The trust here (SCM) is defined as the probability that one enterprise predicts to deliver the products to the associated enterprise, as agreed. More concretely this model will evaluate two types of trust: joint enterprise credibility (which is generated from historical cooperation and the interaction of enterprises) and associated credibility (which represents the degree of trust between two enterprises which do not have any direct cooperation), under BC technology. The key characteristics for trust evaluation between enterprises are considered, the transaction satisfactory, product ability, risk probability of information concealment and penalty factor.

[1]Enterprise Resource Planning. The current software systems used in SCM, such as BI, APS, CM, SCE, FDM, etc., (Jennings and Wooldridge, 1998; Mofarrahi et al., 2014).

This research presents by (Biggs et al., 2017), explores the way to benefits from using BC technology in supply chain management (SCM), such as "Transparency", "Scalability", "Trust", "Security", and "Access to new Markets". This affects the end users by increasing the trust of the suppliers while the information related to products and the journey of products are well documented and accessible. Further, this research explains the difficulties and obstacles to using BC in SCM. Among several constraints, the report enlists "Government Regulatory Status" as uncertain and unsettled for BC cryptocurrency market. "Large Energy Consumption", as for validation of transaction requiring high-energy power; "Cross-Industry Integration" which requires transforming the current systems in the full integration of these systems on BC, and this because BC is not a stand-alone system; "Black Market", which sees the utilization of cryptocurrencies, e.g., Bitcoin for money laundering and other illegal actions. The research in (Malik et al., 2019) presents a trust management framework based on consortium BC. It consists of tracking SpC participants' interactions and using a specific mechanism to assign dynamically trust and reputation scores based on their interactions. (Notheisen et al., 2017), present a proof-of-concept prototype for a real case trading of cars in the "market of lemons". The current systems are based on centralized databases managed mainly by a government organization, and they require a specific volume of work and organization for maintaining them. Since many stakeholders are involved, buyers, sellers, government organization, insurances companies, it is challenging to maintain it correctly and to avoid bureaucratic processes. The main intention of this research is to propose a public BC-based solution to replace the current system in car registering and maintaining the history of usage of cars and dealing with changes in ownership. As a public ledger will provide a sufficient set of information for traders, i.e., buyers and sellers of cars, government organization, and other third parties, e.g., insurance companies or banks, which would help to avoid the asymmetry of information. For the implementation of the proof-of-concept prototypes based on BC for automation of transaction in real-world assets, the design science research is applied. The approach produces an IT artifact, with the intention to provide a novel way of registering and maintaining car history from private sellers and buyers, in a distributed shared ledger, with automation properties. The features of BC technology and the research work organization based on design science research provide a trusted platform with a safeguard mechanism in transactions correction in case of possible errors. The benefits from using such a solution will be in the efficiency of the public registration system for cars, followed by the mitigation of transaction risks (by dividing the transaction process into several steps) in a BC based system, in case of conceding any error, and finally in decreasing the risk on trading in the "market of lemons".

Another approach, which uses also the design science research for developing proof of concept prototype, is showed in (Beck et al., 2016). This article explains in general how the trust-free based on BC can replace the trust-based solution, and this is achieved by designing, developing and prototyping a solution. The case study presented is a trust-free coffee shop payment solution that uses a digitalized punched cards as a part of the cryptographic economic system. The cryptographic economic system presents an autonomous system

of transaction, which is not controlled by third parties, e.g., humans, and it follows predefined rules as a protocol implemented in computers. For conceptualizing this solution, design science research guidelines are used. The artifacts from this research produce an IT solution based on BC technology, which is a new approach and contributes to solving specific problems for society. The smart contracts from the Ethereum BC framework are used for implementing this solution. This helps in the creation of trust-free self-service and automation of rules in the process of purchasing coffee. The potential of BC related to the trust-free system based on BC technology is highlighted, while its applicability faces some obstacles for real-world implementation.

(Yuan and Wang, 2016) evaluates BC technology as a potential for changing the intelligent transportation systems (ITS). The BC potential is discussed to transform centralized systems into a secure, decentralized, and autonomous transportation system. This study outlines a conceptual transformation of BC-based on ITS. It highlights the issues of security risks originating from the centralized authorities for data management and trust issues among ITS entities, which results in the prevention of flow of "entities", e.g., money without intermediaries. Therefore, (Yuan and Wang, 2016) propose a conceptual model composed of seven layers to provide a secured, trusted, and decentralized architecture for enabling a free data flow, money, and assets on the ITS system. These layers presented in the bottom-up form are: "Physical Layer", "Data Layer", "Network Layer", "Consensus Layer", "Incentive Layer", "Contract Layer", "Application-Layer". This layer forms an architecture overview based on BC and IoT. The basic architecture of a BC-based ITS is presented topologically as a P2P network, and it is composed of many autonomous, decentralized, decision-making devices[2]. It follows the BC mechanism's general properties to act on security, trust and distributed intelligent transportation system.

The potential of BC as validation tools by practicing the recording transactions, validation processes and its possible usability for access control are presented by (Anjum et al., 2017b). Following the distributed mechanism, the trust, validation and the compliance issues are studied in this research. The security and trust are enhanced by the decentralized methods, independent verifications which are BCs properties. This analysis considers that the standardization is a crucial issue for achieving the best usability of BC. This research highlights the needs for standardization of storage algorithms, signature algorithms, web-based access protocols.

The findings from research in (Wan et al., 2020) support that using BC technology, the collaboration of stakeholders may be enhanced by sharing verifiable and immutable information. BC technology may transform the current way of storing information (centralized approach) by enabling a decentralized and without single authority on information management (Wan et al., 2020). Several fields of SpC are identified on which sharing information impacts the SpC efficiency. In (Nakasumi, 2017), a BC-based information sharing solution is proposed to solve the issues of asymmetry of information. Businesses hesitate to share information due to today's highly competitive SpC. (Engelenburg et al., 2019) propose a BC-based architecture for storing event rules for information sharing. The business can control and manage these

[2]The most related case study related to the presented architecture is lazoo application.

rules, thus avoiding the risk of sharing sensitive information. (Huang et al., 2020) propose a BC-based data sharing schema, with anonymity and traceability features, which enables sharing of information with multiple parties to improve cooperation. A framework for document sharing and version control, enabling multiple user collaboration, and tracking changes are proposed in (Nizamuddin et al., 2019). (Imeri and Khadraoui, 2018) present a new conceptual approach for the security and traceability of shared information in the context of the transportation of dangerous goods. Business contracts are taken into account, and the information related to contractual terms are secured; the goods are traced while moving through the SpC. The approach compares the current existing technological systems to support the logistics of transportation of DG, and it proposes a conceptual solution based on BC.

4.2.4 Business Process Management (BPM) and Blockchain Integration

In this section, we present the context of business process management (BPM) (A.5.2.1) and BC technology integration. It presents a way of integrating BC into the choreography of the process to improve trust and avoid third parties' involvement.

BC technology's ability to execute processes in a trustworthy manner shows the potential for rethinking a new way of designing inter-organizational business processes (Mendling et al., 2018). SC enables expressing business process (BP) rules since any BP is subject to specific business rules[3]. SC helps in organizing and execute business rules in an inter-organizational environment and global monitoring of the process. Using BC technology to implement an inter-organizational BP, the participants can review immutable data records over the execution of the BP. In case any error (disagreement) happens during the process, the immutable properties help to find (solve) the error or dispute (Mendling et al., 2018).

The research in (Müller et al., 2020), presents trust patterns for collaborative BP. It may be used to highlight the benefits of BC technology to enhance trust in collaborative BP. (Weber et al., 2016) shows an approach for monitoring and coordinating BP by integrating BC into processes choreography. Trust is considered as a pre-condition to develop inter-organizational BP. The proposed approach intends to address trust issues in the collaborative business process execution by mapping BP in a peer-to-peer infrastructure to solve the trust issues since transactions are stored in the BC. It is composed of three main components, *Translator*, which is a design-time component that translates BP into SC. *Choreography* monitor, which is used for monitoring of BP by employing SC. A *Mediator* component that plays an active role in exchanging messages between stakeholders involved in the BP, based on the business logic model supported and implemented by SC. BC *Interfaces* or triggers that play the connector role with an external component such as API to help SCs connect with the external world (off-chain). The BC is used as a choreography monitor, which stores the state of process execution between all stakeholders involved, and SC verifies if the interactions between stakeholders are fulfilling the defined choreography model (Weber et al., 2016).

[3] A business rules may express the following condition: "if the products are not delivered to the buyers for a period, sellers should pay the penalty".

(López-Pintado et al., 2019) presents an approach for deploying BP management system (BPMS) on top of the Ethereum BC. This approach is based on model-driven engineering (MDE), and it proposes deploying BPMN models on top of Ethereum. This research shows the design principles of the *"Caterpillar"* [4] tool, which permits the translation of BPMN models into SC. It enables the creation of BPMN models and allows the users to track the process state. The *Caterpillar* main advancement is that each process's execution state is maintained on Ethereum. The entire workflow is covered by SC generated by BPMN using Solidity compiler (López-Pintado et al., 2019; Mohan and Kampik, n.d.). This generation of the BPMN models covers a large spectrum of the BPMN notions, such as sub-processes, multi-instance activities, and event-handlings. The Caterpillar design principles require that the collaborators (stakeholders) agreed-upon a collaborative BP, and BC provides the necessary mechanism to ensure the parties involved in this process to comply with BP (López-Pintado et al., 2019). The compliance towards the collaborative BP is ensured either by "monitoring" BC transaction logs or by the "compliance by design" approach, which checks the process state before executing a transaction on the BC (López-Pintado et al., 2019).

In a similar direction, the research in (Lu et al., 2020) shows an MDE-based method for generating SC code[5] from BPMN models. This research proposes a methodology approach for solving the issues on asset (fungible and non-fungible) management by using BC, with the motivation of avoiding centralized processing. This work's core advancement is that it allows expression of process model related to asset management and generating SC code according to the Ethereum compliance standard (ERC20 and ERC721) for token registration (Lu et al., 2020).

4.2.5 The Traceability of Goods in Supply Chain Management and Logistics

The issue of traceability is one of the most highlighted use cases from SCM and Logistics stakeholders. The study from (Insights, 2017) enlists the **"Tracking product moving through the SpC"** use case as the most preferred among all the other use cases by evaluating it with 80% as the most voted from stakeholders in SCM.

The traditional logistic system does not match the new market demands. The low food safety and the losses are considered enormous by logistics processes because of a missing traceability system (Feng Tian, 2016). The traceability system is based on the usability of RFID technology for data management from logistics sectors, and it uses BC technology and its properties to ensure that the shared information among stakeholders is immutable. This research intends to improve the quality of food by providing a solution for the traceability system on the agri-food SpC. This would significantly improve the trustability in the SpC since the information shared in the BC system is immutable. The approach presented is a conceptual solution and, enlists the benefits of using RFID and BC in SpC agri-food. Using these technologies on a traceability system, the benefits are significant on information security and fighting against fraudulent products. While the disadvantages are in the high price of

[4] Caterpillar is the prototype that demonstrates the BP execution engine deployed on the BC (López-Pintado et al., 2019). It is publicly accessible in the following link: `https://github.com/orlenyslp/Caterpillar`

[5] The SC code generation is performed according to the previous research work showed in (Weber et al., 2016).

implementation of such a system influenced by RFID price and the immaturity of BC (Feng Tian, 2016). Following in (Tian, 2017), the author extended research in food SpC by proposing systems that collect information by using IoT (RFID, WSN, GPS) devices and stores this information into BigchainDB[6]. The traceability of products is performed in different stages, from the "Production", "Processing", "Warehousing", "Distribution" and "Retail". By scanning the RFID of the product, a consumer can retrieve important information about the products.

The research presented by (Badzar, 2016) studies the possible improvements of transparency for suppliers and consumers, and the development of contractual coordination concerning sustainability clauses. The potential of BC for logistics and transportation is highlighted by using the measurement of innovation for BC in logistics, based on a specific method and analysis of the empirical findings. The evaluation of BC in logistics as an innovation is performed using a the well-known method, composed of five main points: Relative advantages, Compatibility, Complexity, Trialability, Observability (Rogers, 2003). The conclusions reached are that BC has the full potential to generate transparency and ensure the fulfillment of contractual terms between suppliers' operations in the SpC.

(Petersen and Jansson, 2017) present a framework for businesses to evaluate the possible applicability of BC in SpC management to improve traceability. This framework aims to discover the necessary inputs and evaluation tools needed for the applicability of BC technology into SpC management. This framework is based on the theory of SpC traceability and BC technology and is composed of three main steps: (1) identification of drivers for traceability (or areas for improvement), (2) creating the main principles of BC and (3) evaluating the sustainability applicability and technical limitations. The process methods could follow these steps in a disorderly manner, and as a result, it outputs the effectiveness from the implementation of BC technology in SpC traceability (or any area targeted for evaluation).

The research by (Jeppsson and Olsson, 2017) describes the whole process of traceability. BC technology is integrated with the smartphone to follow the production from the manufacturer, then through warehouses, to the store (retailer). The research highlights that for the possible implementation of this technology and its continuous usability, cooperation between suppliers is required, then for having a footprint, the usability of the smartphone is mandatory, and finally, an integrated system should exist between suppliers involved in the process.

The research ((Salah et al., 2019) & (Lin et al., 2018)) proposes a BC-based framework for traceability and visibility in the SpC of soybean. In the food SpC, there are used communication tools such as bar-codes and RFID for precize data acquisition to improve traceability. The authors highlight the traceability issues on the current food SpC by mentioning the data fragmentation and centralized control, which allows modification of data. This research proposes a solution that is based on a design method that uses entity relation and sequence diagram to show participants' interaction in the SpC. The proposed solution is based on the public Ethereum BC framework, and it uses the execution of autonomous SC. As a solution, it covers all participants. It begins with the seeds company and ends with the customer. It

[6]For overcoming the issues BC scalability it propose using BigchainDB[7]. BigchainDB presents the concept of the distributed database with BC properties. The significant concepts of BigchainDB are high throughput, low latency, high capacity, decentralized control, immutability, creation, and movement of digital asset

proposes to store all details related to seeds, then its purchases by a farmer, the growths of crop, harvesting and storing, processing and final product, and finally distribution and selling to end customers. However, this research does not consider the privacy aspects and regulatory framework. The research from (Ahmed et al., 2020) proposes data management traceability architecture for improving quality traceability data. The approach enables secure sharing of information among different stakeholders, and it proposes a generic smart contract for B2B traceability data management.

A conceptual framework for tracking online shipments in the distribution phase by all stakeholders is proposed in (Wu et al., 2017). The proposed framework is composed of several private distributed ledgers and a single public distributed ledger. The private ledgers are used to store and manage information related to business trading partners, i.e., all custody events[8] are stored in the private ledgers. The hashes of the events from private ledgers and monitoring events are stored in the public ledger. The monitoring events are related to providing information for trucks' current positioning, shipment information related to any associated shipment from the private ledger. This framework's architecture is composed of several sub-networks that are created when an order is placed, e.g., transferring goods from supplier to carrier, and it is terminated when the goods are delivered to the customer. A global network that allows all stakeholders' participation, including third parties (monitors), and it is used for timestamping all records of shipments. This architecture is composed of four main elements: a) index server, which stores and maintain the addresses of all nodes in the network; b) peers, which presents many roles of participants in the network, i.e., customer, supplier, and carrier); c) administrative node, which communicates with the ERP of enterprises, and d) external monitors, which validates the geolocation of shipments, and for the reason of visibility on SC, post this information on the public ledger. The information is posted by nodes regarding events and depending on the event's nature; they are posted either in private ledgers or public ledger. For complementing this framework and maintaining the interoperability with current SpC systems, a data model and data representation are presented using EDI-214 standard and JSON[9] (Wu et al., 2017).

(Kim and Laskowski, 2018) treats the problem of determining the provenance for the goods in SpC. In an inter-organizational and complex SpC, the physical provenance of goods, e.g., pharmaceutic or authenticity luxury goods, is not always possible because of technological limitation and complexity of the SpC. For solving such issues, (Kim and Laskowski, 2018) highlights the potential of BC technology. In combination with IoT and using the ontologies that represent knowledge about provenance and traceability, provenance issues are to be answered. This research aims to develop an ontology-based BC approach for responding to the problems of provenance in SpC. Using the ontologies is for better data standards and formal specification for automated interfaces, which helps develop better SpC . In this context, the TOVE Ontology[10] for fundamental concepts of traceability is used

[8]For example, transferring the shipment from suppliers to the carrier is considering a custody event (Wu et al., 2017)

[9]JavaScript Object Notation (JSON), https://www.json.org/json-en.html

[10]http://www.eil.utoronto.ca/theory/enterprise-modelling/tove/

to provide the provenance of goods in SC. Proof of concepts is developed, which uses a "traceable resource unit," an object to be traced from one part to another part of SC.

The research from (Lu and Xu, 2017) presents *originChain*, a traceability system based on BC and smart contracts. This system intends to provide transparent, tamper-proof traceability data, data availability, and also it considers regulatory-compliance aspects by automatically checking them. It is mainly applied to companies that import products to China. It considers the traceability perspectives of the suppliers and retailers. The supplier's traceability perspective is to prove the product origin and quality and regulatory compliance, while the retailer's perspective is product origin and quality. The *originChain* works as traceability providers, and the stakeholder that needs such a system applies for traceability services. The architecture of *originChain* indicates that the nodes are geographically distributed over three different premises and supported by private BC. Data storage aspects in the BC manage several data sources off-chain while storing the hash address of such data on-chain. However, this solution is limited to service providers, and its traceability services are provided following a "contract" signed between parties for the offered traceability services.

4.2.6 Blockchain Based Startups for Logistics and Supply Chain Management

(Kshetri, 2018) examines the influences of BC technology in SCM. SCM's objectives, such as SCM costs, speed, service and product quality, dependability, possibility of risk reduction, sustainability, and flexibility, are investigated in this research. It also considers that the involvement of IoT tends to affect SCM. The real-time tracking for the shipments and containers is enhanced by using IoT devices such as RFID tags, barcodes, GPS tags, sensors, and chips. This research's theoretical framework is based on the selection of existing use cases from the business perspective. BC technology eliminates the need for a middleman, which reduces costs significantly since all stakeholders can check their shipments individually. Further, the digitalization of communication, i.e., the document digitalization, reduces SCM costs, e.g., "Maersk" BC-based solution for container traceability. This digitalization speeds up the process of validation of documentation in the case of transportation. That will reduce the paperwork and increase work efficiency. The quality of goods during transportation is measured by analyzing the real-time data collected during the transportation process. Based on the goods' specificity, the stakeholders estimate if any possible damage occurred during transportation or warehousing, e.g., "Modum" case for pharmaceutic products transportation.

The visibility of acting by partners in SpC clarifies the dependability of SpC partners that are operating together. That drives SpC partners to be more responsible for their actions. Since the validation of individuals' identities and assets operation in SpC is provided by BC technology, the cybersecurity risks significantly reduce.

In Table 4.2, we present several initiatives related to BC, SCM and Logistics.

There are other SCM solutions with similar properties. The first example is "OpenPort" (Lim and Noman, 2018) which supports companies in a variety of ways with a huge list of features. Apart from the fact that there is no build-in option for decentralised authorisation, the main issue is the storage of data in the BC. In case of sensitive data, the data is encrypted.

Project name	Description	Core involvement
TKI Dinalog	A consortium of companies are gathered to develop a BC based project which intend to deliver three concrete use cases: chain financing, supply financing, and circular economics. BC is consider a stepping stone towards a logistics sector to improve the collaboration in the entire chain (TKI-Dinalog, 2017)	finance SpC
SmartLog Project	The SmartLog project intend to integrate BC and IoT, to improve the information sharing for logistics and SpC stakeholders. The main use case for this projects is sharing of containers location. It gather many stakeholders and intends to share information with different ERP systems. The project is at proof of concepts stages (*Project SmartLog* 2019)	information sharing
T-Mining & Port of Antwerp	This projects intends to improve the container handling in the Port of Antwerp. The purpose of this project is to provide a secure and efficient container release, and to provide secure and efficient document release (T-Mining, 2017)	secure container handling; efficient documentation
Maersk and IBM joining to develop TreadLens	Maersk and IBM provide cross-border SpC solutions for improving digitalization in SC. As leading container shipping in SpC, Maersk found that in single container transport from (East Africa to Europe) there are required, stamps of 30 persons/organization and 200 interactions on this occasion. This project intends to help manage and track the paper trail of shipping containers across the world by digitizing the SpC process from end-to-end to enhance transparency and the highly secure sharing of information among trading partners (Taylor, 2017; *IBM and Maersk* 2017; Hackius and Petersen, 2017)	digitization of SpC
Walmart and IBM	The main intention of this project is to provide a distributed ledger technology to track and trace several products, e.g. Mexican Mangos or Pork meat imported from in China etc. (Walmart and IBM, 2017)	food traceability

Project name	**Description**	Limitation (general issues with project
Carrefour	Launches the "first food treacability BC. The core idea behind this system is to ensure complete traceability of product by involving all parties i.e., producers, processors and distributors. Shops can use QR Code (by smartphone) of products and to retrieve information of product at each stage of production the also the journey of product (Carrefour, 2019)	food traceability
Provenance	Provenance provide BC based solution that intends to proof the authenticity of products, e.g., luxury diamonds. It takes up to forty data points and stores them in BC. In terms of SpC it intends to make opaque SpC transparent. This project intends to provide trust related to the goods moving in the SpC (Provenance, 2017)	product authenticity; fight against counterfeits products
Tech Mahindra & Quantoz	Is developing a BC-based solution (Quasar) for financial SpC based on the permissioned ledger. The purposed solution is based on electronic cash system with built-in rules to fulfil regulatory and compliance guidelines (*Mahindra and IBM* 2017)	financial SpC
SyncFab	Uses BC technology for providing transparent order management i.e., order tacking. It provide BC based solution that slash the procurement time and cost and enable secure track-and-trace orders (SyncFab, 2020)	procurement SpC
Seam	The SEAM and IBM, forming an ecosystem of suppliers for a cotton industry and trade, based in Hyperledger Fabric (Seam-IBM, n.d.)	trade SpC

TABLE 4.2: The summary of industrial projects for blockchain in SCM.

This is a valid approach to deal with the problem, but using encrypted data excludes the use of smart contracts on that data. The encryption and decryption keys would have to be part of the SC, which would be transparent to all with granted access to the BC.

A second problem are emergency response teams, who need access to the data, but should not be impacted by the use of crypto. "Skuchain"(Skuchain, 2018) goes a step further and provides encryption on field-level. It builds on top of Hyperledger Fabric and stores confidential data off-chain but also uses a zero knowledge proofs mechanism to support collaboration. "OriginTrail" (Rakic et al., 2017) presents a data exchange protocol and upper layers that are built on top of a BC that can be chosen by the customer. The BC is mainly used as a storage for hash values that proof the existence of data outside the BC. It is used as a decentralized time-stamping service.

4.2.7 Challenges in Integrating Blockchain in Supply Chain and Logistics

Besides the BC's encouragement for bringing values to the SCM, integrating the current SCM is considered challenging. (Korpela et al., 2017) studies the challenges of integrating data in the business-to-business case by bringing a consortium of companies cooperation together for a standard integration and collaboration, under the concept of the digital SpC (DSpC). It proposes a digital transformation of SC towards BC by investigating the possible integration of DSpC into the BC, by questioning on *how this technology would accelerate this integration, and further how BC will support this integration?* From the business perspective, end-to-end integration refers to electronic data exchange between all SC stakeholders. A challenge is to integrate data in the business-to-business (B2B) case by investigating DSpC integration requirements and functionalities. Currently, stakeholders use third parties to share information and execute specific work (Korpela et al., 2017). The primary requirements for SpC integration are specific on "business model development", "information model platform," "developing new business process standards for enabling SpC connectivity", and "new way of transferring data between stakeholders in the SpC" (Korpela et al., 2017). Considered by this study, conducted in the year 2016, end-to-end integration is not fully supported by BC technology since there is not any standardized data model offered by BC. However, BC functionality supports many of the identified functionalities and among the most preferred was system security, privacy, and smart contracts (Korpela et al., 2017).

Challenging is also considered the complexity of the SCM global network, and a variety of laws, regulations, and institutions determine the rules by which members of SpC should comply. Secondly, a BC-based solution for a particular SC needs a comprehensible agreement among all participants. The boundaries between physical and virtual context are considered as the third challenge since BC stores virtually the physical objects, and it follows in that way while in the real world, the physical object may be changed, stoled (the material inside containers) or damaged. Forth, from the technology perspective, the current BC design, e.g., private or consortium BC, partially lose the "decentralized structure", and finally, since the computational power requirements are high, it disables the opportunity of all countries to participate in SpC (Kshetri, 2018). (Mendling et al., 2018) considers challenges are on *"difficulty to integrate BC with enterprise strategies"*, then *"the impact of BC in governance"*, the *"security and privacy risks"*. (Wan et al., 2020) considers challenging (barriers) due to "regulatory uncertainty", "joining many parties together", "lack of technology maturity", "lack of understanding of BC technology", "lack of acceptance in the industry", "data security cancers", and "dependency on BC operators" (Hackius and Petersen, 2017). (Wan et al., 2020) claims that a stakeholders' hesitance to share information with other SpC members is considered a barrier to the BC-based solution.

(Tribis et al., 2018) claims that the significant gaps in applying BC in logistics and SpC are external factors such as legal factors. The interoperability, technology immature, standardization, legal, and regulatory issues are listed as challenges for BC implementation in SCM (Chang et al., 2020; Paliwal et al., 2020). There is uncertainty when deploying nodes (geographically) in terms of legal issues, to which legal authority might decide on possible issues, and which laws to follow.

Interoperability, present, a way of sharing information, transact and operate across various BC systems (Hardjono et al., 2018; Pillai et al., 2020). The current BCs do not communicate or share information, which is considered a major drawback to implementing BC in SCM. As also confirmed in Table 4.1, the lack of knowledge of technological knowledge is one of the main challenges in adapting the BC technology. The current BC platforms present a certain level of complexity, thus preventing an organization from migrating into BC (Chang et al., 2020). Also, BC's technical limitation, such as scalability, imposes additional challenges for widespread BC SCM 3.4.4.

The research in (Staples et al., 2017) presets the primary criteria for selecting a use case for development by applying the BC technology. Depending on the BP, the functional and non-functional requirements should consider measuring and improving this technology adaptivity in BP.

4.3 Integration of Blockchain and IoT

Several approaches use BC as immutable logs for the IoT data, and some others propose specific use case where both BC and IoT technologies are used. There exist also several surveys on BC integration with IoT (Panarello et al., 2018; Conoscenti et al., 2016; Khan and Salah, 2018; Dabbagh et al., 2020; Fernández-Caramés and Fraga-Lamas, 2018). The research presented in (Panarello et al., 2018; Golatowski et al., 2019; Dorri et al., 2016) shows challenges and opportunities on the integration of BC and IoT. The challenges are highlighted for the use cases that use public BC e.g., Bitcoin or Ethereum as an immutable log of IoT data, in the sense that these networks are not scalable. Furthermore, in such use cases, it might not be reliable if the nodes i.e., "miners" do not join the network. On the other hand, the private BC solves the issues of scalability and privacy, but the decentralization aspects decrease.

While the benefits are encountered in designing and developing solutions that incorporate data privacy, data integrity and designing systems that will be able to manage the identity of devices in a tamper-proof manner (Panarello et al., 2018; Reyna et al., 2018). The research from (Huh et al., 2017) proposes a way to manage IoT devices by using Ethereum BC. The defined policy (turn the device on/off in certain conditions, e.g., when the temperature is reaching certain value) and temperature updates are posted into the Ethereum network with the help of a smartphone and Raspberry Pi. Other devices are retrieving certain values from this policy, in a periodic way. The solution uses also SC for updating the temperature and adding policies about devices. In (Košťál et al., 2019), the IoT devices are managed and monitored using BC and SC for managing the configuration files of the IoT devices. In this approach, the certified network administrators are allowed to add new or update configurations of IoT devices and then put them on the BC, which further raises an event to notify the targeted IoT devices. Further, the targeted IoT devices decipher the configuration using their private keys and add them to their configuration files. Authors in (Rifi et al., 2017) advocate that BC technology has attractive properties for decentralizing the IoT, thus proposing an architecture that is based on the combination centralized-decentralized approach. The basic idea is to use intermediate servers between IoT devices and BC framework. The SCs are used to maintain

the authentication, rules, and communication between involved parties. In (Liao et al., 2017), four architectural styles for BC and IoT are presented namely "Fully Centralized", "Pseudo Distributed Thing", "Distributed Things", "Fully Distributed". The first two architecture styles use BC for recoding payment transactions and hosting BC node on cloud respectively, thus not benefiting entirely from the BC. On contrary to them, they remain architecture styles that benefit from BC technological abilities, consequently being robust and with data integration properties (Liao et al., 2017).

IOTA[11] based on the Tangle ledger uses a Directed Acyclic Graph (DAG) to add a transaction on the ledger. For adding a new transaction in the Tangle, nodes should select two previous transactions to be validated, then a small computing power is needed to add a new transaction. This way of adding a new transaction, and without the mining process improves the scalability while as many nodes joining the network, the transaction validation is faster (Iota, 2017; Popov, 2018; IOTABlog, 2018). In (Tolmach et al., 2021) "tendermint" is highlighted as a suitable consensus algorithm for integrating IoT and BC (Tendermint, 2021; Tendermint-SDK, 2017). In (Jiang et al., 2019) a cross-chain solution integrates different BC frameworks for managing the IoT data securely and efficiently. The idea behind this research is a decentralized access model based on a consortium BC that acts as a control system (Jiang et al., 2019). It uses other BC frameworks (several sub-networks), such as IOTA for IoT data management. The role of IOTA (Tangle) in this approach is to provide an immutable log for IoT devices, while the consortium BC role is to record and control any access to these data coming from IoT devices through BC frameworks, i.e., sub-tangle (IOTA) (Jiang et al., 2019).

In Enigma (Zyskind et al., 2018), the privately shared data are stored on the "modified distributed hashtables". In this approach, a BC is also used for managing the access control, identities, and servers in an immutable way. This approach proposes that the public part of the data be stored on the BC while the private part be stored off-chain (on Enigma platform). This solution provides a certain level of scalability and is a good candidate to be used with IoT (IoTBlog, 2020). There are considered limitations, such as decentralized *off-chain distributed hash-tables* (DHT) (Zyskind et al., 2018). The research from (Christidis and Devetsikiotis, 2016) showed that BC and smart contracts in combination with IoT have a significant impact on the automation of processes. In (Lin et al., 2017), a BC is used to secure a Long-Range wide-area network (LoRaWAN) IoT. The authors considered that since LoRaWAN for IoT is usually operated by private organisations, their approach proposes to store the data in the network servers before transferring them to back-end application servers. However, the approach is limited since it does not consider the throughput issues and latency. (Qian et al., 2018), highlights more explicitly the security risks in a use case of "autonomous vehicles" i.e., the internet of vehicles, and propose to solve these issues implementing three layers of IoT are presented: "perception layer", "network layer" and "application layer". For overcoming the security risk for IoT systems, authors propose a BC-based solution with the focus on the traceability of the IoT devices. This traceability is applied to the interactions of the IoT devices with the network (mobility) and the interactions of the IoT devices with the cloud (data) (Qian et al., 2018).

[11] IOTA is known as "distributed ledger" for IoT (Raschendorfer et al., 2019; Harbor, 2018)

For improving the issues of IoT authentication (because of the large number of IoT devices, and efficient authentication system in a centralized approach is almost impossible) and ensuring data integrity, the research in (Hammi et al., 2018) proposes using Ethereum BC technology properties. The approach consists of creating IoT environment virtual zones where each device must communicate only with IoT devices that belong to the same zone. This zone is considered a trusted zone (trusted bubble), and the other IoT devices that are not part of the trusted zone cannot communicate with any device in a trusted zone. The approach uses public BC and SC to authenticate IoT devices based on trust zones to achieve this. For example, if IoT_A sends a message to IoT_B, the message of IoT_A is send as a transaction in the BC; if BC authenticates IoT_A, then IoT_B can read the message (Hammi et al., 2018). Also (Hang and Kim, 2019) propose an integrated IoT-BC platform for ensuring sensing data integrity. The approach delivers an application that offers a comprehensive immutable log of data and improves device owner access for the deployed IoT devices. (Liu et al., 2017), proposes a BC-based framework for ensuring data integrity, which consists of providing mode data integrity verification, in IoT-cloud based frameworks. (Teslya and Ryabchikov, 2017) propose an architecture for integration of IoT information sharing platform (SMART-M3) and BC technology. It consists on deploying SC which manages information received from IoT devices, and distributing message accordingly. The research in (Dorri et al., 2017a; Dorri et al., 2017b) proposes an optimized solution for a smart home use case with a specific focus on IoT security and privacy . They propose to deploy a "miner" in each home to manage the communication with the outside world. The miner manages all devices that are deployed inside the home. (Müller et al., 2019) uses IoT and BC for tracking the handover of parcels between many organization. The IoT devices are used to capture data and store them in BC, thus eventuating possible parcel handover agreements.

The trustworthiness of IoT data is of paramount importance since more and more other systems are relying on these data. (Sigwart et al., 2019) propose a framework for IoT data provenance based on BC. The framework shows functional and non-functional requirements for generic IoT data provenance.

4.4 Smart Contract: Related Works Studies

The emergence of the Ethereum (ETH) BC framework, with the possibility of deploying smart contract (SC), enabled a new way of execution of application over the BC network (Buterin, 2017). The combination of BC and SC enabled a new market for a decentralized application that provides a new level of automation of many business processes (Mendling et al., 2018). Simultaneously, with the opportunities offered by BC and SC, there are various concerns to considers for BC-based applications. These issues are on designing a BC-based application that should behave as intended by the end-user. Further, the security and privacy issues, performance issues, and programmable issues, and maintainability of already running BC solutions are amongst the most highlighted concerns when considering designing a BC-based application (Alharby and Moorsel, 2017). The research and industrial communities already faced the before-mentioned problems. To overcome these issues, they have discovered several

design patterns (Gamma et al., 1995) as the best practices from the community, which help researchers and developers design reliable SC (SCBestPractices, 2016).

4.4.1 Blockchain and Smart Contract Design

SC is the subject of many studies from academia, research organizations, and also industry. Designing SC is one of the main challenges highlighted recently by scholars and industry. The literature review shows that there are presented several design patterns for supporting the best practices for designing a SC. Mainly these design patterns are on "security of SC" (Wohrer and Zdun, 2018), "structural patterns" (Liu et al., 2018), "privacy issues" (Alharby and Moorsel, 2017), "performance issues" (Wohrer and Zdun, 2018; Frantz and Nowostawski, 2016; SCSafety, n.d.). The research from (*SCUpgrade* 2019), proposes an upgradable SC by using a proxy pattern. Research from (Liu et al., 2018) summarizes SC design patterns based on the existing SC and further applies some of the design patterns in a real work BC-based application for traceability. There are presented different classifications of patterns for designing SC such as "action and control", "authorization", "lifecycle", "maintenance", "security" are presented in (Wohrer and Zdun, 2018). Within certain classes of the design patterns for SC, the maintenance pattern is among the most used. A typical example of maintenance pattern is the proposal **satellite** and **contract relay** (Wohrer and Zdun, 2018). The satellite pattern is using two SC: *satellite* and *base* (Wohrer and Zdun, 2018). It enables to update a variable from *base* by calculating the value from the *satellite* thanks to the address reference of the *satellite* held into *base*. This allows to dynamically change the value of the variable by just upgrading the *satellite* contract with the newest calculus and updating the *satellite* address in the *base* contract. Further, the **contract relay** pattern is using two contracts: *base* and *relay*. The *relay* contract serves as an entry point in order to provide the latest version/address of the *base* contract, and then forward any call to it. This is a proxy enabling to upgrade the *base* contract without upgrading the user entry point (*relay*). Nevertheless, the drawback that the newer data storage needs to be consistent to avoid data corruption. These two design patterns propose solutions including good practices for maintenance issues that well fit public BCs. It requires sophisticated programming skills in order to implement it correctly, and further maintain it.

4.4.2 Role Based Access Control: Related Work Studies

Access control is a security technique that determines who has access to specific system resources (Liu et al., 2020). Access control is considered as an issue on the BC, particularly with public BC, e.g., Ethereum and Bitcoin. There is already a movement to define access control by using programming primitives. For example, SC programming language Solidity offers limited access control features applied on built-in methods (Liu et al., 2020). Currently, the consortium and private BC, e.g., Hyperledger Fabric, provides a general access-control policy. The disadvantageous side of such policies is that they are static, while the stakeholders' roles are subject to change continuously, disturbing the current development.

The research in (López-Pintado et al., 2018), presents a model for dynamic binding of actors (stakeholders) with a role and policy specification in the collaborative business process. The dynamic changes on the set of the actors impose dynamic changes in the trust relation as well. This research shows a policy specification that is consistently verified based on Petri Net semantics. Their approach consists of outlining the specified policy into a SC enforce it in BC accounts (roles). In (Liu et al., 2020) a Smart Contract Access Control Service (SMACS) is presented. SMACS intends to manage the access control over the SC by determining who has the right to execute a SC or any function of SC. This SMACS is a framework that enables realizing an updatable access control rule (ACR) for SC. Executing operations (SC function calls) on-chain in the public BC, e.g., Ethereum, is costly, and SMACS intends to overcome these issues by shifting the ACR validation and management off-chain. Further, SMACS implements on-chain only a lightweight token-based access control.

The research showed in (Wang et al., 2019b) addresses the safety and security of SC, executed in BC as a Service[12]. The workflow policy (defined in Workbench of BaaS) of the SC is implemented as a high-level finite state machine (FSM), and it is composed of a "set of user (roles)", "the different states of SC," and "set of functions of the SC restricted at each state of SC". The intention behind this research is to verify if the designed SC implements the workflow policy correctly. For performing a formal verification, the Solidity code is encoded at low-level verification language Boogie[13]. This translation enables verification and counter-example generation to verify if the defined workflow policy is satisfied at any state of SC.

4.4.3 Smart Contract Formal Specification and Verification

SC's security issues need high attention since they contain millions of dollars in virtual coins or run the business process daily tasks. Primarily, a high attention is required before deploying SC. The risks stand that once deploying SC into the BC, they remain immutable (impossible to patch), and there is no way of stopping them (Grossman et al., 2017; Kolluri et al., 2018).

4.4.3.1 The Vulnerabilities of Smart Contract

The well-known case of SC vulnerability is the *DAO* attack, which caused the loss of more than sixty million dollars in Ether (Grossman et al., 2017; Castillo, 2016).

Among the security bugs (Kolluri et al., 2018; Luu et al., 2016; Atzei et al., 2017) and other SC issues discovered we can list:

- *Dependence on transaction-ordering*: Presents security issues where an event (of function) of SC is depended on the other previous events in order to behave correctly.
- *Timestamp*: The dependence of the SC for performing an event.

[12] https://azure.microsoft.com/en-us/services/blockchain-service/

[13] The Boogie intermediate verification language: https://boogie-docs.readthedocs.io/en/latest/

- *Throwing an uncontrollable exception*: This is the situation where a SC calls another SC, or some function of the SC by using *someThing.send(someValues)*. In case there is an exception for a certain reasons, and the called SC or function returns *false*, the process value *"someValues"*, will not reach the destination. In this situation, there is required that the SC that calls any other SC or function should check priory if the calls are made properly.
- *Reentrancy*: This is a security flaw when a SC function calls another SC function on the SC. The called SC, can take control of data flow and make changes over data. The *DAO* attack sourced from the reentrancy issues (SCBestPractices, 2016).
- *Linearizability*[14]: This is a security issue raised in SC when an off-chain service is called.
- *SC misbehaviour*: Present an issue when a SC is not behaving as it was intended.

Blockchain and SC are currently in the highest level of interest as research topics from scholars and market researchers. The results in this study signify that the research field of SC and model checking and verification is relatively new based on the research articles published. For achieving significant results on this survey paper, we defined a research method for selecting, classifying, and analyzing the most significant research results. First, we defined main research questions that are the core of this research: *Model-checking techniques for SC?; How to verify the SC is running as it intended?; How to confirm the correctness of SC with natural laws and regulation?; Tools and best practices on verifying SC?*

4.4.3.2 Overview of Model Checking and Verification Techniques

This section introduces an overview of model checking techniques, and further, we highlight the most relevant scientific approaches for model checking and verification of the SC for responding to SC vulnerabilities presented in section 4.4.3.1. Table 4.3 summary the current most relevant tools, frameworks, and approaches for a secure and well-behaved SC.

4.4.3.3 Formal Methods

The formal method allows expressing a complex model for a computer system based on mathematical expressions. For obtaining the correct behavior of the model, formal methods use mathematical proofs to ensure the correctness of the model (Collins, 1998). Further, the model checking techniques allow verifying all the states that are provided by the model. Initially, there is a required specification of the model, mainly by using temporal logic and then systematically performing verification over all the specifications defined (Clarke, 2001) (Collins, 1998). This means that all possible "theorems" defined on the specification, need to be examined for all possible states of the model (Collins, 1998; Baier and Katoen, 2008). The model would be possible to be implemented when the previous stages "specification"

[14] A classic example of linearizability is that all the users involved in the concurrent process should see the same state of data: `https://jepsen.io/consistency/models/linearizable`

and "verification" are successfully completed (Collins, 1998; Baier and Katoen, 2008). Model-checking and verification is a way to determine the behavior of the SC. For designing SC that is intended to run correctly and securely on the BC, model checking and verification is necessary. Mainly a model checking for the SC provides the necessary verification (checking the SC model against its specification) to avoid the well-known vulnerabilities of SC (4.4.3.1), and possibly to discover new SC security and misbehavior issues.

4.4.3.4 The Scientific Approach for Model Checking and Verification of Smart Contract

The *DAO* attack raised the attention of the researchers and scholars to improve and avoid the vulnerabilities and security issues of the SC. Research in (Nehai et al., 2018) uses NuSMV for the expression of the BC and SC model. The model comprises three main parts, highlighting, first, the Ethereum (kernel layer) as a distributed system for managing transactions between users. Secondly, it uses the SC (application layer) that is expressed in Solidity (Solidity, 2016) to represent them in model checking language, i.e., in NuSMV, and the third part determines the execution environment for the application (Nehai et al., 2018). This research is to verify if the SC is behaving as they are expected. For achieving this, the expected properties need to be formalized into temporal logic (Computation Tree Logic (CTL)) (Nehai et al., 2018). In case the property does not behave as requested, the model-checker produces a counterexample, that allows determining the problem and its genesis (Nehai et al., 2018). The research in (Bai et al., 2018) use SPIN (Mikk et al., 1998) for formal verification of the properties of the SC. This research contributes by formally defining SC and providing a model for SC based on PROMELA (SPIN) (Bai et al., 2018). A formal verification of SC based on user and BC behaviors is proposed by research (Abdellatif and Brousmiche, 2018). The authors highlight the fact that the previous efforts for capturing the SC vulnerabilities by documenting them and by using formal verification fail because user and BC behaviors are not considered. The authors from (Abdellatif and Brousmiche, 2018) use the non-cooperative game theory to model the transaction performed by two players. This is possible since the terms of the contract are agreed and the players act independently. A finite-state machine (FSM) based tool, named FSolidM (Mavridou, 2019), is presented in (Mavridou and Laszka, 2018), for designing secured Ethereum based SC. Further, a formal verification for SC behaviour, using F*[15], is showed in (Bhargavan et al., 2016). The security of SC is an extremely difficult task due to the openness of the BC frameworks (Bhargavan et al., 2016). The research focuses on the behavior of the SC, and proposes a framework for analyzing and verifying the functional correctness and the runtime safety of the SC by using F* (Bhargavan et al., 2016) Initially, there is given a clear guide for translating Solidity and bytecode generated from the SC, into F*. Then a detection of vulnerabilities of SC is presented. Besides, verification of the functional correctness of SC is proposed using the Solidity subset into F*, further, the framework proposed in (Bhargavan et al., 2016) analysis the byte code generated for given SC and intends to prove the equivalent running of SC in solidity level (functional level) and bytecode (runtime level).

[15] A Higher-Order Effectful Language Designed for Program Verification: `https://www.fstar-lang.org`

Model Checking Tools	Main Characteristics	Operation over SC sources	Limitations
NuSMV	model checking-functional correctness	solidity	It does not support the complete expression of a blockchian environment (Nehai et al., 2018)
F*	functional and runtime checking	solidity; bytecode	The presented tool for model-checking SC based on F*, does not support entirely the syntax features of Solidity, e.g., loops (Bhargavan et al., 2016)
BIP Framework	component based and statistical model checking	solidity and BC	The current model does not support entirely the blockchain components, e.g., mining process, block, etc. (Abdellatif and Brousmiche, 2018)
Scilla	intermediate checking	solidity	Explicit exception are not covered on this version of Scilla (Sergey et al., 2018)
EthRacer	runtime checking	bytecode	Focused only on event-order bugs by using notions of linearizability and synchronisation (Kolluri et al., 2018)
ZEUS	runtime checking	solidity and bitcode	It requires to add the policy specification of the SC (Kalra et al., 2018)
Oyente	pre-deployment SC checking	bytecode	Limited only on the bytecode, and thus losing the contextual information e.g. types, integer underflow (or overflow) (Kalra et al., 2018)

TABLE 4.3: Summary of the main approaches related to modeling and verification of SC.

In (Kolluri et al., 2018), the authors present a tool that intends to find the runtime errors of SC, in the class of bugs of event-ordering. Basically, the idea behind this research is to see if the output from SC differs when the input order of the event (functions) is changed.

The formal verification of SC is performed by using abstract interpretation and symbolic model checking, where SC is taken as input, while the output in XACML style (*XACML 3.0* n.d.) is the generation of correctness or fairness (Kalra et al., 2018). Also, an intermediate-level programming language for SC is presented in (Sergey et al., 2018), and the intention behind this research is to verify the high-level language programming language, e.g., Solidity, before deploying it into the BC. Symbolic verification of the SC is showed in (Luu et al., 2016). The values of program variables are represented by the symbolic parameters. The symbolic paths are formulas over the symbolic input, which these inputs should satisfy (Luu et al., 2016). Also, the authors from (Luu et al., 2016) implemented a tool that verifies the correctness of the SC by using the SC byte code. This tool is able also to catch the famous DAO bug (reentrancy) (Castillo, 2016) on the SC.

Ethereum is proposing Vyper environment (*Vyper* 2018) to prevent the reentrancy attack (*VyperDetails* n.d.). Another symbolic approach based on the dependency graph, that verifies the SC behavior in the report with given properties and classify it as safe/unsafe is presented in (Tsankov et al., 2018). Besides being focused on verifying the SC, on the programming level, other research highlight the verification of the SC at the runtime level, i.e., bytecode. In (Ellul and Pace, 2018) a framework for verification of the SC is proposed by combining SC and its specification. The misbehavior of the SC is identified when a specification is violated. The research from (Yang, 2018) uses Coq proof assistant (*Coq* n.d.) for formal symbolic development and verification of processes of virtual machine (VM).The intention behind this study is to prove the reliability and security of the Ethereum-based SC (Yang, 2018). The K framework has been used to build a tool that allows formal specification and analysis of the Ethereum VM bytecode of SC (Hildenbrandt et al., 2018). Another approach that applies formal verification of SC at the bytecode level by using Isabelle/HOL, is explained in (Amani et al., 2018). The bytecode is structured in a block of code, and further creating a logic for reasoning this code (Amani et al., 2018).

4.5 Synthesis: Beyond the Current State-of-the-art

This section synthesizes our research insights beyond the current state-of-the-art from a different perspective. At the genesis of our research approach, we propose a scientific method that applies BC in the SpC and Logistics in an end-to-end manner. We enlist some significant differences from the existing research, and our proposed approach showed later on.

- a) Specific Supply Chain

  The SpC of DG is complex and engages many dependencies in process organization for the stakeholders involved. It is governed entirely by authorities, and stakeholders should comply with the regulatory framework. In SpC of DG, the DG provider is responsible for following the process until the end of its lifetime. Compared to the general SpC, when the good is delivered from party A to party B, all the responsibility is delegated to party B, and party A will no longer follow the process.

- b) The existing scientific and industry proposals towards SCM and Logistics

  - Existing SpC and Logistics startups (4.2.6)

    The proposed startup-solution is intended to serve mainly these enterprises for their own business processes (use cases). In general, they do not provide any open or scientific-based approach that may be used for applying to other use cases. They mainly propose an engineering solution to overcome domain problems.

  - Our use case belongs to a regulated domain

    In the context of our use case, the regulatory framework is deeply involved, and competent authorities entirely govern the process itself. It is mandatory to consider the regulatory framework at any stage of the future system design. Applying the

existing scientific or industry process approaches by customizing them is almost impossible.

- Cross-border context

  We propose an approach that includes several countries. The cross-border context involves different stakeholders, authorities, regulation frameworks, and requirements for providing a trusted, efficient, and transparent system.

- The "trust", "transparency-traceability," and "efficiency", in the existing scientific and industrial approaches is covered partially, and none of them considers them in a single use case.

- The business process (BP)-related approach (4.2.4) is dependent on BPMN components, even-though, not all BPMN components are supported. In these approaches, stakeholders must agree on an inter-organizational (collaborative) business process before deploying it on BC. Once a new stakeholder needs to be included in BP, or additional changes are required on the BP, a new inter-organizational BP should be designed and deployed again on BC. There is no clue how the previous transaction is managed after deployment of the new inter-organizational BP. The privacy issues are not clearly addressed for the use cases where private data exchange is required.

  We propose an approach where new stakeholders can join the existing network. They are validated as digital virtual objects (twins) and adapted as part of the system. Any involved stakeholders may manage their own BC node.

- c) MDA Based Architecture for SpCDG Management

  We propose a design method that introduces high-level (abstract) concepts related to SpCDG and allow to transform them into lower-level concepts related to BC and IoT. Inspired by Model Driven Architecture (MDA), a software development approach that introduces several well-defined models. We extend MDA to explicitly introduce novel concepts such as SC, dynamic aspects, and digital components (DT). Also, it enables to model the regulatory framework documents in an abstract layer. Our design method permits to specify the TDG process using an abstract layer and then transform and refine it at the different layers towards specific target technology. The targeted system (architecture, application) can be reused in different contexts.

- d) BC and IoT integration

  To go beyond existing approaches, firstly, we propose in our approach a solution that avoids any dependency to any BC framework that needs a cryptocurrency incentive for their maintenance. Secondly, we propose a novel approach to performing secure transactions by using certified IoT devices and lightweight nodes as transaction validation. Thirdly, we propose a data flow model that is slightly different from existing approaches. Indeed, we propose to aggregate data on the lightweight nodes before sending them to the BC. Fourthly, we aim to overcome the manufacturer's (e.g., IoT solution providers) impact on designing and developing IoT based system. Fifthly, we

propose to use the BC as a full application layer in an IoT system. Finally, we propose to use open-source BCs such as Hyperledger Fabric, which is scalable, support IoT data management, privacy, and confidentiality.

- e) The research in (Perboli et al., 2018) highlights the fact that for implementing BC technology in SpC, it is required to start from "an analysis of the needs and the objectives of the different actors involved, with the aim to create a business model capable of highlighting the returns (both in economic and customer satisfaction terms) of this solution" (Perboli et al., 2018).

  We provide a design method (shown in c) for the BC-based application that enables analysis of the stakeholders' needs and requirements from an early stage. Our method considers the regulatory framework to fulfill the stakeholder's requirements. We propose a model-driven architecture (MDA) based method to manage the lifecycle of designing the BC-based application. This will facilitate designing BC systems by providing high-level models and performing the model transformations, which are more explainable and understandable. Our design method is a BC technology-agnostic that allows system designers to decide on their own to choose BC technology according to their use case requirements.

- f) The formal specification and verification aspects considered

  In the context of our research, we consider business rules (constraints) from a legal perspective (regulatory framework). We consider two different perspectives in applying formal specification and verification:

  - Formal reasoning in Common Information Model (CIM) for TDG

    We apply axiom-based reasoning by using Description Logic (DL) to the CIM defined for the TDG.

  - Temporal logic

    We extract and formulate business rules from the regulatory framework. Then, we formally specify these business rules with the help of Linear Temporal Logic (LTL). We define the model based on these specifications and perform symbolic model checking using NuSMV (Cimatti et al., 2000) to verify the model's timely properties. The defined model represents the targeted (future) smart contracts that are entirely based on the specification (as shown in 8.12).

4.6 Conclusion

In this chapter, we analyzed the current state of the art for BC and SCM, and Logistics. Our primary focuses were on trust issues, traceability context, and the current development of BC-based projects for SCM and Logistics. The analysis performed in this study intends to highlight the level of applicability of BC in SCM and Logistics. The advent of BC brought a new era of designing and developing applications for SCM and Logistics. The benefits

of using BC technology in SCM and Logistics bring a new way of designing applications whose architecture is decentralized. The scientific community and business enterprises from various domains are researching to solve their workflow management challenges by using BC technology. A considerable effort is shown in SCM's domain, and the current developments promise future improvements in the SCM and Logistics. Enlisting possible improvements in workflow management, we consider BC technology to overcome the SCM and Logistics issues. Several research types are focused on integrating BC and IoT, improving transparency along with SpC, or supporting different business cases. Enormous attention is also given to designing, verifying, and deploying SC.

Chapter 5

Context of Study, Problem Statement and Scientific Objectives

5.1 Introduction

This chapter presents the context of the thesis and definition of scientific questions (objectives). The chapter introduces the general problem statements and highlights the specific scientific objective. To highlight the problem statements, we present a broad description of the general process for the TDG and introduce how the existing processes of TDG are organized. Further, we show how it is possible to improve the efficiency of these processes using emerging IT technologies. Identifying the current challenges and issues around organizing the process of TDG allowed us to define our scientific objective (questions). A set of research questions is then presented to highlight all research challenges and motivate our scientific approach. We further define the use case in the transport and management of medical waste (TMMW) for more concrete, real-world research challenges. We identify concepts, processes, and workflow aspects.

5.2 The Current TDG Organizational Aspects

There are several specific contractual terms to fulfill in the business contracts between the "DG Providers" and "DG Receiver", that collects and performs treatment of DG, on one hand, and the "Transporter" responsible for transportation of these goods, on the other. The "Transporter" entity may have several subcontractors, which cooperate for TDG. Any activity (business contract) with the TDG requires prior authorization from relevant competent authorities due to a legal requirement because of the DG specificity. It becomes more complicated when the transport has to cross borders (5.1) for DG delivery purposes. These conditions are in compliance with legal terms dedicated to TDG in local and international operations (UNECE, 2017). A prior request is forwarded to the different competent local authorities, which consider "DG Provider" (or "Transporter") requests. The actual local authorities in each country examine the request for TDG authorization. These authorities will authorize TDG based on the specific conditions to be fulfilled for the given request. The whole operation should satisfy all local laws and international procedures related to TDG (Imeri et al., 2017).

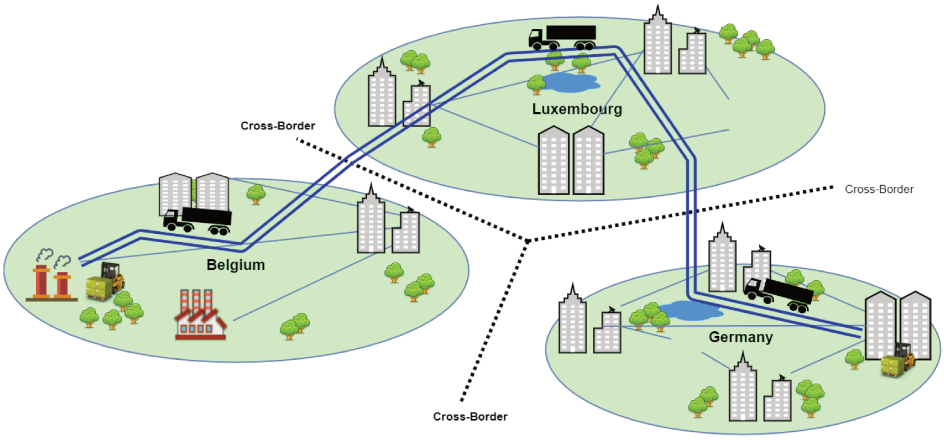

FIGURE 5.1: The example of DG transport map in a cross-border context.

The DG will arrive at the "DG Receiver" only after the final examination of its current state following the transportation process. The "DG Receiver" usually provides these reviews (in cooperation with the local authorities), which check whether the state of the DG is entirely in accordance with the compliance rules, prior to releasing these DG for treatment. If any of the compliance rules are not satisfied, the safety procedures are applied, which engage specific work to be performed by the transporter to continue with the process (*"Regulation (EC) No 1013/2006"* (Commission and Parliament, 2006)).

The TDG process events generate a considerable amount of data. This data may contain confidential information, which should not be disclosed to any unauthorized third parties. The timestamp of the DG movement, the type of these goods, the current warehousing location, the route to be taken, and other related information expose human safety and security, living organisms, properties, and the environment to risk.

TDG has several challenges closely related to the DG classes. Accidents with DG during transportation may expose a high risk to human beings, private and public properties, and the environment as it usually passes through both urban and non-urban areas. For this reason, the process of TDG is strongly governed by specific rules and regulations, i.e., a "regulatory framework", that are determined by the competent authorities representing the countries involved. The process of transport and management of DG by competent local and international authorities, which impose these regulations. The regulatory framework requires the strict compliance of the process via its workflow (Imeri et al., 2017). As the most suitable means of TDG is using roads (due to the low costs, compared to other modes of transport), the shipping (transport) organization usually selects the most cost-effective route. This usually exposes problems because these routes pass through populated areas (Torretta et al., 2017). In the context of globalization, transporting DG to some specific destinations requires crossing the borders of several countries, as depicted in Figure 5.1, and therefore, the process is automatically extended to the international level. The cross-border situation increases and extends the challenges in TDG in terms of compliance, information sharing, efficiency, and trust (detailed in section 5.5).

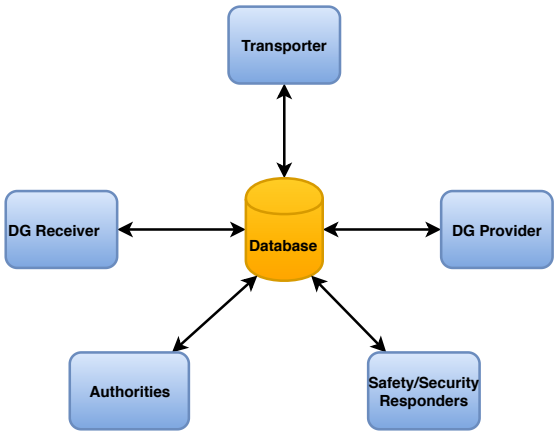

FIGURE 5.2: The classic "client-server" approach for the TDG process management (Imeri and Khadraoui, 2018).

5.2.1 Today's Enhanced Process Organization in TDG

The emergence of new technological solutions has greatly influenced the improvement of the business sector in different domains. The design of new service-oriented solutions has permitted business improvements by adding value to the service deliveries to customers (Tallon et al., 2000). The perceived benefits are in the area of a better management of the process by the involved stakeholders, e.g., "DG Provider", "Transporter" , "DG Receiver", "Authorities," and "Safety/Security Responders", which allow them to exchange information in a more flexible way.

With the involvement of technological components, the process of TDG is enhanced compared to its beginnings. The processes are continuously improved by managing the information more efficiently and reducing delays in this process. The technological components enable the usability of standard user interfaces to exchange information between the parties involved in this process. Figure 5.2 presents the general schema for the associated parties in this process. The presented architecture is the classic "client-server" communication mode (Hanson, 2000). In this mode of communication, for a particular case in TDG, the "DG Provider" provides specific information for the DG subject to transportation. At the same level, the "Transporter" is granted access to this information. This enables coordination between them for the communication of valuable information. Furthermore, this architecture allows the "DG Receiver" entity, "Safety/Security Responders," and "Authorities" to have access to a limited set of information (based on contractual clauses) regarding the DG, which are in the process of transportation. Even though technologies exist for this, mainly the TDG is still organized rather manually (paper-based) due to lack of trust, information security, and other drawbacks not supported by the classic technologies (client-server).

5.3 Transportation and Management of Medical Waste (TMMW)

This section aims to present the use case of the transport and management of medical waste (TMMW). We use the ADR's definition of medical waste (MW) according to the ADR (UNECE,

2017). Further, we define the *use case* by examining the TMMW process, from an information point of view, including the applied regulatory framework. We perform workflow analysis, which allows us to identify all necessary components, such as the processes required to compose the entire TMMW, the stakeholders involved in the TMMW, and their interactions.

5.3.1 Toxic and Infectious Substances

The subject of this study is MW (DG), which belongs to Class 6, "Toxic and Infectious Substances", in the classification of the DG shown in Table (2.1).

The ADR criteria for *toxic substances* covers any substance that can damage human health or death by inhalation, absorption, or ingestion, namely *"Organic liquids", "Solids", "Solids used as pesticides", "Inorganic liquids", "Oxidizing substances", "Corrosive substances"*, etc. (**Section 2.2.61**; UNECE, 2017).

Infectious substances are any substances that contain pathogens (bacteria, viruses, fungi, parasites, or many others) or the substances that might contain prions that can cause human diseases, namely *"infectious substance that affecting humans", "medical or clinical waste", types of "biological waste"* (**Section 2.2.62**; UNECE, 2017).

5.3.2 The Use Case of the Transport and Management of Medical Waste (TMMW)

First, we present the concepts used in the use case definition that we will show later on. To select the use case, we examined several possibilities in line with TDG. In cooperation with the competent authorities and after several meetings with the stakeholders responsible for TDG at the national level in Luxembourg, we selected the use case of transport and management of medical waste (TMMW). The regulatory framework presented in Table 5.1 is used to describe these concepts:

- *Waste (W)* is any material or substance that is generated with certain processes, and it must be disposed of or recovered in a certain manner (Commission and Parliament, 2006; UNEP, 1989).
- *Medical Waste (MW)* is any substance that is generated from medical establishments such as hospitals, biological or chemical laboratories, pharmaceutical companies, etc. (UNEP, 1989).
- *Management of Waste (MW)* refers to the collection of waste from different sources, and the transport and disposal (or recovery) of waste. This process also includes the handling procedures after disposal (or recovery) of the waste (UNEP, 1989).
- *Waste Disposal* means any operation for waste processing, such as "land treatment", "deposit into or onto land", "incineration at sea", and other ways as described in Annex IV in (UNEP, 1989).
- *Waste Recovery* means any operation for waste recovery such as the "recycling of organic substances", "use as a fuel", "Regeneration of acids or bases", and other ways, as described in Annex 1B in (OECD, 2008).

TABLE 5.1: Laws, directives and regulations for transport and management of Waste, MW and DG.

Directive/Regulation	Regulative body	Description
Directive 2008/98/EC	European Commission and Parliament	Directive on Waste
Regulation (EC) No 1013/2006	European Commission and Parliament	Shipments of Waste
Basel Convention	UNEP	Basel Convention on the control of transboundary movements of hazardous wastes and their disposal
Law of 21 March 2012	Government of Luxembourg	Waste Management
C(2001)107	OECD	Control of transboundary movements of waste destined for a recovery operation

- *Competent Authority* represents any authority that is responsible for governing the process of the transport and management of waste (UNEP, 1989).
- *Cross-border (transborder)* means moving to an area under the jurisdiction of another state, for example, transporting medical waste from Luxembourg to France (UNEP, 1989).

5.3.3 Regulatory Framework Applied for the Transport and Management of Waste

National and international laws regulate the transport and management of waste, including MW. General waste is not considered DG unless it belongs to a DG class, such as MW. Despite not being considered DG (non-hazardous waste, e.g., household or office waste), the transport and management of waste requires an authorization process for the cross-border transport of waste, and all stakeholders must comply with the regulatory framework. Table 5.1, shows these regulatory frameworks for the transport of waste.

The regulation that is applied to the transport (shipment) of waste is *"Regulation (EC) No 1013/2006"*, and this serves as the main regulatory framework in waste shipment. The procedures for the transport of waste, control regimes, classification of waste as dangerous and non-dangerous are clearly defined in this legal document. Furthermore, it uses (refers to) other legal documents, e.g., the Basel Convention (UNEP, 1989), as shown in Figure 5.3. This regulation must be applied by the member states of the European Union[1] when importing waste to other member states (or from a third-party country), exporting waste to third-party countries. It should also be applied if the transport of waste is in transit in the EU area (Commission and Parliament, 2006).

[1] Also, it must be adapted from the European Economic Area.

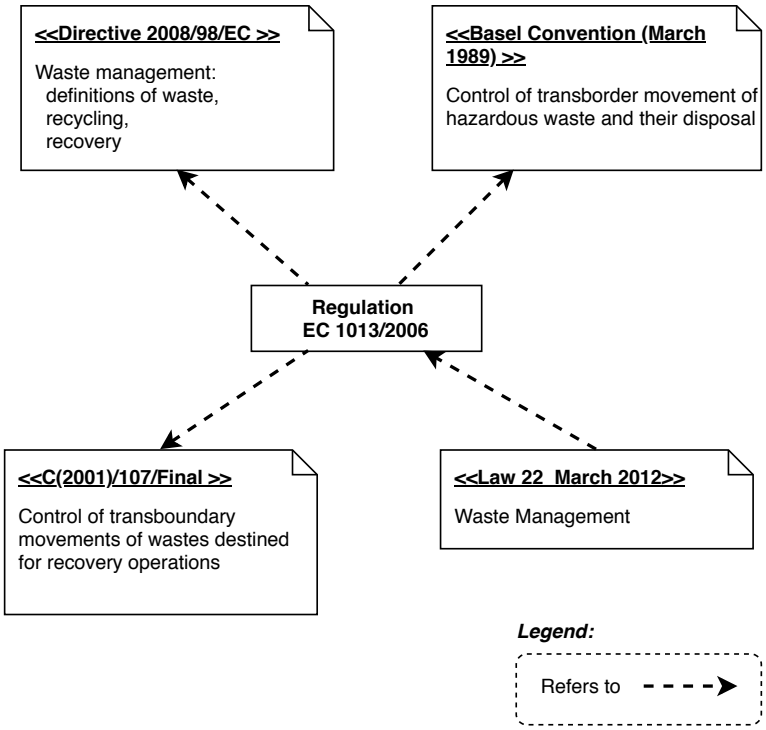

FIGURE 5.3: The interaction and references of legal documents for TMMW.

The *Directive 2008/98/EC*[2] provides a clear definition of the concepts related to the transport and management of waste, such as "waste", "producer", "bio-waste", "dealer", "broker", etc. (Commission, 2008)

The *Basel Convention* clearly classifies waste as dangerous waste (List A) and non-dangerous ("other waste") waste (List B). This convention provides information regarding the control of the cross-border transport (movement) of waste, a proper way of exchanging information, and the guidelines for its disposal (UNEP, 1989).

The *C(2001)107*, provides specific legal information for cross-border controlling of the waste that is destined for recovery purposes (OECD, 2008). For any cross-border transport of waste for recovery purposes, it should be organized according to international transport engagements. The waste recovery should be performed following national laws, general practices, and regulations under which the treatment facility operates. Furthermore, this regulation is an official document on the cross-border transport of waste, and provides specific information on completing the notification (request for transport authorization) process (OECD, 2008).

The *Law of 21 March 2012*, represents the national regulation for waste management at the national level, e.g., Luxembourg. This law provides legal information for the organization of the transport and waste management process at the local level and also at the international level, by also referring to *"Regulation (EC) No 1013/2006"*.

[2]This regulation is amended from Directive 2006/12/EC.

5.3.4 Description of the TMMW Process

For the transport of dangerous goods (TDG), i.e., medical waste (MW) (5.3.1) there are several stakeholders involved in this process. These stakeholders compose a consortium of involved parties at the national and international level. This process is delicate because of the risks for human beings, living organisms, and the environment. Typically, the process is covered by a regulatory framework that organizes this process at the national and international levels. The applicability of the regulatory framework gives another level of complexity in this process. For the stakeholder involved in this process, compliance with the regulatory framework is strictly required by the authorities that govern this process. This implies that the involved stakeholders should follow procedures that indicate the process of TDG, i.e., TMMW, before starting this process, then during the process, and then after finishing the process. For the TMMW from the point of origin to the destination, many processes must be followed, which generate sensitive information to be shared among the stakeholders. Examples of these processes are the prior certification of stakeholders from the authorities, the authorization process for transport of MW, the transport (move) process of MW, contractual terms and possible concerns between stakeholders, types of goods to be transported, the timestamp of goods movement, and the warehousing.

5.3.5 Workflow Analysis for the TMMW Process

The objective of this section is to present the workflow analysis over the TMMW process. The objective of these analyses is to discover the involved stakeholders, processes, sub-processes, and the administrative procedures shown in the TMMW. The workflow analysis for the process of TMMW is based on *"Regulation (EC) No 1013/2006"*.

Initially we discover the stakeholders involved in the TMMW process:

- **Waste Producers:** This entity represents any stakeholders that produce waste, or any entity that changes the nature of the waste by mixing, pre-processing, or composting the waste.
- **Waste Traders or Brokers:** The trader (broker) represents any professional in a commercial business that requires "waste collection or transport" on behalf of third parties.
- **Waste Collectors:** Any entity that collects and transports waste, including the preliminary storage of waste, and then further transports this waste for treatment purposes.
- **Waste Transporters:** Transport waste to and from a specific location. This entity can be a "Waste Collector" or a specialized waste transporter.
- **Emergency Response:** In the event of accidents with DG, including waste, emergency activity is required from the "Emergency Responders" performing a specific activity to prevent catastrophic consequences.
- **Treatment facilities (Waste receiver[3]):** Perform waste treatment activities, such as disposal and recovery based on the type of waste.

[3]The *"Regulation (EC) No 1013/2006"*, defines the "Waste receiver" entity as "consignee".

- **Authorities:** Govern the process of transport of waste by certifying stakeholders, granting permission for waste transport, and also aiming to monitor the process of waste transport.

Three different phases determine the organization of the TMMW process. Figure 5.4 shows these phases and contains all the significant steps for fulfilling an end-to-end TMMW process.

The *first* phase is the *planning phase*, which is subject to fulfilling *conditions before* the process starts. This phase contains activities for stakeholder certification, the process of requiring authorization for the transport of MW (movement of MW). These activities are presented in the "Condition Before" in Figure 5.4. The *second* phase is the transport phase (shown in the UML sequence diagram "Condition During" in Figure 5.4), and it is the subject of fulfilling the conditions during the MW's transport process, such as monitoring the movement of MW. This phase incorporates the pre-planning of the MW transport, providing the necessary information for the authorities, monitoring the process of transport of MW, and providing the essential documentation. Finally, the *last* phase of the TMMW process covers the procedure of receiving the MW (shown in "Condition After" in Figure 5.4). Then, a clear procedure is required for the treatment of MW and sending back the *certificate* for this treatment.

From these workflow analyses, we identified the main processes that are part of the *"global"* TMMW process. The workflow presented on these processes below shows how these processes are organized and performed by the involved stakeholders. The paragraphs below shows the main characteristics of these processes, their impact on the TMMW and applicability on the global TMMW process.

P0: Certification of stakeholders. *Applied for Conditions: Before TMMW*

Any stakeholder that intends to perform an activity (purchasing, selling, reselling, collecting, processing) with waste should require certification. This certification means that they have permission to perform waste-related activities. Certification areas are "Waste Trader or Broker", "Waste Transporter or Collector", and "Waste Treatment Facility". The certification procedure begins by initially fulfilling the pre-requisites and sending documents to the competent local authorities. For administrative reasons, the competent authorities might require additional documents for specific requests before granting or denying permission. Mainly, the stakeholders' certification process is governed at the national level, under the jurisdiction of the country where these stakeholders are operating.

P1: Authorization for TMMW (Notification process). *Applied for Conditions: Before TMMW*

For any entity (person or organization) required to transport waste, an authorization (notification) is mandatory. The process of requiring authorization for the transport of MW is known as *"notification"* in the legal and procedural documents, e.g., *"Regulation (EC) No 1013/2006", Article 4*. This authorization is required at both the national and international levels. The *notifier* is the person or entity (organization), e.g., waste producer, certified waste collector, a certified waste dealer (broker), under the jurisdiction of the country in which it

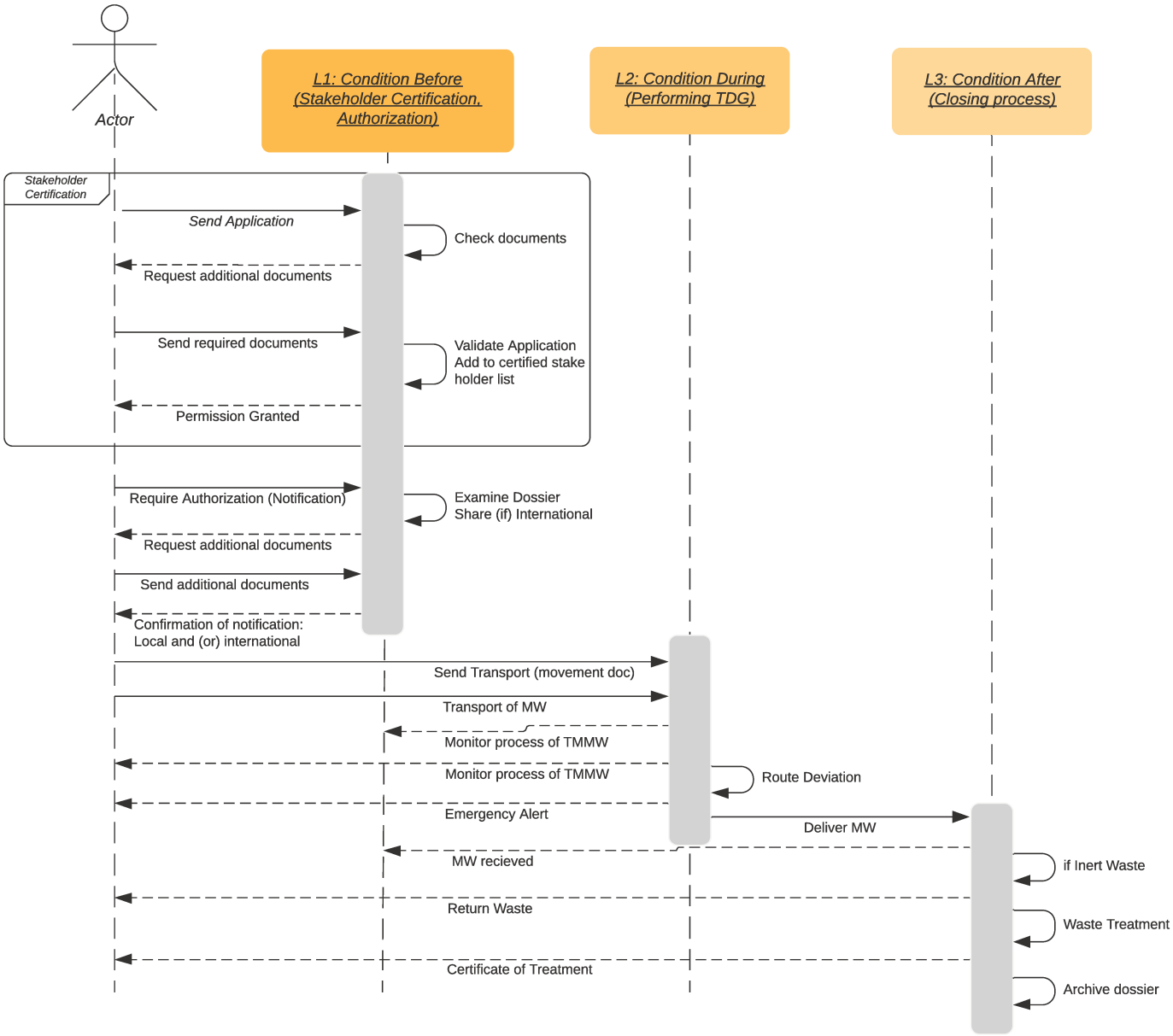

FIGURE 5.4: The UML sequence diagram of different processes involved in TMMW.

operates. For the transport of waste, the authorization process is required by the *notifier* by sending the request to the competent local authority, e.g., "Administration of Environment" [4]. The competent local authority will examine the request received, i.e., the "dossier". If any relevant document is missing, the *notifier* must send these additional documents. Once the "dossier" is completed, and in the event of an international request, the competent local authority will submit it to the destination authorities and share the same "dossier" with the transit authorities. To obtain the authorization, the destination and all transit authorities should respond positively within the pre-defined timeline.

P2: The process of transport (move) of the MW. *Applied for Conditions: During TMMW*

This process is enabled only after possessing authorization for the transport of waste. The authorization process might have specific conditions expressed by the local government, relevant transit authorities, and destination authorities. The process of transport might start only if these conditions are fulfilled. Furthermore, pre-defined transport (move) should be notified to the authorities before the transport starts. All the transport documents should be completed, and in the event of an intermediate stop, it should be reported.

P3: The process after delivery of the MW. *Applied for Conditions: After TMMW*

When the waste is delivered to the destination, the waste receiver notifies the waste collector (trader, broker), also known as *notifier*, and the competent authorities of the waste arrival. The waste receiver (treatment facility) should further provide information regarding the treatment (recovery or disposal) procedure if the treatment is performed in an intermediate way or immediately. An *intermediate* way means the processing is performed in third-party countries, which requires a new notification and still maintains the current notification. If the treatment is performed in another EU member country, another notification is still required. If the treatment is done *immediately* at the same place (destination), then a "certificate" for this treatment should be sent to the *notifier* and *competent authorities*. The waste treatment should be accomplished within one calendar year of the waste being received.

5.3.6 An End-to-End Example for TMMW

We show the sequential steps of an example of elaborating the processes (P0 - P3) expressed above.

1. Before even starting a TMMW process, the stakeholders involved in this process, e.g., Waste Collector, Waste Broker, or Waste Traders, should be certified by the competent authorities of the origin country.

2. The process starts when the MW producer has gathered a significant amount of MW. As required, the Waste producer prepares the MW by classifying it according to DG classes, and packing and labeling it. The waste collector picks up the MW from the waste producer and moves (transport) it to its warehouse.

[4] Administration de l'environnement: https://aev.gouvernement.lu/fr.html

3. There may be different MW producers. The MW collector bundles the waste according to its classification (DG Class).

4. *Sending the notification*. Once the dossier is completed, according to the requirements requested for the notification process, the MW collector sends the notification dossier to the competent authorities of the origin country.

5. The competent authorities of the origin country should *share the dossier* with the competent authorities of the transit countries the MW will pass through when transporting it and with the destination country. In our example, the transit countries are Germany and Austria, while Hungary is a destination country. The dossier will mainly include information regarding the transporter, the quantity, planned number of trips for the given quantity, the type of MW, the information for the consignee, i.e., the waste treatment facilities (Waste Receiver), and the approximate date of the first and last shipments.

6. The competent authorities of each country involved should check the completeness of the *dossier*. The destination country also checks the consistency of information regarding the MW Receiver (waste facility treatment).

7. The involved authorities *validate the dossier* and send the decision (including their conditions for it) to the competent authorities of the origin country.

8. If MW's transport is authorized from all the countries involved, then the dossier is considered to be approved, and the shipper (MW Collector/Transporter) receives the notification.

9. The *pre-notification* for the shipment (transport): This shipper (MW Collector/Transporter) should notify the authorities at the latest three days before the transport date. The competent national authorities notify the other competent authorities involved. Meanwhile, the MW Collector/Transporter proceeds with MW transport preparation according to the pre-defined procedures.

10. The MW Collector/Transporter monitors the movement of the waste, and in the event of any unpredicted event, e.g., changing the routes during the transport, then reacts to resolve the process according to in the notification.

11. MW received. The treatment facilities will notify all authorities involved to receive the waste within the pre-defined time, e.g., within three days from the reception date.

12. MW treatment. The treatment facility either treats the waste temporarily or applies procedures for final treatment within a year, as required. If the treatment is to be done in another third-party country, a new notification is required.

13. The treatment facility should provide notification for the treatment of the MW. The MW Collector and all authorities involved should receive the MW treatment certificate.

14. The Waste collector and MW facility keep documentation regarding this process for later reference and reporting.

5.4 Background: Definition of the Used Concepts and Terminology

This section presents the way we address the information security, trust and transparency in this thesis. Furthermore, it presents some basic concepts and terminologies that we use later to define and describe our approach.

Terminologies used for the TDG and related use case

- *DG* - Any dangerous goods, including medical waste (MW).
- *TMDG* - Used in the application of the approach and focuses on transport and management issues.
- *Waste (W)* - General term that describes waste.
- *Medical Waste (MW)* - Waste (W) related to the use case.
- *TMMW* - Term used particularly for the use case, focusing on transport and management issues of MW.

When referring to DG, we include MW goods. Also, the term TMDG includes also TMMW. Similarly, we use the term "DG Provider" to refer to Waste Producers, Waste Brokers (Trader) or Collectors, and the term DG "Transporter" for Waste Transporters or (Waste Collectors). We use the term "DG Receiver" for Waste Receivers (Treatment Facilities).

5.4.1 Information Security. Secure Information Sharing

To secure information sharing in the TDG, we propose**role-based access control** and consider **BC principles for information security** (decentralized consensus (3.4), hashing, public-key infrastructure (A.7)). For any piece of information in TDG, the cryptographic mechanism is applied before sharing it. The shared information is validated based on an applied consensus mechanism and stored immutably in a distributed-decentralized ledger. Beyond the BC principles, we propose role-based access control (7.5.5) with the help of SC to enhance the information security based on the access policy of the roles (users).

5.4.2 Trust

The TMDG process analysis highlights the concern of stakeholders regarding trust in the TMDG process. "Trust" in the TMDG is enforced by a reliable system consisting of:

- **Secure information sharing**

  Relying on information security (as presented in 5.4.1) attributes for trust enhancement.

- **Certified stakeholders**

  The stakeholders involved in the process of TMDG should be certified. Before entering into the "ecosystem" of the TMDG, all participants should go through the certification process. This certification will be done in compliance with the national regulations in each country in which the approach (service) is in operation. Furthermore, this does not mean a technological certification (e.g., fingerprint ID). This certification directly involves the competent authorities to support this certification. "Certification" covers all procedures for operating as a stakeholder in the process of TMDG.

- **Enforced compliance standards (regulatory frameworks)**

  The TMDG is ruled by the regulatory frameworks applied in this process. At any step, the process must be compliant with the regulatory framework, and in addition, we will use a Smart Contract (SC) to manage this process and cover the compliance. SC will react by denying or approving any further step on the TDG.

- **Reliability**

  In the context of system organization, none of the stakeholders hold the entire system data, and a third party will not be able to divulge (or own) the system entirely. The proposed approach is a system that will be **reliable** and a single source of truth. It will be unified (with data standards, e.g., EDI) and easily accessible from the authorized parties. Our approach consists of using BC properties to support data security and IoT devices to enhance the process of TMDG by providing authentic information.

- **Auditability of the TMDG process**

  The process of TMDG should be auditable by non-involved stakeholders e.g., Authorities, General Public (with permissions), and involved stakeholders (for their internal usage). The whole process must be auditable from the beginning (certification, authorization), at any point (destination) of the DG movement, and up to the end of the process.

- **Traceability**

  "Following all the information, from the point of origin by starting the process of TMDG and continuously updating information by tracking the physical movements of goods, our conceptual approach ensures a reliable traceability mechanism. For the context of traceability, data immutability is essential. The data retrieved time after time from SC up to finalizing of the process is stored in the BC, which is considered immutable and valorizes the process of traceability" (Imeri and Khadraoui, 2018).

- **Process Digitalizing**

  Our approach proposes digitalizing the process of TMDG, which will allow a better management of information. Digitalization allows stakeholders and authorities to govern this process at any level.

  Why does TDG digitalizing matter?

In the paper base mode, usually, when the truck arrives, the physical and environmental parameters, e.g., temperature and other parameters, are written down on paper. After a certain time, there is no mechanism to prove that these parameters were written on paper or proof if we want to trace back that particular process. We apply IoT to add these data and store them in BC in order to refer to them at any time.

Technological advancement allows at last two levels of verification, for example, when goods arrive at the destination, the "driver" signs to confirm this and at the same time, an IoT device verifies the arrival of goods. This is known as two-level authentication of "goods", "trucks", "IoT" devices, and all necessary objects that are part of the ecosystem of TMDG.

5.4.3 Transparency

Our approach intends to clarify and provide necessary information for the involved stakeholders to verify the end-to-end treatment of the DG. The transparency intends to clarify the involvement of stakeholders, i.e., "Who is providing the DG?", "Who is transporting DG?", "Where is the final destination of the DG?", "What is the treatment method for the DG?". Moreover, our approach provides a possibility for public (civilians) accessibility since more and more people are willing to know what is happening with the DG, e.g., medical waste, nuclear waste, and how they are treated. Our approach enables civilians to be informed about the process of TMDG at any time and with any level information available (based on the regulatory framework for public access to information).

Another critical component provided by our approach is related to the management of the DG, in the sense that the quantities of DG introduced are calculated in an end-to-end manner. For example, this means that when a stakeholder transports a quantity of DG to a specific destination, e.g., a "Warehouse", and unloads them there, our approach keeps these DG quantities with that "Warehouse". In some cases, these DG are entirely processed at this same "Warehouse", and in others, they are transported to another destination. Furthermore, when the transporter makes intermediate stops, our approach calculates the quantities both when the truck enters the "Warehouse" and when it leaves the "Warehouse". If there is a difference, our approach focuses its attention on what happened to the remaining amount of DG.

5.4.4 Safety

In the context of our approach, "Safety" means any pre or postcondition for providing a "safe" process of TMDG. This signifies that the main concern (care) is human safety, living organisms, property, and the environment. The source of safety in TMDG emerges from the regulatory framework, and it should be respected in order to ensure a compliant process. Subject to transport needs, the identification, surveillance, and risk-degree management of the DG must be provided and continuously monitored (maintained) to ensure a safe TMDG process. Our approach concentrates on safety conditions, as follows:

- Human Safety:

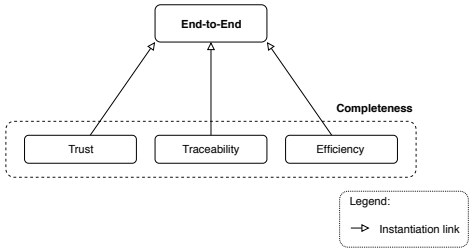

FIGURE 5.5: The end-to-end concepts for TMDG.

1. Avoid any possible damage from DG. All precautions should be announced in advance by the stakeholders, e.g., "DG Provider", and stored digitally.
2. Avoid any long-term exposition to the DG (to drivers or other involved employees) that might cause illness in the future.

The area of "human safety" presents a stakeholder's perspective of *shared responsibility* to employees and the general public. The shared responsibility means that, for example, claims from stakeholders for the DG provided remains unchanged. In the event of an accident, infection with these DG, or any misinformation toward the employee, information concerning this event may be revealed and easily verified by the authorities. As we provide a digitalized system with immutable information storage, the stakeholders' claims remain unchanged in the event of an accident with DG. The shared responsibility concepts ensures transparency over the process of TMDG.

- Living organisms:

  The treatment of DG requires close attention from the stakeholders that are certified for this process. Any non-appropriate treatment of DG or disposal of them in a non-appropriate place has the consequence of damaging living organisms. Our approach seeks to *certify the treatment* of DG and to avoid any damage to domestic and wild organisms, from the *DG disposed of on open land*.

- Properties (public or private):

  Take all precautions to avoid damage to public and private property.

- Environment:

  The environment should be protected from any exposure to DG in an open environment, e.g., throwing away medical or organic waste in any place that might expose a certain risk level.

5.4.5 Completeness and End-To-End Concept

In the context of our approach, the concept of *completeness* indicates the consideration of trust, transparency, and efficiency in an end-to-end manner.

The *end-to-end* concept covers all the above mentioned challenges (1.3) in the process of TMDG. The end-to-end concept means that we consider the condition before, during,

and after TMDG. Trust, transparency, traceability, and efficiency are associated with the end-to-end concept, indicating that we consider these at all levels in TMDG. Furthermore, these concepts are a subset of *completeness* as presented in Figure 5.5.

5.5 TDG Challenges and Research Objectives

Based on our previous analysis work, the study of the general process of TDG (also the concrete use case for TMMW), and in cooperation with stakeholders that transport and manage DG (at the local level, i.e., Luxembourg), we discovered general challenges in the TDG. From these challenges, we found niche problem areas in information sharing and security, lack of trust in stakeholder cooperation, reliance on the regulatory framework for the TDG systems, and process transparency. Later on, we define the research question more precisely, with these issues being at the source of the thesis's research objectives.

5.5.1 Efficiency Issues

In the context of TDG, many stakeholders cooperate to ensure the successful fulfillment of the transport process. This process is very delicate because of the risks for the environment and human life. It reaches a certain complication level due to the many **standards and strict local and international regulations**. For the TDG from the point of origin to the point of destination, many processes must be followed, which generate sensitive information that needs to be shared among the stakeholders. Examples of this information are prior (and post) administrative processes, contractual business issues, varieties of goods to be transported, the timestamp of DG movement, warehousing, mode of transport, and DG final treatments. The current way of managing the transport and treatment of DG is not fully transparent or efficient. Furthermore, this process is administered in a traditional paper-based or semi-digitalized way (phone, email, fax, or centralized databases). Although in some cases there is an initiative to define a TDG-related system (centralized approach), there is also deep concern about interoperability of their systems with stakeholders using specific (different) technologies. Consequently, the current process of TDG encounters many issues related to efficiency.

5.5.2 Trust Issues and Information Security

The issues of trust in the area of supply chain management are an immense concern among the stakeholders cooperating in the supply chain. For a sustainable process of logistics and transportation, efficient information-sharing is considered critical. The current systems that serve as the basis of SpC operations have several drawbacks in terms of data security and trust among stakeholders, who share information as part of their cooperation. Information is shared in a paper-based or semi-digitalized way due to lack of trust or the risk of competitive disadvantages in the current systems. Concerning the TDG, the process generates specific information, which must be shared with the stakeholders involved in this process. This information is considered sensitive because it may contain the timestamp of the movement of

goods, information related to the goods, contractual business details, etc., and unauthorized parties should not be able to access them. At any level, the process should remain transparent between stakeholders, with immutable properties on data sharing, and the whole process should be auditable. The concern remains that the current way of sharing and managing information does not guarantee the immutability of the information provided by involved stakeholders.

5.5.2.1 Security Issues with IoT

The use of IoT devices significantly improves the quality of the process for TDG since it allows monitoring, traceability, and the triggering of appropriate actions in the event of situations, i.e., accidents or other disturbance to the process of TDG.

The current TDG-related systems are mainly designed as a centralized system hosted in a private data center or in the cloud (Ammar et al., 2018), and they remain the only point of reference for data exchange. IoT frameworks such as Amazon Web Services (AWS) (*AWS IoT* n.d.), Salesforce (*IoT-Salesforce* n.d.) and any many others [5], do not provide any formal way to verify the reliability and integrity of the stored data. Mainly, such frameworks use their cloud storage to store client data.

In the context of TDG, where "nuclear materials or nuclear waste, infectious materials, e.g., medical or biological waste" might be among the transported substance, the security, confidentiality, auditing, and monitoring of processes in real-time are extremely important, as is the ability to respond to the following question: *"Why do we need to secure the information transmitted by IoT devices ?"*.

The process of TDG is essentially an international activity that crosses the borders of countries whose stakeholders are involved. For this process, different international and local regulatory frameworks are applied, and usually, the stakeholders involved are those with a big market reputation (Imeri et al., 2017). In the event of an accident or irregular process in TDG, the secured information captured from the IoT devices is currently not immutable, and this allows the possibility of big market players impacting the process by tampering with the information. The design of current technologies that support IoT data storage does not guarantee this level of data integrity (Medaglia and Serbanati, 2010; Yang et al., 2017; Ammar et al., 2018). To ensure the objectives of such a system are met, one should answer the following questions:

- How can we remove the single point of failure problem of existing systems?

- How can we efficiently and in nearly real-time monitor the transported DG's state using IoT devices?

- How can we securely store (immutably) the information generated by IoT devices?

[5] There are many other IoT frameworks such as Microsoft Azure, IBM Watson IoT, Intel IoT, etc., see `https://www.educba.com/iot-framework/`.

5.5.3 Traceability Issues

Following the DG movement from the point of origin to the destination point remains a challenge using the current systems, which manage the process on which TDG is based. The stakeholders cannot track and trace the movement of dangerous goods at any time, and the system that provides a certain level of traceability is centralized and is largely ineffective against manipulation. When the DG cross through many countries, current TDG process management systems are not able to maintain the end-to-end traceability. In the past, cases were identified where a contracted transport stakeholder was sub-contracting other transport companies, making further traceability difficult and vague for the authorities and "DG providers". Identifying such cases allows us to expose the problems of disposing of waste on open land or inappropriate management that might result in accidents.

5.5.4 Interoperability Issues

The systems that are currently developed for the SCM, in general, are based on specific, classic technologies, and stakeholders might have different kinds of technologies for their services. This forces stakeholders to adapt to such a system and is the main difficulty for the new market actors (stakeholders) who are required to have expertise in setting up software technologies in order to exchange specific information with the majority of stakeholders on the ecosystem. This imposes interoperability issues between the stakeholders of the SCM.

5.5.5 Problem Statement and Scientific Objectives

- In the context of SCM for the transportation of dangerous goods (TDG), i.e., medical waste (or organic waste) many stakeholders need to cooperate to fulfill the process of transportation.
- This process is very delicate because of the associated risks (for the environment and human life).
- The process is very complex due to the many standards and strict local and international policies in place.
- This process requires the management of sensitive information that needs verified, securely stored and shared among stakeholders.

5.5.6 Identified Challenges

- How to digitalize the process (as much as possible if not entirely) by removing/reducing human interventions and related constraints?
- How to translate paper-based regulation articles (national and international level, e.g., ADR) into digital, formally verified workflow process and enforce them in the SpC?
- How to collect and share information from the physical and logical entities/stakeholders involved in the process in a secure and trustable way?

- How to ensure traceability and transparency along the supply chain, even in the transborder context?
- How to automatically detect anomalies and violations in the process of the transportation of dangerous goods?
- How to automatically detect/prevent disasters and automatically manage the dissemination of alarms to the appropriate actors (authorities, first responders, etc.)?

5.6 Conclusion

In this chapter, we analyzed the general problem of TDG and identified the research challenges. The process of TDG is presented in general by highlighting the current way the TDG is organized. We further identify the general issues and present our research objectives. Furthermore, we present a use case related to the transport and management of medical waste (TMMW), which we will develop and refer to by presenting our design method. Inspired by the research questions and in order to resolve the issues raised, we propose a design method that responds to the research questions, and we present the design method for the general system design.

Chapter 6

Scientific Approach for Designing the Blockchain-Based System

6.1 Introduction

In this chapter, the objective is to present our proposed scientific method to design a blockchain-based TDG management system. The proposed design method is based on the so called model-driven architecture (MDA) approach and includes a new architecture-based method for designing software systems, including blockchain-based systems. This method enables the definition of different models to support the specification and the implementation of the system. These models are technology-independent (agnostic), which further allow our proposed architecture to be more robust, flexible, and agile, in the sense that it could easily respond to changes in the business process, regulatory framework, and other provisions that further impact the technological components, e.g., new blockchain framework. In short, in our design method, the entire business logic is kept outside of the technology dependence, making the purpose of the application (architecture, system, or solution) more powerful since it can be reused in different contexts. The MDA itself presents an approach and not a method (Rhazali et al., 2016). To exploit the MDA approach, a design method must be defined, and this is the aim of our research. We propose to leverage the general MDA approach with several additional layers to address our challenges specifically.

The motivation behind the definition of such a method is the fact that designing a blockchain-based solution by directly coding the business logic yields many issues in the areas of solution maintenance, adapting new business logic changes (since it is statically coded), and disability for using business logic in different contexts. In particular, designing and developing systems dependent on regulated domains (regulatory framework), as in the case of TDG, might face plenty of similar challenges if not following an appropriate design method. From the engineering side, understanding coding is more challenging than first understanding the model itself, then following principles to generate code.

In the second section of this chapter, we present the first layer of the design method. This introduces the common information model for the TDG (CIM-TDG), which allows us to express a knowledge representation model.

The terminology used for architecture, modeling language, and the relationship between meta-model and model is shown in Appendix A.5.

6.2 Design Method: Blockchain Based System Model (DG-BCSM)

Our design method is composed of seven different layers and is performed in sequential order. The first layer in our method describes the **common information model for TDG (CIM-TDG)** for the general TDG, including the use case that we are examining and developing. The second layer presents a **platform-independent meta-model (PIMM)** consists of the regulatory framework explored in the previous step (L0). The third layer presents a **platform-independent model (PIM)** that aims to perform operational analysis and discover functional and non-functional requirements. The fourth layer is composed of a **platform-independent smart contract model (PISCM)**, presenting an independent technology model for designing a smart contract. The **platform-independent system architecture (PISA)** is the fifth layer. This is an independent technology architecture for the future design of a system. In this layer, we emphasize the need to specify a future system architecture and deployment architecture. In the sixth layer, the **platform-specific model (PSM)** defines characteristics for the selected blockchain-related platform, the IoT devices, interfaces, and other related technology components. The last layer (seventh) of our approach defines the **platform-specific smart contract model (PSSCM)**, which presents technology-specific coding artifacts for developing the targeted[1] system. Figure 6.1 illustrates the diagram of our design method. The left side (blue) of the diagram presents the part that supports our method's agility, and this is supported by the middle part, which presents the layers of our design method. Each layer is composed according to the information acquired from the previous layer. The right-hand side of the diagram presents MDA conceptual models and the association of our design method layers with the MDA layers.

Before exploring our design method's layers, we initially present some background information and MDA-related concepts.

6.2.1 Model-Driven Engineering (MDE)

Nowadays, the modeling approach presents an important aspect in the field of designing and developing software systems . The models are not remaining just documentation in the systems' lifecycle, but play an important role as significant software artifacts in the system development lifecycle (Silva, 2015; OMG-MARTE, 2019). Models are used to present the simplified and abstract view of the future system (system in the study). Model-driven engineering (MDE) (Schmidt, 2006) presents a broad approach for integrating models into the software development lifecycle. To use models in an appropriate way in the field of software system development, several **"model-driven"** approaches have been developed, such as *"model-driven development (MDD)"*,*"model-driven testing (MDT)"*, and *"model-driven architecture (MDA)"*. These "model-driven" approaches propose a different way of integrating models into the software system lifecycle (Hutchinson et al., 2011; Khalil and Dingel, 2018). The proposed "model-driven" approaches are instances of the MDE (Silva, 2015). Our study focuses on the usability of the MDA approach presented in the following section.

[1]The term "targeted" stands for the system that is being designed and is intended to be developed in the future.

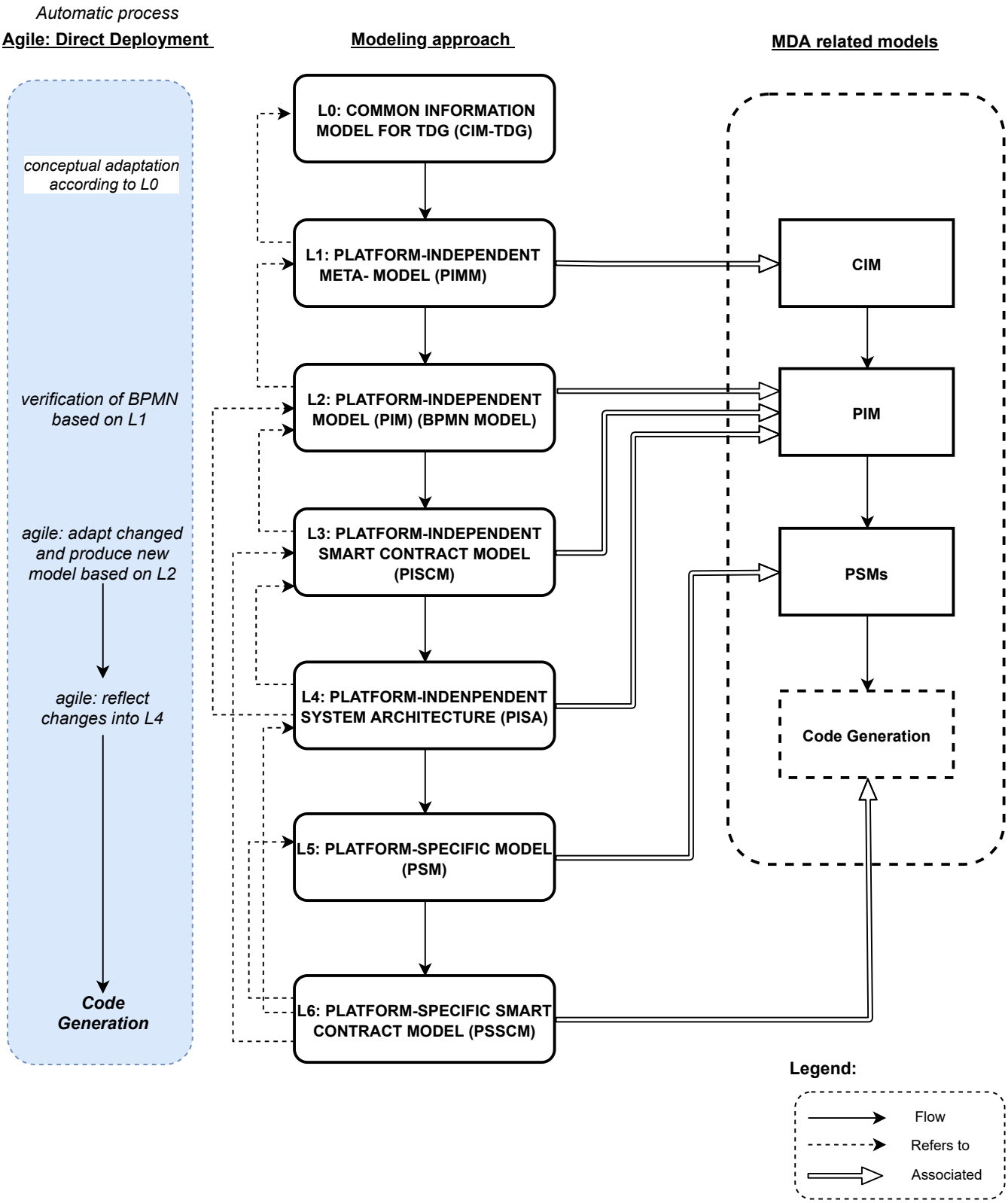

FIGURE 6.1: The Blockchain Based System Model (DG-BCSM) design method referential schema.

6.2.2 Model-Driven Architecture (MDA)

OMG[2] proposes the MDA as an architectural framework that provides an approach for deriving values from models to support the lifecycle development of the targeted system. MDA is a software development approach that is driven by software models (Kleppe et al., 2003; Sharma and Sood, 2011). It introduces several models, i.e., artifacts during the lifecycle of the development of the system. The MDA highlights the power of the models in software development. It uses models to understand, design, develop, deploy, and maintain the targeted system. MDA leverages models for the agile lifecycle and other processes to improve and maintain the produced results, i.e., the final system developed (OMG-MDA, 2014; Brown, 2004). Further, the source, i.e., the MDA models, allows modifications and other model changes. The artifacts created during the MDA lifecycle present a significant difference between the MDA and traditional software life cycles.

6.2.2.1 MDA Models

The MDA's core functionality is its models produced throughout the system development's lifecycle. MDA facilitates the communication and understanding of the system by providing unified models and a modeling language that is understandable by the teams involved in developing the system. This is one of the MDA's ultimate goals for the models and modeling language (OMG-MDA, 2014; OMG, 2014).

MDA enables several viewpoints (or abstraction levels) for the targeted systems. The meta-models defined at the early stage of the development are considered the architectural kernel of the MDA approach (OMG-UML, 2010) and the requirements for the targeted system should be defined at the very beginning of this approach (formulated from the domain practitioners). This step or model is seen as the expression of the business rules or the domain model and is mainly expressed textually (with the domain practitioners' vocabulary) or with low-level modeling schemes, for example, a UML Class Diagram (or Use Case or Activity Diagram). This model is known as the *"Computational Independent Model (CIM*[3])"*, and describes how the targeted system is expected to behave and a variety of tasks to fulfill the requirements. CIM*[4] does not show any information about how the system will be developed or any other technology-related information (OMG, 2014; Truyer, 2006; Brahim et al., 2013). CIM* is the primary source of information shared between the domain expert and software engineer (Rhazali et al., 2016).

For any targeted system context, the fundamental step when developing an MDA-based system is the definition of the *"Platform Independent Model (PIM)"*. Contrary to CIM*, which mainly expresses business requirements, the PIM defines high-level system architecture, which aims to meet the business requirements expressed in CIM* (OMG-MDA, 2014; Truyer, 2006). The PIM is a model that represents a certain level of independence from any technology, which then allows it to be mapped to one or more technological platforms.

[2]Object Management Group. https://www.omg.org/mda/
[3]The CIM* indicated CIM from MDA, thus CIM-MDA.
[4]The CIM* directly indicates what the system is supposed to do according to defined requirements.

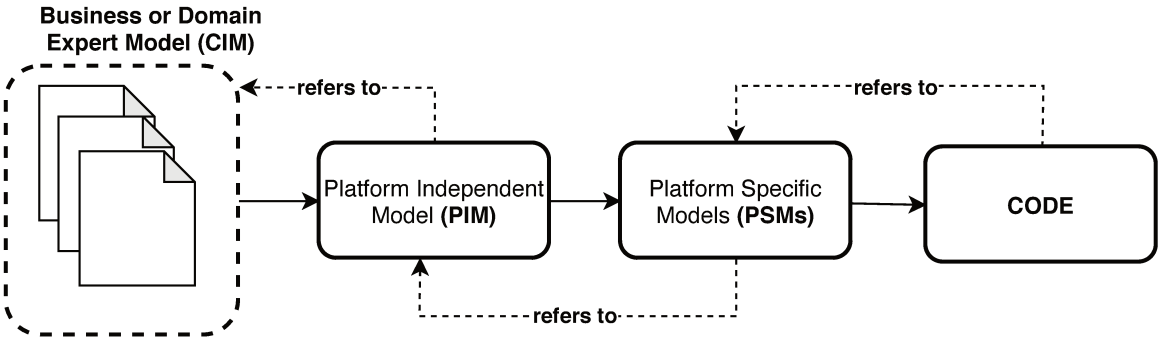

FIGURE 6.2: The MDA approach for the software development lifecycle.

The next step in the MDA approach is transforming PIM to one of the different models related to the technological platform. This model is called the *"Platform-Specific Model (PSM)"*. The PSM aims to specify the implementation components that are available in a specific implementation technology (Kleppe et al., 2003) (e.g., in our study context, blockchain components, and related technologies). PSM is a model that is mainly technology-specific (OMG-MDA, 2014). For a single PIM, different PSMs may be produced that are needed to respond to the technological aspects, since most systems nowadays are composed of several technologies (Kleppe et al., 2003; Vidales et al., 2005). The properties in PIM have several benefits. Firstly, multiple technologies might be derived from the same PIM. Secondly, any change in the business requirements reflected in PIM can then be propagated in PSMs, and, finally, any change in technologies, e.g., evolving or adding new technology components, can then be further reflected in the PSMs (OMG-MDA, 2014; Brown, 2004).

The final step in the MDA-based system is transforming the PSMs into the *code*. This step is considered straightforward due to the information contained in PIMs, related to the technology (Kleppe et al., 2003; OMG-MDA, 2014; Truyer, 2006).

Figure 6.2 show the models of MDA and their association. The PIM emerges from the CIM*, then when the PIM has been defined, we can design further PSMs, and finally, the code is generated from PSMs.

6.2.2.2 The Benefits of MDA

MDA's primary goals, that we intend to achieve by architectural separation are portability, interoperability, and reusability (OMG-MDA, 2014).

6.2.3 MDA and Model Transformation

Due to the rapid development of society, the end-user requirements can be changed at any time. This obliges enterprises and other service providers to change and adapt their business processes to respond to these changing requirements. Without pre-defined model transformation (MT) techniques, transforming models into executable code is considered manual work and is time-consuming. The MDA is foreseen as an efficient mechanism that reduces the time between modeling and transformation toward the executable code

(Argañaraz et al., 2010). MT refers to the production of different models, artifacts, and viewpoints within the same system (Kleppe et al., 2003; Truyer, 2006). The transformation itself is an automatic mechanism that takes a model as an input (source) and generates the targeted model as an output, based on the defined transformation rules (Kleppe et al., 2003; Argañaraz et al., 2010). The transformation definition presents a set of rules that define how the model will be transformed (mapped) from the current model (modeling language), to the targeted model (modeling language) (Kleppe et al., 2003; Argañaraz et al., 2010; Truyer, 2006). The transformation rules present the way a specific building block (from the input model) is transformed (mapped) to another building block (or element, or other characteristics) of the targeted model. The MT from CIM* to PIM or PIM to PSM presents Model-to-Model (M2M) transformation while transforming PSM to Code presents Model-to-Text (M2T) transformation.

The OMG has defined a standard language for the definition of MT, such as *Query-View-Transformation* (QVT) (OMG-QVT, 2016). This standard enables the generic transformation between meta-models, and is based on the Meta-Object Facility (MOF). Although QVT is standard for MT, it is not the only way to perform such actions. Several kinds of research have emerged in line with MT beyond the QVT. The MT has been emerging according to the needs and specific requirements in transformation aspects. Depending on the language used for source and target models, various transformations are present, i.e., *Endogenous transformation* (when meta-model and model are the same) and *Exogenous transformation* (when meta-model and mode are different). The MT is not possible for all models. Some models are not even intended to be automatically transformed (OMG-MDA, 2014). MDA distinguishes two automating paths from the model to the executable system. The first one recommends applying a *transformation pattern* to the model, which then produces a technology-specific model or artifacts. An example of such a transformation is when a pattern is applied to the business information model, and is further transformed into XML Schema (or C# or Java code). The second automation recommended applies the model to a *model execution engine*, which is implemented on a specific platform and directly executes the model. This engine treated the model as interpretable source code (OMG-MDA, 2014).

The MDA model PIM is more related to the concept of business requirements, while the PSMs are related directly to technology. MDA intends to automatically transform the PSMs from the PIM (OMG-MDA, 2014). A specification or model should clearly specify how this transformation will occur, based on the parameters provided, i.e., by defining which transformation platform is selected, e.g., XML Schema (OMG-MDA, 2014). There is an extensive list of research studies that propose MT, e.g., UML to BPMN. (Argañaraz et al., 2010) shows the MT for the UML Activity Diagram into BPMN.

For an efficient MT, support is necessary from modeling tools. Modeling tools play a significant role in MT. The models defined can produce a standard format, e.g., the *ECORE* (Rahmani et al., 2010; Schätz, 2009) model, which is a standard and recognized modeling framework from many other tools. It can be exchanged between several tools, and there are MT that could be applied according to the defined transformation rules.

In the following section, we introduce the first layer of our design approach.

6.3 Common Information Model for the TDG (CIM-TDG)

DG transport and management is vast and diverse, involving different DG classes, including many stakeholders located in different countries. The complexity in TMDG is evident, considering that, for a specific mode of transport, a particular regulatory framework is applied, which implies different authorization procedures, transport conditions, and management processes. Because of the DG's diverse characteristics (e.g., infectious, radioactive, explosives, etc.), different authorities are responsible for specific DGs. Choosing a particular transport context (local and international) or changing DG or transport modes implies different legal frameworks, procedures, and responsible authorities. To facilitate the TMDG, we propose a data model as a source of knowledge that helps manage the organizational aspects of TDG. We define a common information (data) model for the TDG (CIM[5]-TDG), and observe it as necessary to cover different use cases in TDG. Beyond the current use case that we are developing for the TMMW, the CIM-TDG includes the formal expression necessary to cover any possible TDG use case by providing the essential information and help in defining requirements for the specific TDG use case. We develop a formal knowledge representation model for CIM-TDG, which intends to serve as a knowledge source for any TDG-related application. Such a model is designed to serve as a referential data model and be shared among different applications. Furthermore, another scientific component that motivates us to work with CIM-TDG is the current model's capability to be extended to support new TDG that rely on the regulatory framework.

For stakeholders who intend to cooperate or have a means of application (IT service) that facilitates the TMDG, we propose using the common information model (CIM) as a referential information model for the TDG (CIM-TDG). The CIM-TDG provides a standard model for managing information for DG from a stakeholder perspective, a regulatory framework, and process flow aspects by considering conditions (constraints) at any process workflow level. The CIM-TDG provides the fundamental source of knowledge for the TDG process, with the possibility of sharing it among different applications as a common information model (common knowledge base). It intends to facilitate the incorporation of existing and newly developed applications that aim to support the TDG. Considering that the CIM-TDG is developed using UML as a modeling language, this facilitates the development or translation of this model to new applications, including blockchain-based applications.

6.3.1 CIM Overview

The IT environment requires an information exchange between the involved agents. From the perspective of process management and workflow, TDG presents a distributed process, including distributed stakeholders, application, and IoT-based systems. For the system resource (objects) components, for possible cooperation, a standardized (referential) information model that enables specific data format exchange is required. The Common Information

[5]CIM-MDA (CIM*) and CIM-DMTF have significant differences, as we have summarized them in Appendix A.2

Model (CIM) is a data-model standard defined and maintained by the Distributed Management Task Force[6]. It provides a common definition of management information for applications and services, networks, and systems (DMTF, 2012). CIM provides a common vocabulary used to integrate applications, database schemas, data structures used internally by an application, and information exchange structures (RDF or XML)[7]. CIM allows the extension of the current information model (DMTF, 2012). It provides the means to show how the represented elements in an IT environment are presented by means of sets, objects, and relationships.

- CIM Specification (Infrastructure)

  The CIM Specification allows the definition of the architecture and the concepts of CIM. It determines the language for developing the CIM. It proposes a method that allows the usage of the CIM by other information models. The CIM Architecture is based on UML (DMTF, 2012). For developing the CIM-TDG schema, we use UML as a modeling language, and we use RDF/XML as a method for data exchange.

- CIM Schema

  Presents any conceptual schema that defines a specific set of objects and their relationships. It follows an object-oriented approach, and the resources (objects) managed are presented as an object class with attributes and methods (DMTF-MM, 2014; Keller et al., 2001). The CIM schema presents a common base for elements managed in the CIM model, the actual model description, and the building blocks for the application (management platform, device configuration, etc.). The structure of the CIM schema is composed of the core model, common model and extension schemas [8] (DMTF, 2020; Keller et al., 2001; DMTF-MM, 2014). In the context of our study, we present a new CIM schema, called CIM-TDG.

- CIM profile

  The CIM profile presents the set of restrictions on the defined CIM schema. Applications that intend to cooperate should share the same CIM profile. The CIM profile presents a subset of the general CIM. An example of expressing CIM profiles is Web Ontology Language (OWL). We apply CIM profiles to the CIM-TDG schema.

6.3.2 CIM-TDG Schema and CIM-TDG Profile

In this section, we present the general organization of the process of TDG. The study identifies the TDG components and their relationships in the fulfillment of the process of TDG. These components are shown in the CIM-TDG schema, presenting core components for organizing the TDG. A CIM-TDG schema presents the organizational aspects of TDG, including specific

[6] Distributed Management Task Force (DMTF). CIM published standard documents: https://www.dmtf.org/standards/published_documents

[7] Resource Description Framework (RDF); eXtensible Markup Language (XML); RDF Syntax Grammar

[8] CIM Schemas available in HTML or XML as shown in the following link: http://schemas.dmtf.org/wbem/cim-html/2+/

aspects in the TMMW use cases (5.3). The aim of the proposed CIM-TDG is to be a referential information model for the process of TDG.

6.3.3 CIM-TDG General Schema

A wide range of components is involved in the TDG process and must be considered when designing it. To demonstrate all these core model TDG components, we have defined a CIM-TDG schema, as shown in Figure 6.3. We develop it further to show all the components required in TDG. The CIM-TDG schema is designed based on the analysis of the entire process of TDG ((Imeri et al., 2017), 2, 5, 5.3). For the concrete use-case in particular, we also refer to the regulatory framework *Regulation (EC) No 1013/2006*, as shown in Section 5.

In the CIM-TDG schema, "Transport" is among the main components. It is associated with the "Stakeholder" (6.3.3.3) and "Logistics" entities (6.3.3.2). The "Stakeholder" component presents any stakeholders (actor) involved in the TDG while the "Logistics" component presents the components necessary for managing TDG and organizing the transportation process for the DG. The "Regulatory Framework" component shows the legislation that is applicable in each country (represented by the "Country" component) for the process of TDG. This component is the foundation for compliance with legal issues, standards, and policies for operations with DG. The "Regulatory Framework" component is composed of instances of "International Laws", which cover any international regulations, "Local Laws", representing any local laws that must be consulted for the TDG. The "Information Exchange" presents any data exchange policies from a "Stakeholder" perspective and is applicable at the local and international levels. The "Information Exchange" component allows the stakeholder to determine which data can be shared and with whom, and also determines the level of details. "Stakeholders" can define this sort of policy following laws related to data protection. The "Business Activity" (6.3.3.4) component presents TDG activities performed by the "Stakeholder". The TDG activities are performed by stakeholders with their headquarters in a specific "Country".

In the following section we presents several CIM-TDG schemas (6.3.3.4, 6.3.3.3, 6.3.3.1 and 6.3.3.2)), which jointly make up the entire CIM-TDG schema shown in Figure 6.4. These CIM-TDG schemas may be subject to a CIM-TDG extension with a new component with- out disturbing the entire CIM-TDG schema.

6.3.3.1 CIM-TDG Transport

The CIM-TDG Transport schema shown in Figure 6.4 presents the principal components required in the process of transport. It consists of "Transport Context" components, which contextualize transport operation at the local or international level. The "Transport Infrastructure" determines the required physical infrastructure that will be used in transport, e.g., "airway" or "highway," etc. Usually, in the transportation process (particularly for TDG), related documentation is required. Transport vehicles and transport operations (public or private transport) are determined by the "Transport Elements" component. The "Transport Time" component determines the transportation time by measuring transport time in *years*,

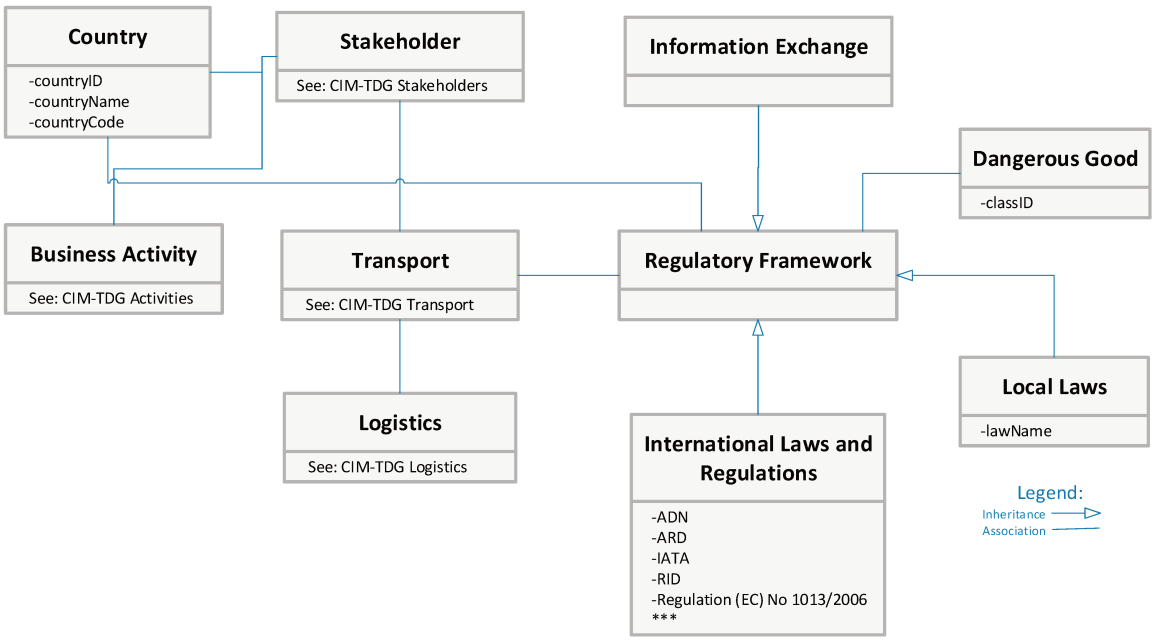

FIGURE 6.3: The CIM-TDG general schema for the process of TDG. It presents a global referential CIM model for TDG.

months, days, hours, minutes, seconds. The component "Transport Mode" describes the mode of transport. It includes "Road" transport, which determines roads as a transport mode. The other modes of transport include "Air Transport", "Railways", "InLand Waterways" and "Sea Transport" as the possible transport modes for the DG. The "Transport" component is associated with the "Logistics" component, thus organizing the TDG.

6.3.3.2 CIM-TDG Logistics

The "Logistics" component has its own activities, and is composed of the following sub-components: "Warehouse", "Terminal", "Loading Procedures", "DG Physical Measurements", "Routing" and "Transport", as shown in Figure 6.5. The "Warehouse" component represents any certified physical object and has the physical capacity to store and maintain DG. The "Terminal" component represents any transport terminal point where a capacity of DG will be received, exchanged or delivered for transport. There are different terminals based on the transport modes. The "Routing" component determines the physical paths (based on the transport mode and infrastructure) which the transport process will follow. The "UnLoading Procedures" component determines the procedures of loading DG based on DG class. It intends to provide clear guidance on preconditions and postconditions in the loading of DG. These conditions are specific for any "Loading Type", e.g., bulk loading, container, or other loading types.

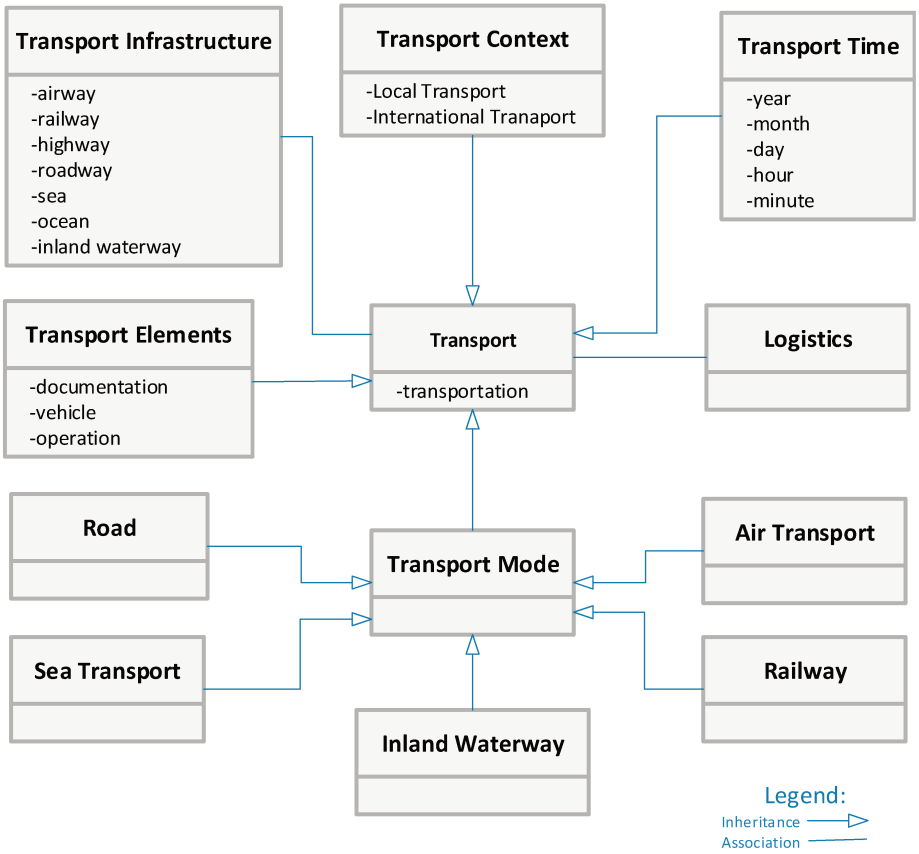

FIGURE 6.4: The CIM-TDG Schema for "Transport" and its related components.

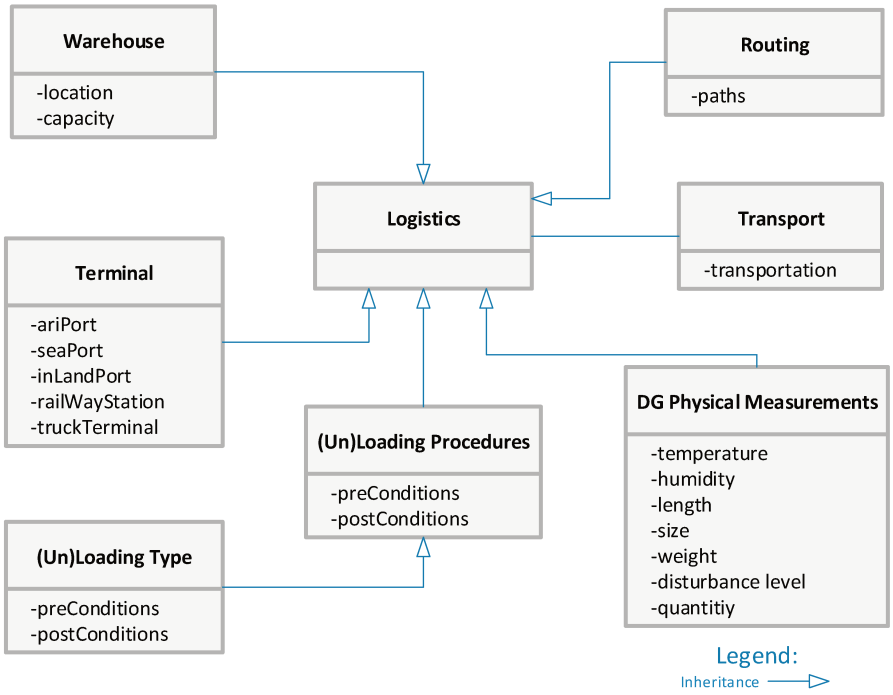

FIGURE 6.5: The CIM-TDG schema for the "Logistics" and its related components.

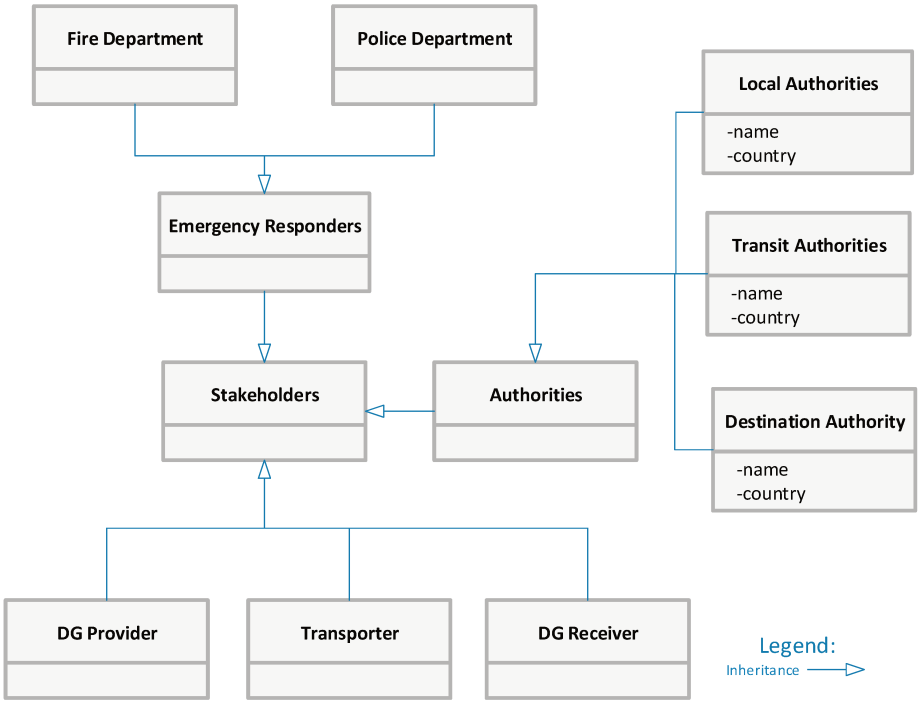

FIGURE 6.6: The CIM-TDG schema for the "Stakeholders" involved in the process of TDG. This schema may be extended with other involved stakeholders depending on DG specificity and regulatory framework.

6.3.3.3 CIM-TDG Stakeholders

"Stakeholders" are responsible for the organization, monitoring, and completion of the TDG process. In the TDG, the main TDG stakeholders are: "DG Provider", "DG Receiver", "DG Transporter", "Authorities" and "Emergency Responders", as shown in Figure 6.6. The "Stakeholder" may have its headquarters in one specific country and operational offices in other countries. The "Stakeholder" component is associated with the "Country" component (as shown in 6.3). TDG is a regulatory-based process governed by the competent authorities. The "Authorities" component represents any relevant-competent authority responsible for TDG. Based on the TDG context (local or international), the competent authorities involved are the "Local Authorities" responsible for authorizing and monitoring TDG in the local and international context. The "Transit Authorities" are involved in an international TDG and are responsible for authorizing and monitoring TDG. The "Destination Authorities" are responsible for authorizing TDG, monitoring the DG treatment aspects for the received DG, and providing the necessary information for the involved stakeholder in an end-to-end international TDG process. The "Emergency Responders" component represents any stakeholder responsible for reacting in the event of an accident with DG. This component is mainly composed of first responders from the Fire and Police Departments, but the list of such stakeholders might be considerably longer and specific to the DG that are subject to transportation.

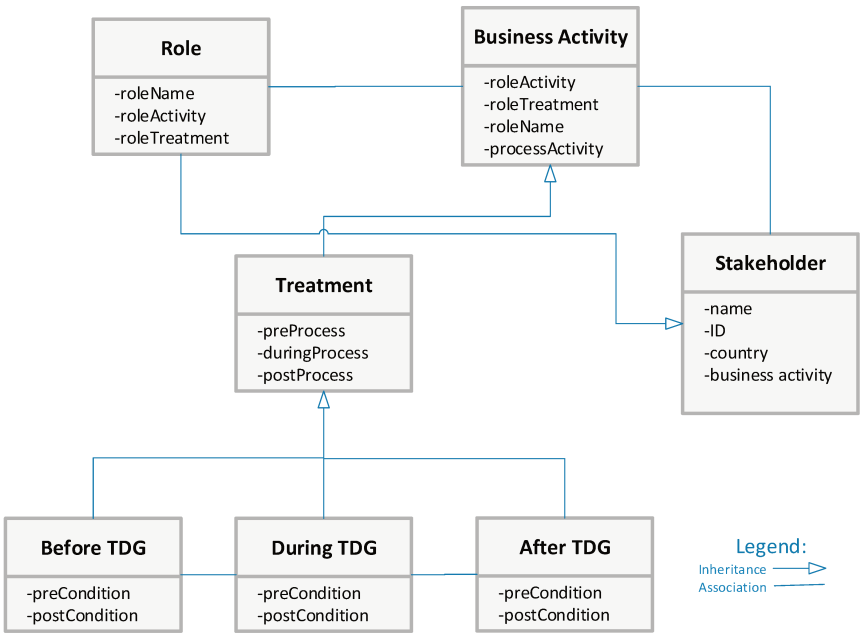

FIGURE 6.7: The CIM-TDG schema for the business activities performed by "Stakeholders" in relation with TDG.

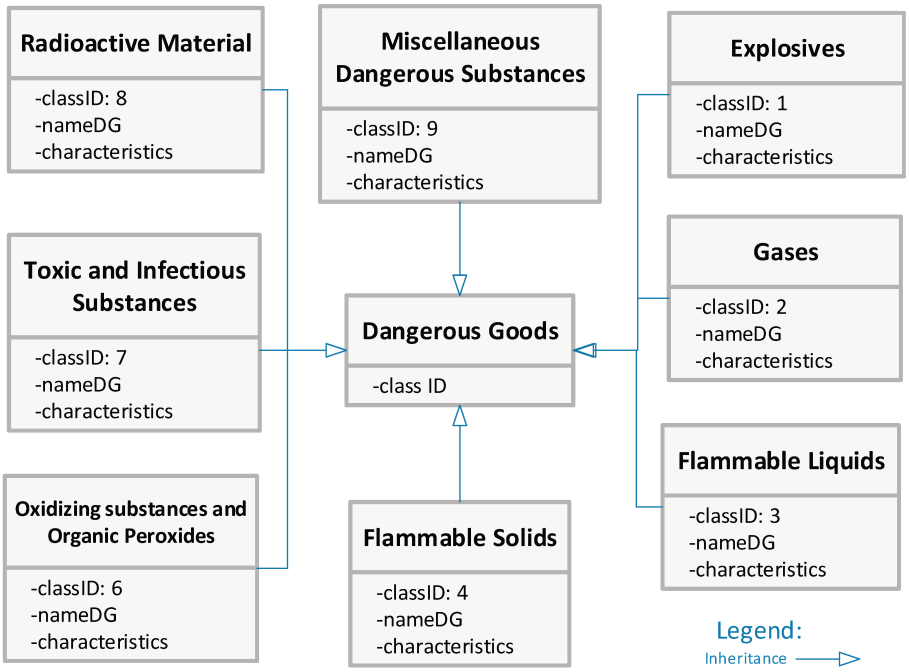

FIGURE 6.8: The CIM-TDG schema for the "Dangerous Goods" component.

6.3.3.4 CIM-TDG Activities

Figure 6.7 shows the CIM-TDG Scheme of Activities. The stakeholders perform the "Business Activity" in the TDG. All "Business Activity" should be coordinated and authorized by "Authorities". "Business Activity" is composed of "Treatment". The "Treatment" represents any process (activity) that is performed in the TDG. It is composed of the components: "Before TDG", "During TDG", and "After TDG". The "Before TDG" includes any activity before the transportation starts. It presents precise guidance on which process should be followed to continue to the next process. "During TDG" covers all activities and events (data events) that are performed during the transportation process, and "After TDG" covers any activity after the DG delivery. In the "Stakeholder" activities, different roles may perform specific operations, e.g., *examining the vehicles that will transport DG, or collect and share TDG related information*. We consider the "Role" component to represent these particular roles, including "Safety Adviser". All "Roles" are associated with "Business Activities".

6.3.3.5 CIM-TDG Dangerous Goods

The "Dangerous Goods", component as shown in Figure 6.8, presents the general schema for introducing any DG. It is composed of all DG classes (presented in Chapter 2). Besides showing all DG classes and their characteristics, this schema serves as the main source for future extensions and for presenting a DG instance according to its class.

6.3.4 CIM Profile for CIM-TDG Schema

The previous section showed a general CIM-TDG schema composed of unary and binary relations. The CIM-TDG schema defined in UML does not provide any restriction (in terms of reasoning or cardinality) since it presents a semi-formal relation between the components. That means that the specific component may be ambiguous in the CIM-TDG schema, decreasing the CIM-TDG schema's ability to be shared as a common model between other applications or providing reasoning over the current model. We propose a formal representation of the CIM-TDG schema to avoid such issues. Using CIM Profiles, we propose applying restrictions to the proposed CIM-TDG schema, as is the main suggestion of the scientific literature. The purpose of the CIM Profiles is to apply a set of constraints on the CIM-TDG schema relations. To establish the CIM Profile, we use Ontology Web Language (OWL)[9]. OWL is an ontology language standard recommended by W3C[10], which uses the strength of description logic (A.8.1) (Horrocks et al., 2007) and practical reasoning for formal knowledge representation (Majewska et al., 2007).

To define the CIM-TDG Profile, we follow two steps:

1) Mapping the CIM-TDG schema into OWL.

For the mapping of the CIM-TDG schema into OWL, we follow the existing approach on MT (modeling language (ML) transformation ML_1 ->ML_2). We follow the one-to-one

[9]The Ontology Web Language (OWL) is presented broadly in Appendix A.6.
[10]https://www.w3.org/TR/owl2-manchester-syntax/

corresponding component semantics of UML and OWL. Mapping OWL from UML has been the subject of several studies, as shown in (Baclawski et al., 2001; Textor et al., 2010; Cranefield, 2006; Quirolgico et al., 2004; Majewska et al., 2007; Vo and Hoang, 2020). The objective of mapping UML diagrams representing the CIM-TDG schemas is to provide semantically consistent and valid CIM-TDG Ontology as a representative of the CIM-TDG Profile. In order to facilitate interoperability and reasoning over the CIM-TDG schema, we model the CIM-TDG as a formal ontology. A formal ontology represents an ontology in which the semantics of its vocabulary are based on axioms (Quirolgico et al., 2004). The semantics of the CIM-TDG schema, modeled in formal ontology, can be used in the other TDG related applications without having prior knowledge of the CIM-TDG schema. This enables us to provide a certain level of interoperability between TDG-related applications.

Figure 6.9 shows the simplified model of the main OWL classes[11]. These classes are representative of the CIM-TDG schema. We remind the TMDG to presents the general concept for the transportation and management of the DG. The highlighted concepts (red eclipse) in the CIM-TDG conceptual map present the corresponding CIM-TDG schema and related concepts.

2) Define the CIM-TDG Schema constraints by using OWL axioms.

The mapping of CIM-TDG schema into OWL, i.e., CIM-TDG Profile, allows us to express the properties of CIM-TDG schema formally. CIM-TDG Profile applies restrictions (constraints), based on *first order logic and reasoning*, to the CIM-TDG properties. To apply such restrictions, we use OWL axioms. Axiomatization is at the core of our CIM-TDG Profile and determines its behavior by formally specifying constraints in the CIM-TDG schema. Besides applying restrictions, we enhance the CIM-TDG Profile by applying semantics to the OWL model concepts.

Axioms

Axioms are used for imposing restrictions on the CIM-TDG schema. We use axioms to provide additional semantics for the TDG data model, which may be shared among different applications. We consider this model a source of knowledge that different software agents may easily use as a common data model (knowledge base). The defined axioms directly impose certain constraints, which avoids ambiguity and enables a common agreement on concepts and relationships. Following this, we present the main axioms that directly impact the process of TDG. Figure 6.10 presents the main axioms used in the TDG, and also in our specific use case for TMMW (5.3.2).

This axiom (*Axiom I* (6.10)) determines (restricts) that for any DG, there is a need for a treatment process (recovery or disposal), and that the DG belongs to a specific DG class, and has a quantity, labeling and marking of DG. This axiom determines that for any DG treatment, the specified constraints are the minimum constraints necessary. This is mainly true for the TMMW use case and, in general, for the TDG.

The *class: DG Management* presents the management of DG, including the conceptual treatment of DG. For the DG presented by *class: DG*, which is the subject of the treatment,

[11] An extended model of CIM-TDG Profile is shown in Appendix A.6.1

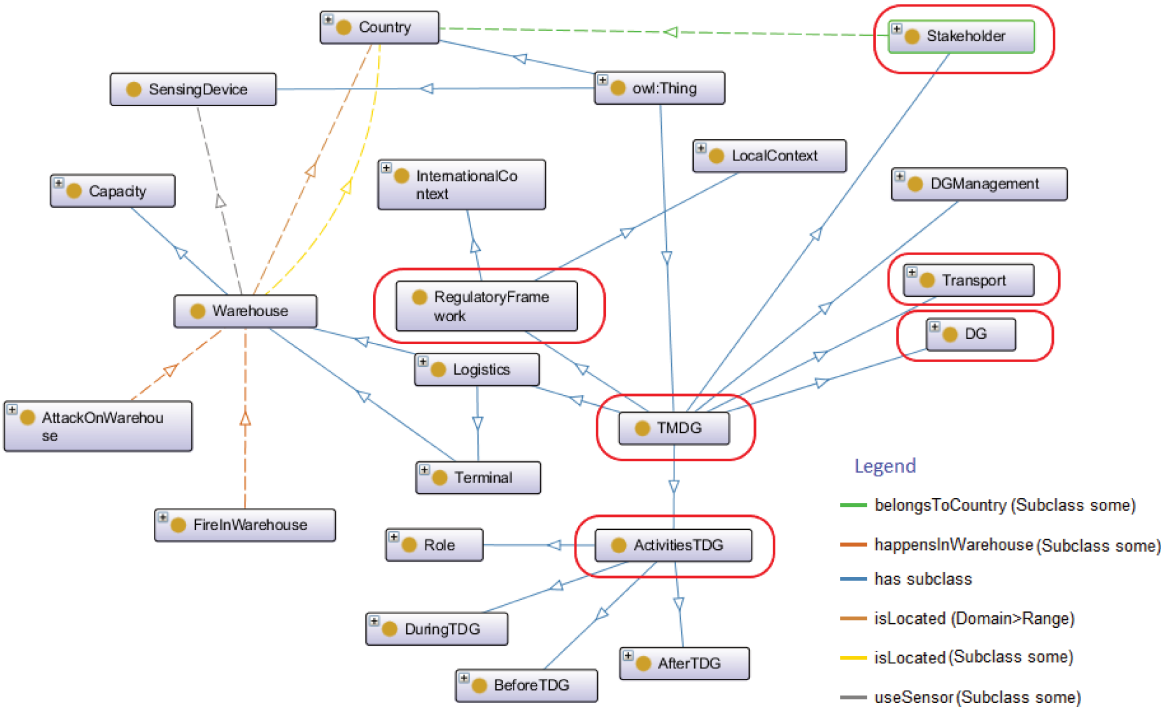

FIGURE 6.9: The main classes in the OWL model representing the ontology of TDG.

the axiom (*Axiom II*) determines the condition of the DG treatment *class: DGTreatment*. The treatment can be local or international, and should follow processes based on the local and (or international) regulatory framework. This axiom also defines that for any DG treatment, authorization (permission) from the competent *class: Authorities* is required. In this way, this axiom supports the condition (5.3.6) specified for the use case of TMMW.

For any activity with DG, an authorization is required. The combination of axioms shown in (*Axiom III*), determines that the *class: DG Provider* (or *class: Transporter*) requires authorization represented by *class: Notification_Authorization*, from the *class: LocalCompetentAuthority*. After performing an evaluation, the *class: LocalCompetentAuthority* transmits the dossier to the *class: TransitCompetentAuthority* and *class: DestinationAuthority*.

The axiom shown in (*Axiom IV*) describes the transportation[12] process. It involves several other concepts such as *class: GeoLocation*, then *class: CrossBorderPoint*, which determines the physical cross-border (and localization) address. During the transportation process, a *class: Terminal* may be used to deliver the DG to its temporary or permanent location. In this axiom, we determine a transport use specific infrastructure *class: Transport_Infrastructure* and also a specific transport mode *class: Transport_Mode*. The movement of DG is performed by using specific vehicles (*class: Vehicle*) and via given paths *class: Paths*.

The axioms shown in (*Axiom V*), intend to determine: *a)* The services of *class: EmergencyResponders* are used only in the event of the *class: DG_Events*, which conceptualizes the

[12]The transportation concept presents the process of movement of the DG

((needTreatment some Disposal) or (needTreatment some Recover))
and (belongToClass some DGClasses)
and (hasQuantity some DG_Quantity)
and (isLabeled some Labeling)
and (isMarked some Marking)

Axiom I

DGTreatment and
((dgTreatmentContext some InternatonalTreatment) or
(dgTreatmentContext some LocalContext))
and ((requiresRegulatoryFramework some InternationalContext)
or (requiresRegulatoryFramework some RegulatoryFramework)
or (requiresRegulatoryFramework only InternationalContext)
or (requiresRegulatoryFramework only LocalContext))
and (requirePermission some Authorities)
and (requiresDGTreatment some DGTreatment)

Axiom II

(requireAuthorisation some Notification_Authorization)
and (sendAuthorisationReq some LocalCompetentAuthority)
Stakeholder
LocalCompetentAuthority
and ((dossierTransmitTo some DestinationAuthority) or
(dossierTransmitTo some TransitCompetentAuthority)) or
(dossierTransmitTo only DestinationAuthority)
Authorities
TransitCompetentAuthority
and (InternationalContextTDGAuthorities and
(notifyOtherAuthority some DestinationAuthority) and
(authorisationDecision only LocalCompetentAuthority)) or
(requireAdditionalDocs only LocalCompetentAuthority)
InternationalContextTDGAuthorities
DestinationAuthority and
((notifyOtherAuthority some TransitCompetentAuthority) and
(authorisationDecision only LocalCompetentAuthority)) or
(requireAdditionalDocs only LocalCompetentAuthority)
InternationalContextTDGAuthorities

Axiom III

DuringTDG
isPerformedBy only
(DG_Provider or Transporter)
Transportation and ((hasDestinationPoint some GeoLocation) and
(hasStartPoint some GeoLocation)) and ((movesThrought some Paths)
or corssBorderPoint some CrossBorderPoint
and (useTerminal some Terminal)
and (useTransportInfrastrucure some Transport_Infrastructure) and
(useTransportMode some Transport_Mode)
and (useVehicle some Vehicle)) and (hasArrivalTime some
(Days and Hour and Minutes and Seconds and Year))
and (hasStartingTime some (Days and Hour and Minutes and Year)) and
(requiresDocuments some Transport_Documentation)

Axiom IV

a) EmergencyResponders and areUsedInCase only DG_Eve
b) Stakeholder and belongsToCountry some Country
c) DG_Treatment and isPossibleForTreatment only DG_Delivery
d) Certificate and isComposedAfter only (Disposal or Recover)

Axiom V

FIGURE 6.10: The main axioms used in the TDG and particularly for the use case of TMMW.

emergency situation with DG, e.g., accidents, fires in warehouses or any other accident. *b) class: Stakeholders* belong to *class: Country*. *c)* The dangerous treatment *class: DG_Treatment* is possible only after the DG has been delivered *class: DG_Delivery*. *d)* After the treatment of the DG a certificate should be issued *class: Certificate*.

To enhance the process of TDG, we use axioms to define TDG missions. A mission intends to gather the necessary information for an end-to-end process in the TDG. In terms of information, the TDG-mission identifies the stakeholders (DG Providers, Transporter, and DG Receiver), authorities (local, transit and destination), DG characteristics, and identifies paths (movement possibilities) used for the TDG. The information from the TDG mission provides essential information for starting any TDG process. The axioms shown in Figure 6.11 determine a mission for TDG from Luxembourg to Portugal. The result of these axioms gives the exact procedure for TDG, and is a critical point that enables substantial TDG processes to be shown. Starting from this result, we might contact stakeholders to perform such a process. The axiom in the TDG mission for the use case of TDG from, e.g., "Luxembourg" to "Portugal" activates other related axioms. The axioms activation, as shown in Figure 6.11, in this process, indicated the TDG process flow based on the regulatory framework.

6.3.4.1 CIM-TDG Schema Extension

a) Extension of the current CIM-TDG with new object (use case).

Another powerful aspect of the CIM-TDG is the possibility of extending it with a new component and adapting it to specific needs or use cases. We extend the current CIM-TDG schema by adding new components.

We use the CIM-TDG schema inheritance to extend it for a specific purpose. For example, let us present some additional components in the context of the CIM-TDG schema for "Transport". In the TDG, a diverse range of DG is present, and plenty of "normal" goods may become DG if specific conditions during transport or storage are not respected. Two examples of such goods are "blood" or "vaccines" with potential infectious threats if not properly managed. For such goods, only a special transport vehicle should be used. Thus, we present a "Special Vehicles" component, with the attributes "bloodTransportVehicle" and "vaccineTransportVehicle", which determines the purpose of these components, as shown in Figure 6.12.

In this sense, we perform the mapping of the "Special Vehicle" into OWL as classes. Figure 6.13 shows these components into ontology (OWL) as subclass of *class: Vehicle*. The *class: SpecialVehicles* has subclasses *class: Blood_Trasport_Vehicle* and *class: Vaccine_Transport_Vehicle* which define the characteristic of the vehicles for blood and vaccine transport respectively.

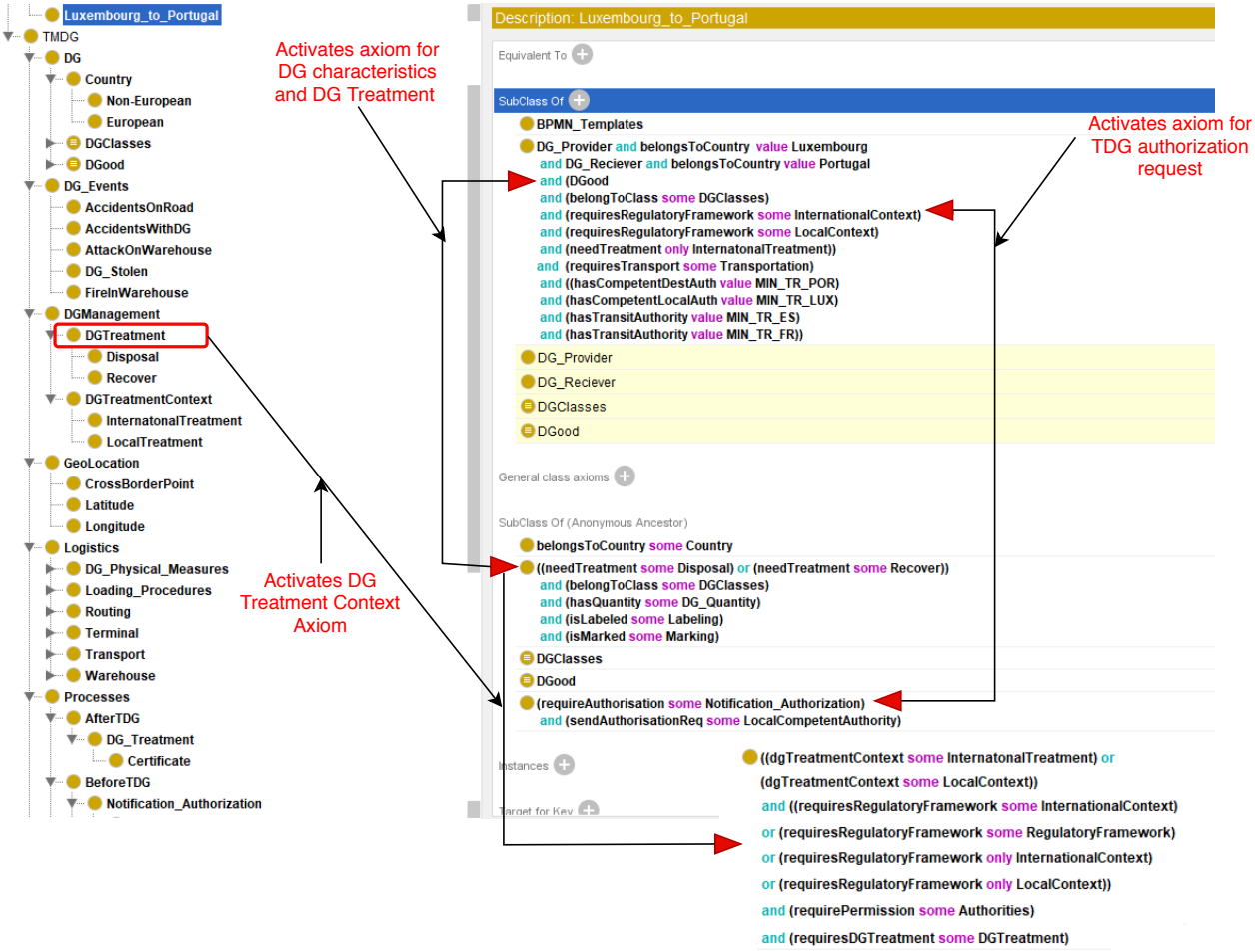

FIGURE 6.11: The TDG-mission expressed with the help of axioms.

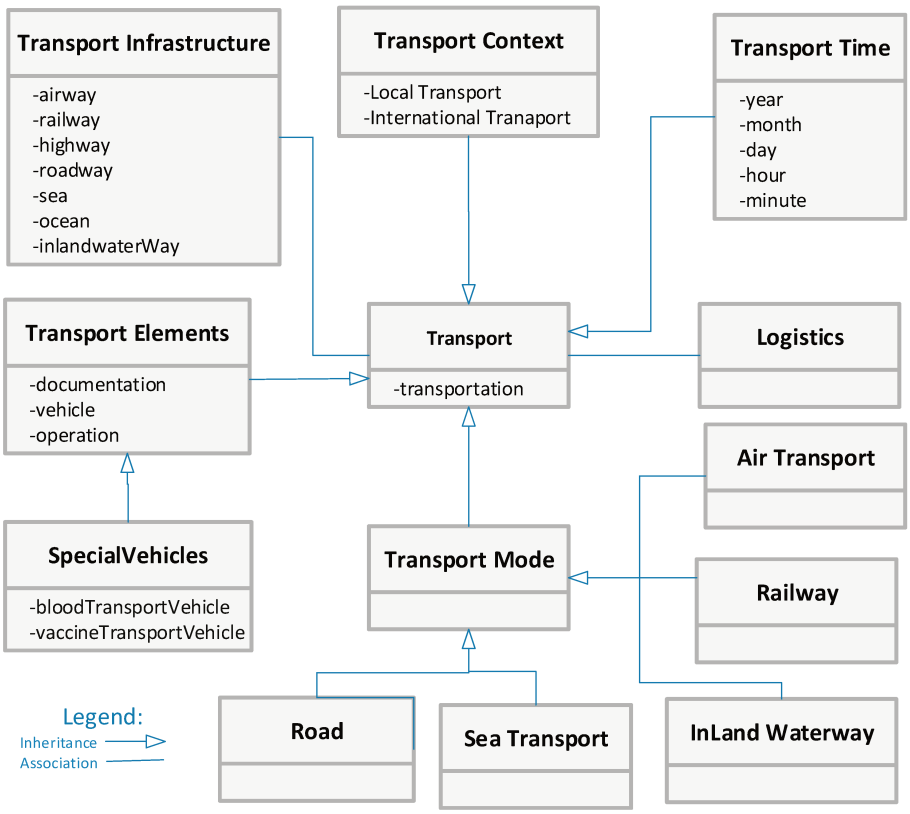

FIGURE 6.12: The extended CIM-TDG schema for the "Transport" and its related components.

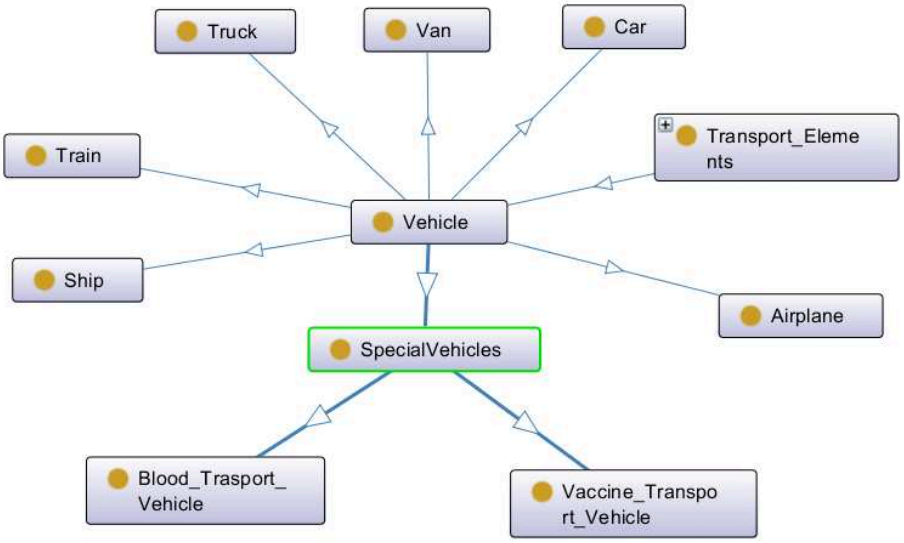

FIGURE 6.13: The changes in the CIM-TDG schema for "Transport" reflected in the OWL model.

All the axioms mentioned earlier are valid and applicable in international and local transport for these newly defined components.

This extension shows the CIM-TDG schema's ability to adapt the current model for a specific use case. "Blood" or "Vaccine" transport may be a use case in the TDG. Although at first sight these goods might not be considered DG, their possible contamination, thus making them DG, poses an extreme threat for human beings.

b) Extension of the current CIM-TDG and exploitation of the other existing Ontology.

Beyond the current semantics offered by the CIM-TDG, it allows the exploitation of the existing well-defined ontology that is already accepted for standardized use. In this context, we attribute using other ontologies to enhance the CIM-TDG and avoid the overlapping work on defining similar ontologies. At the use case level, we use additional concepts from the specific case in the CIM-TDG. For example, we mentioned above that *many "normal" goods might become DG if specific conditions are not fulfilled*. These conditions may imply the environmental parameters that potentially change the state of DG. Hence, to maintain such parameters under surveillance, we propose other necessary concepts in the CIM-TDG. The "SensingDevice" concept is presented as IoT devices that allow the capturing of specific data in the process of TDG. This concept belongs to the *Semantic Sensor Network Ontology*[13], and presents the semantics of a sensor network. It is used in the CIM-TDG to support several activities such as transport and warehousing. Similarly, we can use other existing ontologies for "Geo Location" knowledge, "Cross-Border or Countries," and many others.

[13] http://purl.oclc.org/NET/ssnx/ssn

6.4 Conclusion

This chapter showed our design method by exploring its primary purpose and presenting a general description of it. It presented our design method and the corresponding layers and their semantics. The first (of seven) layer of our design method was presented and fully specified in this chapter. This layer presents a common information model for TDG (CIM-TDG).

The CIM-TDG model was designed based on the identified requirements of the TDG process. It captures the general common information required to organize, maintain, and complete the TDG management's lifecycle. CIM-TDG shows a general common information model for the TDG by providing the required and necessary information to organize, maintain, and complete the TDG cycle. We considered choosing CIM to present concepts in TDG and proposing a model that optimizes the organization of TDG and enables the management of different DG-related use cases. We use the current CIM-TDG and its related components to support a particular use case in the TDG. The design and development of the CIM-TDG are based on the use case for TMMW that we are developing, but we have further generalized it to support the TDG more broadly, thus enabling almost any TDG-related use case.

Another motivation behind the definition of the CIM-TDG was the composition of a common and structured data model that will possibly be used for different applications in the framework (context) of transport and DG management. With the proposed CIM-TDG schema, we provide a source of knowledge for the process of TDG. This collects a certain level of information as a base for any workflow applications in the process of TDG. The CIM-TDG Profile maintains the aspects of the regulatory framework for any application on the TDG.

The set of required information in the context of TDG is available for query (by using Description Logic Query) on the CIM-TDG Profile. Furthermore, the current model may be used as a source of knowledge for the process workflow in TDG by using it in the external software libraries[14].

Defining the CIM-TDG and performing the initial analysis over the TDG in general and also in the context of our case enables further in-depth analysis of business rules (regulatory framework) and relevant process workflows for our use case (TMMW). The following chapter will present two other layers of the proposed design method that are related to business analysis and the discovering of requirements related to TMMW.

[14]For such purposes, we have used JenaAppache: https://jena.apache.org/documentation/ontology/.

Chapter 7

Platform-Independent Meta-Model, Verification Aspects and User Model

7.1 Introduction

This chapter outlines six layers (L1 - L6) of our design method (6.1). The L1 layer defines the platform-independent meta-model for the TDG, representing the static aspects of the CIM*. It maps the business rules into a meta-model for the TDG. The L2 layer presents the dynamic aspect of the platform-independent model (PIM). We define the BPMN models in L2 in order to determine the process requirements for the TMMW use case. The specific aspects of the defined BPMN model are validated according to the mapped business rules in L1.

The business rules are a set of conditions (policy) that must be satisfied (Martin and Odell, 1994; Morgan, 2002). The main purpose of the business rules[1] is to determine how an organization will perform operations (Bajec and Krisper, 2005). The business rules determine the process flow by defining the "know[-how]" from which the "flow" is determined (Halle, 2001; Brahim et al., 2013; Morgan, 2002). Business rules determine how the business is organized and executed by setting out the constraints and other related conditions that must be respected in the different stages (activity levels) of the process (Martin and Odell, 1994; Brahim et al., 2013).

In the context of TDG, the *business rules* determine any activity with DG, not just at the application level, but also at any stage of the process. The business rules in the TDG are based on the regulatory framework. The law articles determine the specific conditions that must be met while performing TDG-related activities. The stakeholder involved must rely on and respect the regulatory framework entirely to be able to operate with DG in a regulated (compliant) way. The use case selected (5.3) belongs to the regulated domain, where the processes are governed according to the specific regulatory framework (5.3.3).

To compose the (L3) layer, the previous layer (L2) is transformed from BPMN into a UML Sequential Diagram (USD). This transformation allows us to define the inner interaction schemas between the targeted system and its components. These schemas are formed based on the activities required in the use case for TMMW. The interaction or exchange of information between the system and its components is performed based on the function

[1] Business Rules Group https://www.businessrulesgroup.org/theBRG.htm; OMG Business Motivation Model https://www.omg.org/spec/BMM/1.3/PDF

as a means for information (input/outputs) treatment and for imposing specific conditions on the process flow. We call these schemas Function Exchanged Diagrams (FED), and the corresponding algorithm expressions are extracted from these schemas. The FED allows the exchange of any information on the system, and in order to specify the right to access and manipulation information, we present a general access control policy.

Further, we define a platform-independent system architecture (L4), which collects and digitally presents each targeted system component functionality and its interaction with systems. Subsequently, the deployment architecture is refined to support the future deployment of the functional system architecture.

Layer (L5) is entirely technology-related. It determines the platform elements, including the technology platform and its main characteristics, IoT devices (or IoT platform), the specification of technological components (sensors and other devices), end-user interface, and the access control policy definition. Finally, in (L6), we present a code generation model based on the technical specifications included in (L5).

7.2 L1: Platform-Independent Meta-Model (PIMM)

The selection of a regulatory framework (when designing the use case and performing the initial analysis while defining L0) allows us to perform an analysis that captures concepts, abstraction, and relationships. Based on this analysis, we define a static domain-specific meta-model to express the business rules. The definition of a domain-specific meta-model allows us to be more precise in defining the concept and relationships related to the application domain, e.g., the use case of TMMW. Furthermore, it means we avoid designing "general-purpose" concepts. This layer aims to respond to the question of *"How to establish a domain-specific meta-model that allows the definition and automatic validation of the user model, i.e., BPMN, based on the regulatory framework?*. The following sub-questions are raised to provide clear answers to the previous question, thus determining *"How to define an abstract syntax or meta-model?*; *How to define a concrete syntax that allows the expression or presentation of meta-model concepts into a model, e.g., with BPMN elements?*.

- 1) Defining an abstract syntax for a domain-specific meta-model for TDG

  The analysis performed on the regulatory framework is immensely important for designing a system that relies on it. This analysis serves to define the abstract syntax (or meta-model). Primarily, the definition of the meta-model (or abstract syntax) begins with the identification of the main concepts, abstraction, and relationships (Silva, 2015). This is the initial step, and the system architect should know how to perform it or cooperate with competent experts from the domain. In general, the usual techniques for knowledge extraction and conceptual formalization for defining the abstract syntax are "grammars" for natural language, or more particularly, "meta meta-modeling" e.g., Meta-Object Facility (MOF), as recommended by OMG (OMG-MOF, 2019).

  In the context of our study, we perform knowledge extraction from the regulatory framework, which serves to identify concepts, business rules, and relationships between

stakeholders involved in the TMMW. This step is mainly performed once and manually, and it assists the development of the entire system. The enormous impact of the meta-model comes from the fact that it defines *concepts* and *relationships* between the concepts and also maps the *business rules*. Furthermore, it captures the mechanisms in terms of operations by which the system will be organized in the future. The meta-model defined in this layer is considered particularly for the use case of TMMW and intends to cover all aspects of the regulatory framework. Compared to the previous meta-model defined in the previous layer (6.3.3), which presents a general TDG-context meta-model, the domain-specific meta-model presented is specific for the use case. The specificity means that we consider specific dangerous goods, e.g. *Medical and Infectious Waste*, a transport mode e.g. *Road Transport* and a transport context e.g. *local or international*, as presented in Figure 6.3. As a result of this particularity, it yields a particular regulatory framework that is the meta-model's backbone. In this sense, for other TDG cases, for example, "Radioactive Waste", the same techniques for selecting the relevant regulatory framework might be applied.

Beyond the identified concepts "Authorities", "Stakeholders" and "Transport Modes" that we have introduced previously (6.3.3), in this layer primarily, we identify specific relationships, conditions, and concepts involvement according to the law articles (business rules). Table 7.1 shows the fundamental component we have analyzed to compose a domain-specific meta-model. Following this, the "Concepts" column presents the main concepts used to comprise the domain-specific meta-model. These concepts emerge from the regulatory framework sources shown in the "Legal Source" column. In particular, the "Involved Concept" column shows the involvement of the concepts (stakeholder) according to the "Law article (s)". The "Relationship Purpose" column presents the reference for the relationship between the concepts. The "Action Performed" column indicates the necessary action from the presented concept.

The domain-specific meta-model is composed of a simple UML class diagram (A.5.1), including class attributes and their relationships (associations or inheritances). Figure 7.1 shows the meta-model[2] that is based on the components presented in Table 7.1. In addition, we present further components that are used to improve the process of the TMMW, for example, an external component such as an IoT device.

[2]This meta-model is related to the transport process, and we can extend it (or develop newer versions) for authorization and other related processes in the TDG.

TABLE 7.1: The summary of concepts, relationships and business rules based on the regulatory framework.

No.	Concept	Legal Source	Law Article (s) (Business Rules)	Involved Concept	Relationship Purpose	Action Performed
1	Dangerous Goods (DG)	ADR; *Regulation (EC) No 1013/2006 (EC)*	ADR 1.4.3.1 EC (1)	4; 5; 6; 7; 9;	General Information	Identification, classification, packing, labeling and transport instructions
2	Competent Local Authorities	ADR; *Regulation (EC) No 1013/200 (EC);* Basel Convention (BS); *Law 21 March 2012 (LW);* Directive 2008/98/EC (D)	EC (2) ADR 1.5.9.1 EC(5; 11; 12) EC (4 (3); 7)	4; 5; 6;	Information exchange	Authorization Process Monitor Movement Process Certificate of treatment of DG
3	Competent International (transit and destination) Authorities	ADR; *Basel Convention (BS);* Regulation (EC) No 1013/200 (EC); *Directive 2008/98/EC (D)*	EC (2); ADR 1.5.9.1; EC (8; 9);	4; 5; 6; 7; 8;	Information exchange	Authorization process Monitor the Movement (transport) process Certificate of treatment of DG
4	DG Provider	ADR; *Regulation (EC) No 1013/2006 (EC)*	EC (2; 3; 4; 5)	2; 3; 5; 6; 11; 12	Provide informatio Information exchange DG handover	Requests Authorisation Handover DG for transportation Transport DG
5	DG Receiver	ADR; *Regulation (EC) No 1013/2006 (EC)*	ADR (1.4.1); EC (2; 5; 9; 22-25;)	2; 3; 4; 6; 7; 9;	Receive DG Information Exchange	Receive DG, inform stakeholders for DG arrival. Perform DG traitment
6	Transporter	ADR; *Regulation (EC) No 1013/2006 (EC)*	EC (2; 16)	2; 4; 5; 7;	Receive DG for transport Information Exchange Deliver DG	Preform the process of transport of DG and provide information during this process. Informs for DG delivery.

7	Warehouse	ADR	//	4; 6; 2; 3	Temporary Receive DG Share information	Host temporary the DG. Maintain DG in pre-defined condition.
8	Cross-border	Basel Convention (BS)	BS (2; 6; 9;) EC (30)	2; 3; 6;	Information exchange	Identify cross-border arrival and fulfilling the necessary procedures for crossing border with DG.
9	DG Treatment	Regulation (EC) No 1013/2006 (EC)	EC (2; 9; 9(7); 15 (d);15 (e); 15 (f))	4; 2; 3;	Information exchange	Provide information for DG Traitment.
10	DG Return	Regulation (EC) No 1013/2006 (EC)	EC (22-25);	2; 3; 4; 5;	Return DG Information exchange	In case the received DG are not as mentioned in business contract.
11	DG Documents	Regulation (EC) No 1013/2006 (EC); *ADR;* Basel Convention (BS)	EC (10(5);16(c); 20) ADR (1.4.2.2) BS (4(7))	2; 3; 4; 5	Create, maintain and archive documents	All the process in TDG should be documented.
12	Transport Process	Regulation (EC) No 1013/2006 (EC); *ADR;* Basel Convention (BS);	EC (10; 16)	2; 3; 4; 5;	Collect information Information exchange	It covers all the transport activity, from point of depart until the point of destination.
13	Emergency Responses	ADR	ADR 1.4.2	2; 3; 4; 5; 6; 7; 9; 12	Receive information	In case of any accidents with DG at any stage of the process, they intervene to avoid consequences.

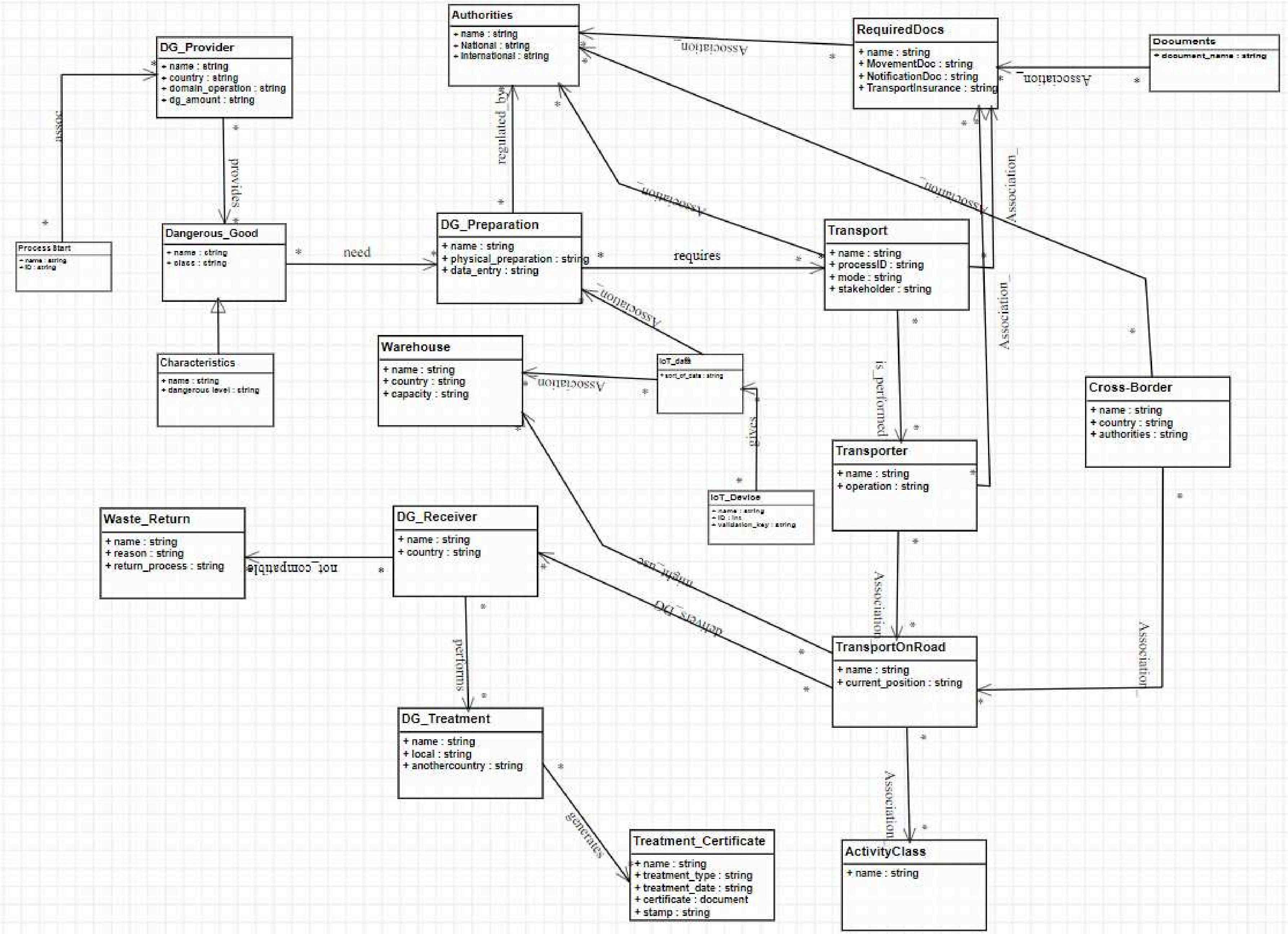

FIGURE 7.1: The domain-specific meta-model is based on the regulatory framework concepts.

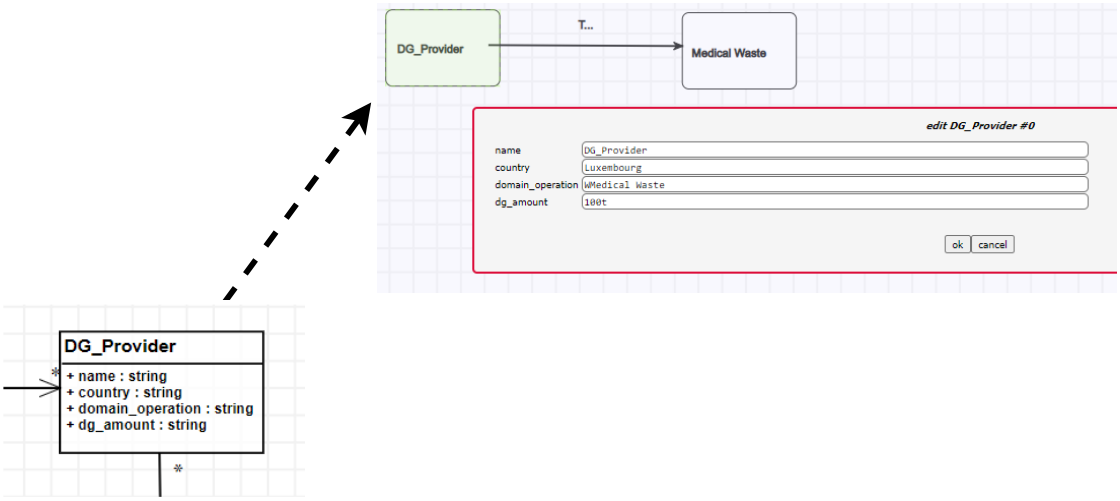

FIGURE 7.2: The representation of the concrete syntax for the meta-model (abstract syntax).

- 2) Definition of the concrete syntax for TMMW models

  The concrete syntax presents a notation that allows the user to express the modeling language. For the defined meta-model presented in the previous section, all of its components and relationships are presented graphically using specific visual objects, thus forming the concrete syntax (Silva, 2015; Cho et al., 2012).

  The "class components" are composed graphically by an icon that presents it and its attributes. There is a dedicated icon for any class component on the meta-model, e.g., an icon similar to BPMN icons that represent it. The relationships are presented by an icon-link that determines the relationship. Its name on the meta-model distinguishes an icon-link on concrete syntax representing any relationship. Figure 7.2 presents the general representation of the concert syntax based on the abstract syntax (meta-model).

- 3) Validate (compatibility of) the model, i.e., the BPMN model based on a meta-model

  The definition of the concrete syntax enables the user models to be defined. The defined model is validated entirely or partially according to the specifications of the domain-specific meta-model. The TDG use case is process-oriented; thus, we consider features for creating a business process modeling language (BPMN) to express it. When defining the concrete syntax, we presented a BPMN-related[3] modeling object that allowed us to show our BPMN model and validate it according to the meta-model.

  Regarding the validation of the models according to the meta-model, we highlight the meta-model's semantics according to the regulatory framework. One of the primary aspects of the regulatory framework was the stakeholders' interaction at different stages of the process. The validation of the model is applied based on the stakeholder interactions and information flows. Figure 7.3 presents the validation aspects for the

[3]In the same way, we might use other standardized modeling languages such as the UML Sequential Diagram A.5.1.

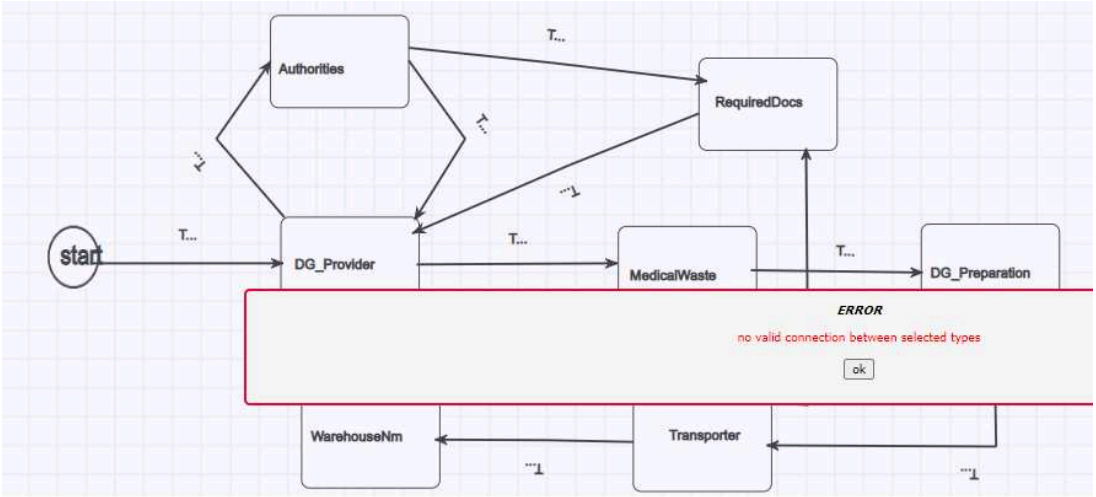

FIGURE 7.3: An example of a throwing error when the connection between components is not compatible with the meta-model.

process of transport of DG i.e., Medical Waste. If an interaction between stakeholders is not defined, an *error message* is triggered. We consider this detection an essential step for compliance validation in terms of validation since the domain-specific meta-model will stand as the backbone of the "targeted" system. The business contract[4] is managed based on the business rules (law article) that are mapped into the meta-model. Any business contract intended to operate with DG at the local or international level is validated based on the domain-specific meta-model. This determines, for a business contract, the achievement of the required compliance level according to the regulatory framework, and beyond this, the fulfillment of additional business-related terms and conditions. For example, an international business contract for TDG requests authorization before the process starts. Without explicit authorization, the "targeted" system will not formally allow the transport process to start; even those indicated in the business contract require the transport process to start on a pre-defined date.

7.2.1 Maintainability: Traceability and Adaptability of Changes in the Regulatory Framework

The proposed approach is based on the regulatory framework, and for the targeted system, it is required to act according to the regulatory framework. A continuous link between regulatory framework and business rules must be maintained to follow the regulatory framework changes. These changes may be at the level of law articles. To achieve maintainability and enable continuous adaptability, we refer to the existing CIM-TDG model (6.3) to define a new meta-model.

Figure 7.4, presents the meta-model composed of components that allow the traceability of the changes in the regulatory framework that are further reflected toward the SC. The

[4]In the context of TDG, a business contract is considered any legal agreement document that intends to arrange DG operations.

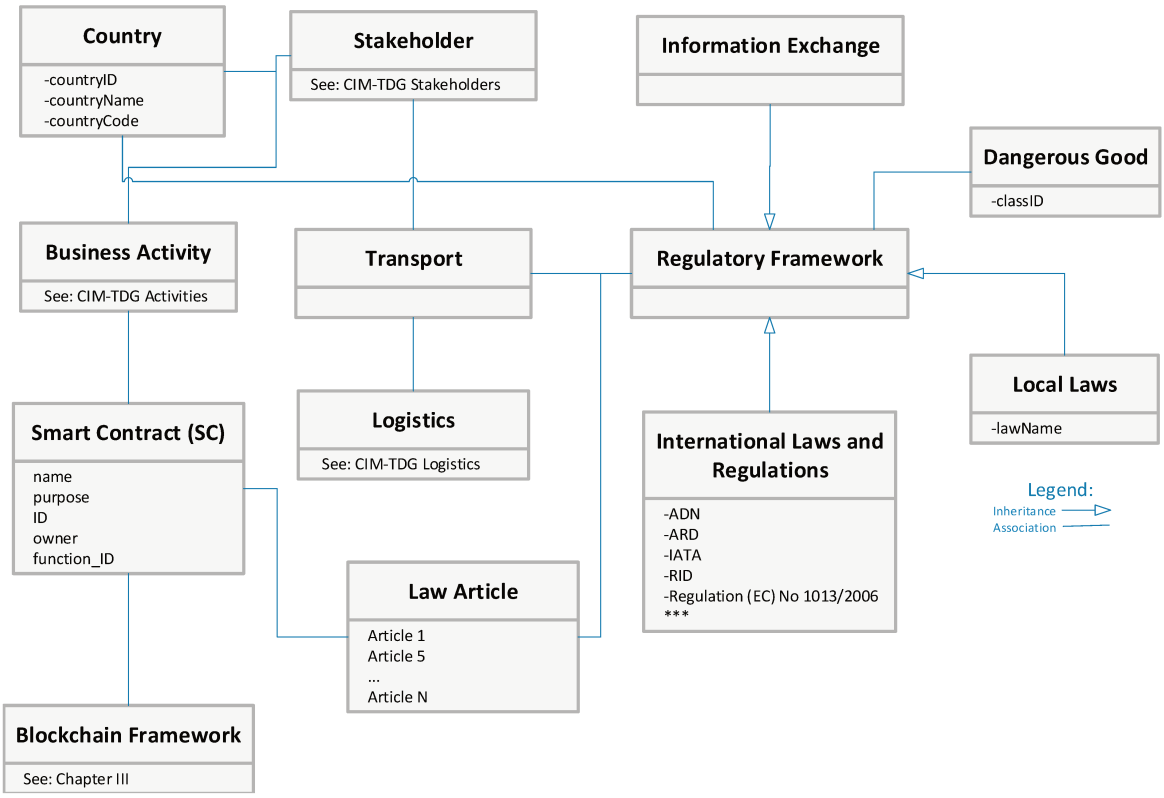

FIGURE 7.4: The meta-model to support the maintainability and adaptability of changes in regulatory framework into SC.

component "Law Article" is associated with the component "Smart Contract (SC)". The components "Law Article" determine the "Smart Contract (SC)" source deployed in BC as part of the targeted system. The SC will act in compliance with the "Regulatory Framework". Any change made on the law articles must be followed (in the other levels of our design method) and reflected in the SC by redeploying SC. With the help of the meta-model, we can identify particularly the concerned SC that needs to be changed (or adapted). For example, if article "X" in the regulatory framework changes, we identify which SC covers these law articles and also its specific function that needs to be adapted. The component "Blockchain Framework" manages the deployment of the SC. The component "Smart Contract (SC)" is associated with the components "Business Activity", which determines different activities that the particular SC is intended to perform, thus helping on identifying the impact caused by changes on law articles.

7.3 L2: Platform-Independent Model (PIM): The Definition of BPMN

We intend to determine the business requirements by developing the user model. The stakeholder interaction and operation are defined based on the knowledge extracted from the previous section. This layer includes the modeling perspective for the TMMW processes, shown in Section 5.3.5. To present the entire process of TMMW, we show the process models separated. The TMMW processes are expressed visually to highlight various stakeholders'

interactions and the process flow. The entire process is composed of activities (events) that take place before starting the TDG, during the TDG process, and after the DG delivery (TDG after). The models presented in this section show sequential steps on the designated processes, conditions, and the timelines specified within processes. The process for the *"Certification of Stakeholders"*, *"Notification (authorization) process"*, *"Transport of MW (move) process"*, and *"Processing (treatment of MW) process"* is presented. On top of these BPMN models, we may define different instances of TDG, including transport and management of general waste, Medical Waste (MW), Nuclear Waste any other related TMDG.

7.3.1 Knowledge Extraction for Composing the BPMN Models

For the specification of the model components, we have distinguished processes, stakeholders, and their interaction. The legal articles from *"Regulation (EC) No 1013/2006"* determine any steps forward (backward) on the TMMW, which are further reflected in the BPMN models. We use the law articles shown in Table 7.1 to describe the workflow in the BPMN model. Figure 7.5 shows the flowchart diagram that describes the translation process of the regulatory framework into the BPMN[5]. This translation is performed manually (by a human expert) by reading and analyzing, and annotating the various relevant concepts from the regulatory framework documents.

7.3.1.1 BPMN: Certification of Stakeholders

A principal condition for the stakeholder's operation in TMMW (also waste management) is being certified (authorized) by the competent local authorities of the country in which the stakeholder operates. Each country provides its own legal rules regarding the certification of stakeholders. Since our use case is taken from the Luxembourg context, we present the stakeholders' certification based on the regulatory framework proposed by the competent authorities in Luxembourg, i.e., the *Ministry of Environment*, which is responsible for "waste management". The procedures described in (Guichet.lu, 2020) combined with those at the international level, e.g., *Regulation EC 1013/2006*, are followed in the stakeholder certification process.

7.3.1.2 BPMN: Certification for Waste Trader (Broker)

In order to be certified, waste traders (broker) need to fulfill several conditions. Figure 7.6, shows the certification process for traders (or brokers). In the model presented, the process starts with the "Request for Permit" task, related to the gateway, which indicates the "New Request" for a permit or "Resubmission" of current dossier if it was "refused" by the authorities in the first application. For a "New Request", the application process starts by collecting the documents required to complete the "dossier" requested by the legal authorities. Furthermore, the applicants send the dossier to the relevant competent authority, such as the "Environment Agency" (Environment-Agency, 2020) the legal authority in Luxembourg,

[5]To facilitate understanding of the process of knowledge extraction, an algorithmic expression is showed in Appendix A.3

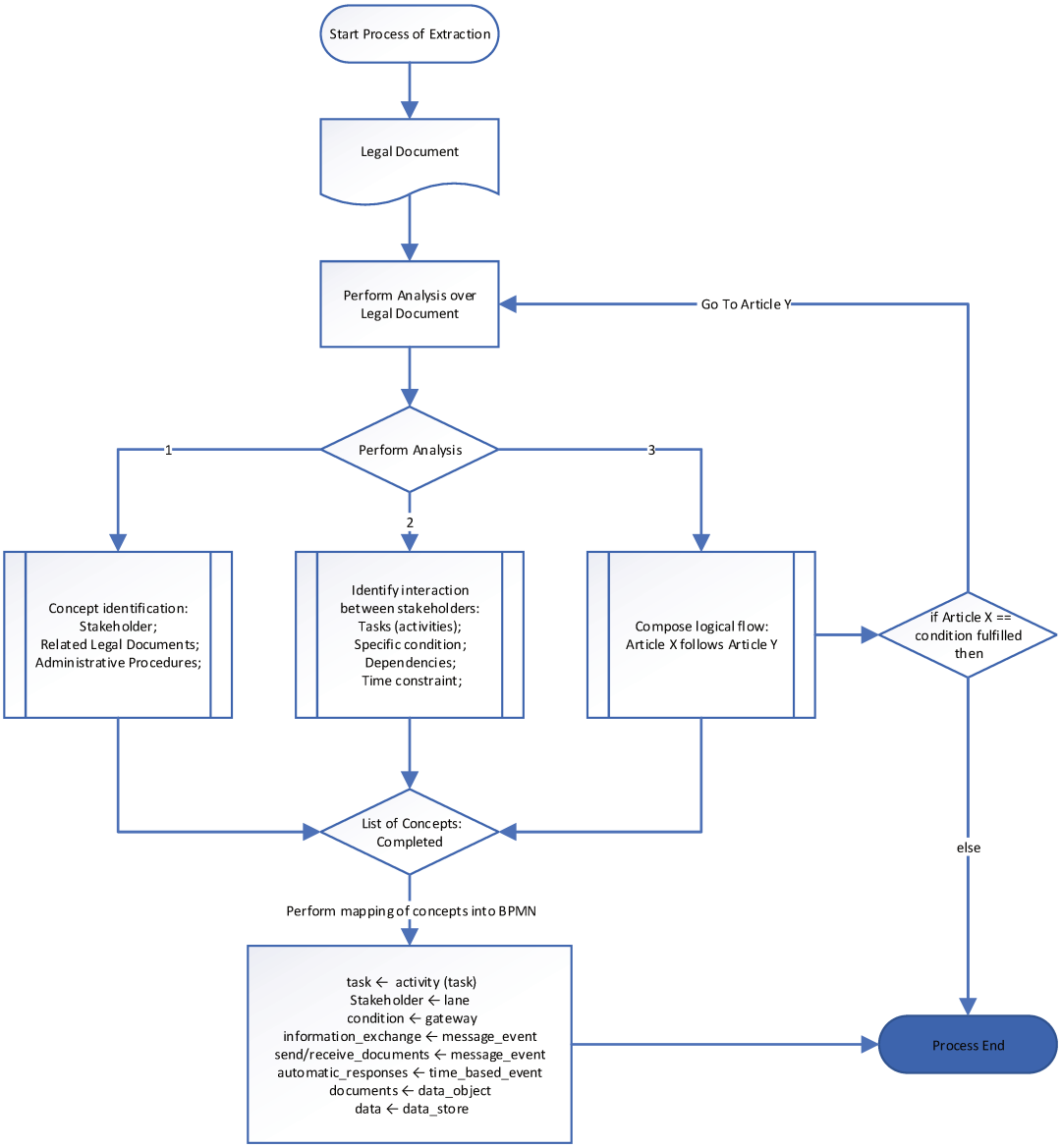

FIGURE 7.5: The flow chart diagram for mapping the BPMN model from legal documents (text).

as described in (Guichet.lu, 2020). When the authorities receive the dossier, they first check whether it can be accepted (admissible), according to the requirements described (Guichet.lu, 2020). If the dossier cannot be accepted (inadmissible) for specific reasons supported by the legal and regulatory documents, they immediately notify the applicant. The application is considered acceptable (admissible) if all requirements are fulfilled or if the authorities fail to notify the applicant about the dossier acceptability within fifteen days of the dossier being received date. Further, after informing the applicant of the acceptability, the authorities must evaluate the dossier within 15 days and send their final response to the applicant. The possible responses are "permit granted", "permit refused", or "partial refusal". If the "Re-submission" task is selected, then the trader (broker) needs to collect the specific documents for the dossier. The completed dossier must be submitted within 60 days. If the dossier is received within this period, it will be evaluated by the authorities. The applicant might require additional time, in accordance with the laws. If the submitted dossier is accepted after meeting the conditions of the competent authorities, it follows the evaluation procedures, and receives one of the following responses: "permit granted", " permit refused", or "partial refusal".

7.3.1.2.1 BPMN: Waste Collector and Transporter Certification The waste collector and transporter certification procedures are almost the same as those of the broker (trader). The main difference is in the pre-requisites imposed by the regulatory framework. The waste collector (transporter) must have had a contract with the Waste Producer or with a certified Broker or Trader (WasteCollection, 2020). Figure 7.7 highlights the difference in the Waste Collector (Transporter) certification process, while the other procedures are the same as those in the Waste Trader (Broker) certification process.

7.3.1.3 BPMN: Notification of Authorization or Rejection

The authorization (notification) process is that with the most complicated, time-consuming, and intense human resources involvement. The stakeholders, waste collector, or waste trader/broker (in general, DG Provider), require authorization from the competent authorities to operate in the transport and management DG (TMMW). The authorization procedures, conditions, and other related obligations are regulated by the regulatory framework document *"Regulation No 1013/2006 (EC)"* (Commission and Parliament, 2006). Figure 7.8, shows all the required steps in the notification process, entirely based on the articles of "Regulation No 1013/2006 (EC)", as shown in Table 7.1. The flow charts presented in the authorization BPMN model are obtained from the law articles[6]. These articles determine logical flow, condition, and events, which are further are expressed based on the BPMN notion, according to the Algorithm presented in 7.5. For a more detailed illustration and as required by BPMN 2.0 rules, the stakeholders involved in this process are presented with BPMN pools, and message events manage all communications and documents exchanged by stakeholders.

[6]In the following, we will refer to the articles of "Regulation No 1013/2006 (EC)" with standard legal article references, e.g., we will refer to Article 2 with (2), and when we refer to a specific point in the article, e.g., Article 2, point 3, the reference will be 2(3).

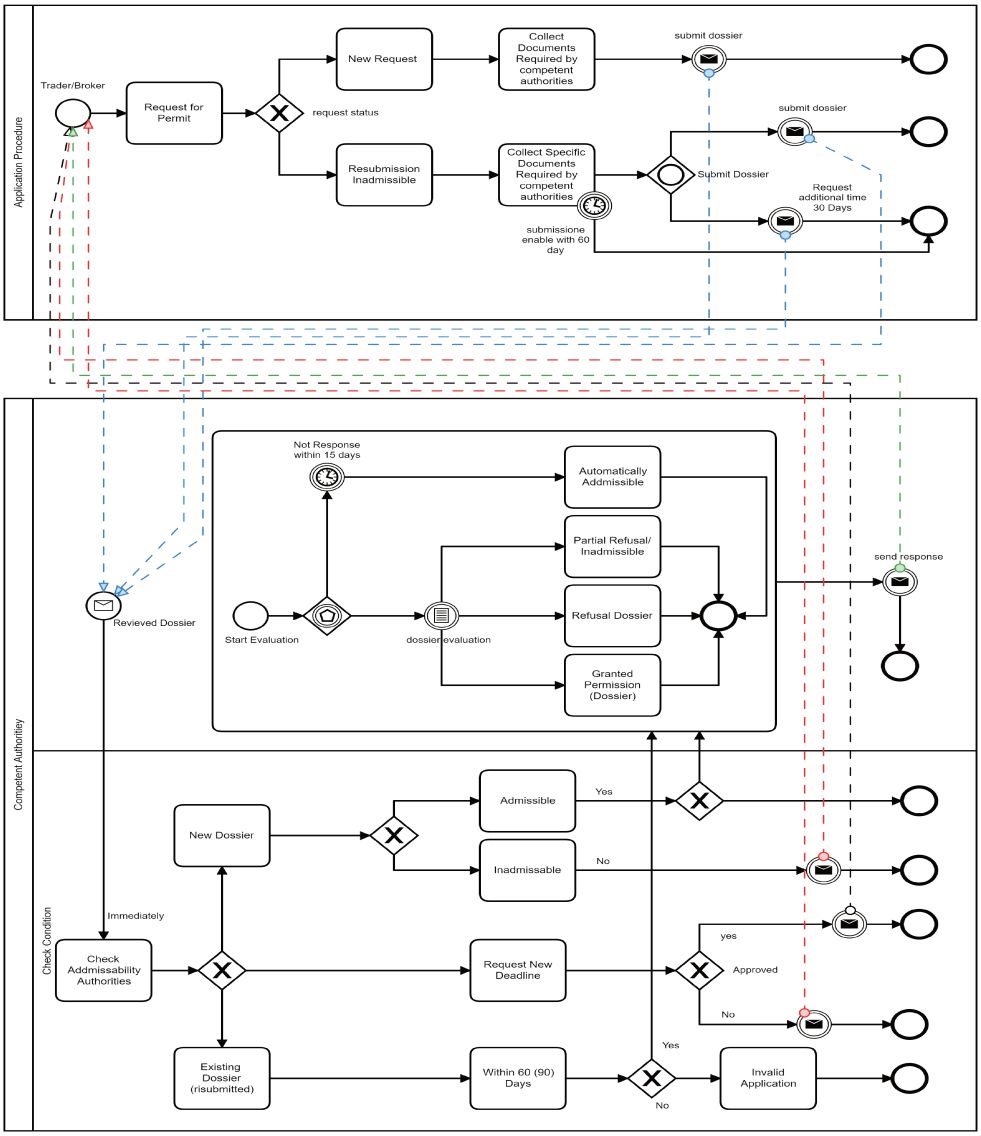

FIGURE 7.6: The broker certification (trader) process.

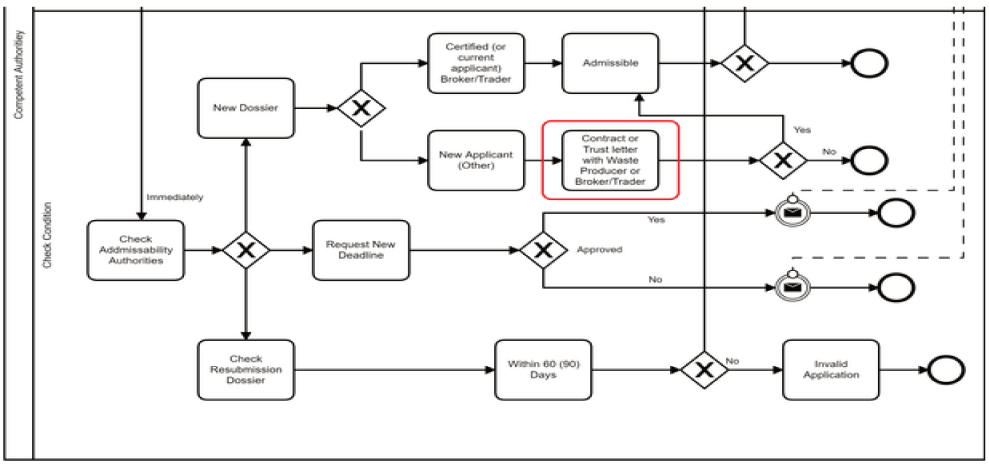

FIGURE 7.7: The Waste Collector (or Transporter) certification process.

The process was initiated when a certain amount of DG (waste), e.g., MW, required a particular treatment such as *disposal* or *recovery operations*. The notification process is required for any waste, including MW, that is subject to disposal, and for specific waste in the case of recovery, as described in (3). For the *new request* for notification, the notifier (DG Provider) described in (2), collects the required documents. The list of documents required for the dossier is defined in articles (4) and (5). Among the documents, the notifier must provide the *"notification and movement"* documents with the required information sections completed 4(2). These documents and the related information in their annexes are presented in Appendix A.11. The notifier sends the dossier to the competent authority of the dispatch country. After receiving the dossier, the dispatch authorities must react within three days by providing the notifier with one of the following responses:

i) refuse the dossier immediately due to obstacles described in (11), (12) or (5);
ii) request additional information 4(3);
iii) notification of the sharing of the dossier with the transit and destination authorities (7).

Further, when the transit and destination authorities receive the dossier within three days, they may ask for additional documents 8(1) and inform the other authorities involved regarding their request.

If the destination and transit authorities do not respond within 30 days, they are obliged to provide a reasonable explanation to the notifier 7(8), stating a valid reason for the delay. After receiving the requested information, the destination and transit authorities send their agreement based on (9). The possible conditions for waste shipment are specified in the notification dossier as described in (10).

7.3.1.4 BPMN: The Process of Transport (Movement of DG)

The process of transport (or movement) of waste, i.e., MW can start just after the notification dossier is accepted by the competent authorities of dispatch (also transit and destination).

The process of transport (movement) of waste, i.e., MW is presented in Figure 7.9. In this business process, we identify the "DG Provider", this is the headline we use to refer to the "Waste Provider," and "Transporter", and covers any certified carrier of MW.

The process starts with the completion of the movement document A.11, by inserting the date of the shipment 16(a), before submitting it to the competent authorities of dispatch (transit and destination) 16(b), and to the consignee (Dangerous Goods Receive). The "Transporter" prepares the MW for transport, as well as the accompanying documentation required (notification document, movement document, and proper labeling (according to Basel Convention, article 4)). The continuation of the process is based on the fulfillment of the condition specified in the movement document. The transporter should always have the movement and notification document 16(c) present when transporting (carrying) MW.

During the transport, if an intermediate stop is required, then the authorities should be notified in advance. They should also be informed immediately in the event of any unpredictable change in routing.

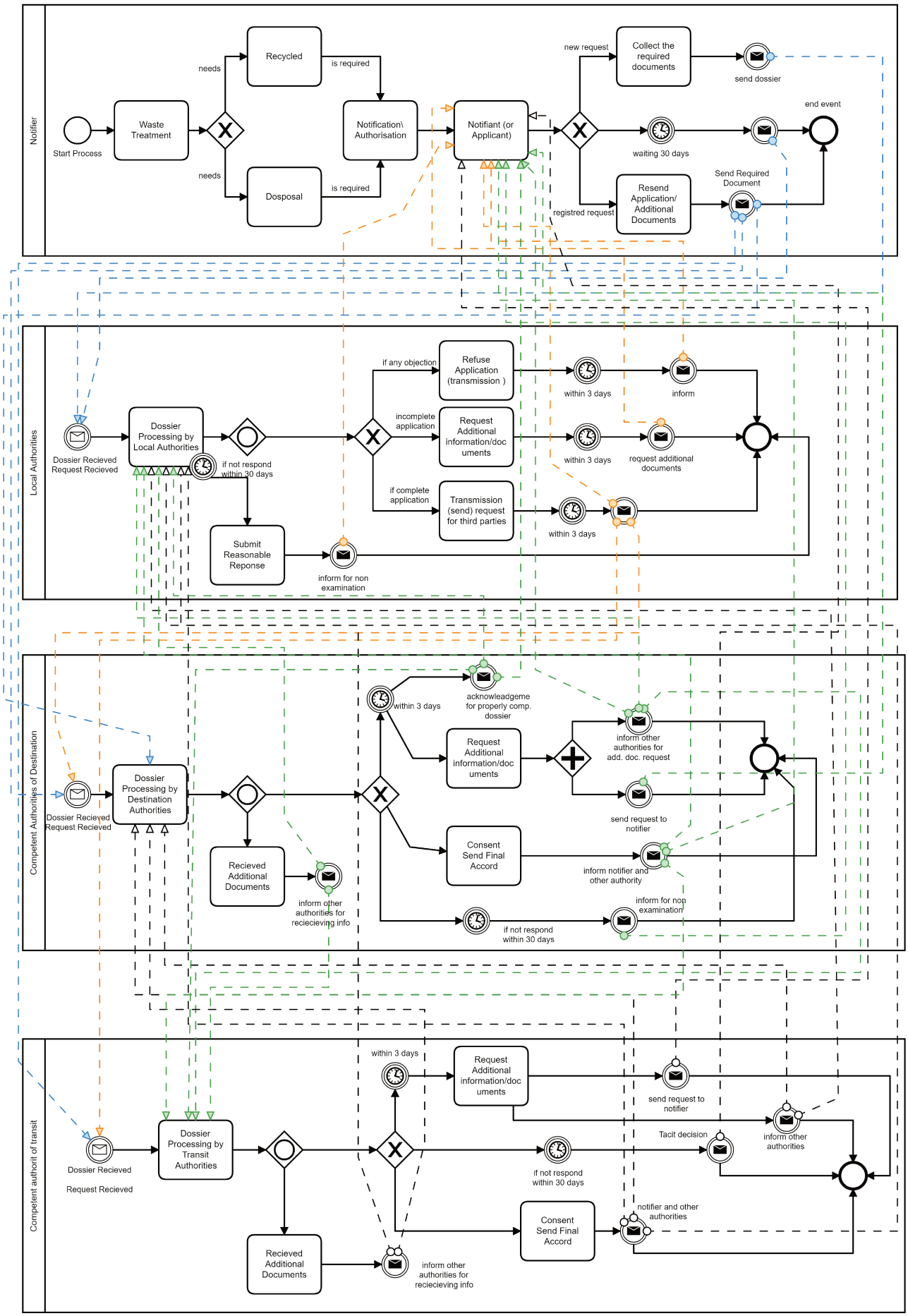

FIGURE 7.8: The process of notification (authorization) for the transport and management of MW at the international level.

7.3.1.5 BPMN: The Process after Transport of the MW

This process is illustrated in Figure 7.9 in the "P4 Dangerous Goods Receiver" diagram. This is a separate process, but for the purpose of interaction (exchanging information) between stakeholders, we have integrated it into the transport process.

When the MW is received from the consignee (DG Receiver), it is obliged to inform the authorities (local and international) of the MW received within three days 16(d). Further, the consignee (DG Receiver) checks the type of waste, to verify that the received MW is in accordance with the signed contract (5) with the waste provider. If the waste is different to what was agreed in the contract, the waste provider is obliged to take back (return) the waste (22)(23)(24)(25). Otherwise, if the waste is in accordance with the contract signed, then the consignee proceeds with the MW treatment. The waste must be treated within one year (10). If the treatment of the MW is performed by the same treatment facility, then immediately after treatment and at most after 30 days, the consignee should send the certificate of treatment 15(d). If the treatment of MW is performed in the same country but at another facility 15(e), the certificate of treatment should be sent within one year (10). If the treatment is intended to be performed in another country, then a new notification process is required 15(f).

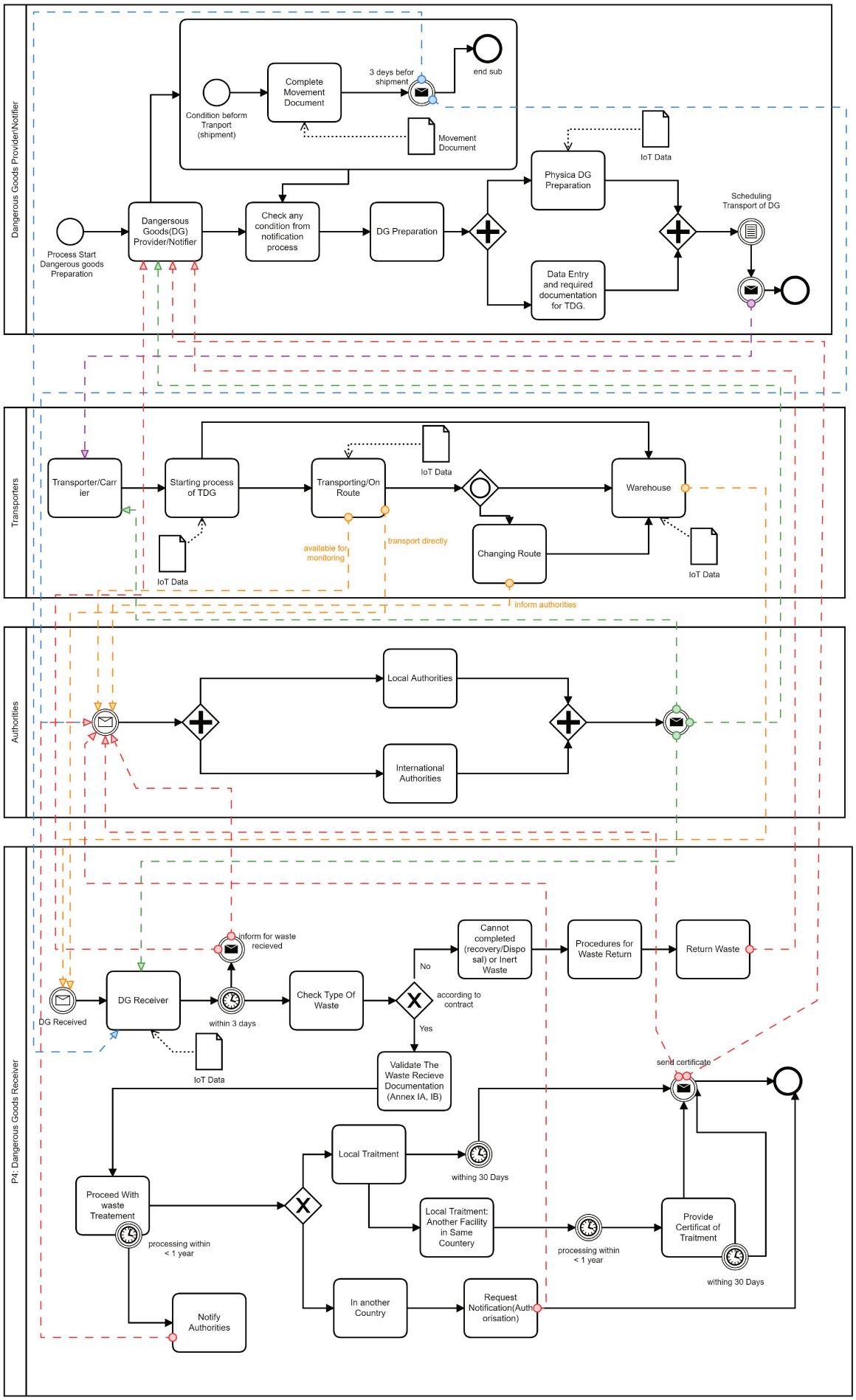

FIGURE 7.9: An end-to-end BPMN model for the MW (DG) transport process.

7.4 Transformation from PIM Smart Contract Model (PISCM) to PS Smart Contract Model (PSSCM)

The following sections address the remaining layers of the proposed design method (L3 - L6). To compose the (L3) layer, the previous layer (L2) is transformed from BPMN into a UML Sequential Diagram (USD). This transformation allows us to define the inner interaction schemas between the targeted system and its components. These schemas are formed based on the activities required in the use case for TMMW. The interaction or exchange of information between the system and its components is performed based on the function as a means for information (input/outputs) treatment and for imposing specific conditions on the process flow. We call these schemas Function Exchanged Diagrams (FED), and the corresponding algorithm expressions are extracted from these schemas. The FED allows the exchange of any information on the system, and in order to specify the right to access and manipulation information, we present a general access control policy.

Further, we define a platform-independent system architecture (L4), which collects and digitally presents each targeted system component functionality and its interaction with systems. Subsequently, the deployment architecture is refined to support the future deployment of the functional system architecture.

Layer (L5) is entirely technology-related. It determines the platform elements (L5), including the technology platform and its main characteristics, IoT devices (or IoT platform), the specification of technological components (sensors and other devices), end-user interface, and the access control policy definition. Finally, in (L6), we present a code generation model based on the technical specifications included in (L5).

7.4.1 BPMN to UML Sequence Diagram (USD) Model Transformation

The BPMN models presented in the previous layer (7.3), besides presenting the way the operations are performed in the TDG (i.e., TMMW) process, also describe a foundation for the targeted system specification functionalities. Currently, as they are presented, in particular, they do not show specific concepts, such as the time required for a single step in the BPMN model, function calls (details of events and sub-events, data generations), the behavioral aspects of the interaction of the stakeholders and the components of targeted system, in detail. So, the targeted system components (artifacts) are avoided or not exposed at the BPMN level. To achieve this, we propose a transformation of the BPMN model into a UML sequence diagram (USD) (A.5.1.0.2). We intend to clarify and begin to specify our targeted system in more technical aspects by performing a model transformation.

The core idea behind the BPMN to USD transformation is to associate BPMN model notation with a more detailed behavior description of the interactions (information exchange) between stakeholders and targeted system components. This is a starting point for our design method to enable the system designer or any software engineer to interpret the model more efficiently to fill the gap with any underlying software artifacts of the targeted system. The transformation of the BPMN models into the USD serves to extract a detailed interaction between stakeholders and other system components in the targeted system. We define and

map these interactions in terms of the functional exchange that outlines the entire system "functionalities", indicating the system behavior. Besides this, designing the interactions between stakeholders through the "targeted" system and other related components allow us to improve the current (existing process for TMMW) system by proposing new and more sophisticated digitalized steps to validate and exchange information between the "targeted" system components. This transformation allows us to express the problem domain (showed by BPMN models) into future software development artifacts. We used the BPMN model to analyze and discover requirements. An explicit motivation for transforming the BPMN into the USD in the context of TMMW is to examine the required time for a particular process. The USD supports the time issues such as time constraints (or time duration constraint) (OMG-Superstructure, 2014)[7]. Besides the time-related options given by the BPMN notion of time, expressed by "Timer Event", it does not quantify (count) the entire time needed to complete an end-to-end process. In the context of TDG (TMMW), time is a crucial aspect, and we consider the core components required to model and evaluate at the design level in our design method.

The MT of the BPMN into USD is common practice for system designers and software engineers to design more sophisticated software. Several researchers have proposed a transformation from BPMN into USD. A classical MT is proposed in (Kardoš and Drozdová, 2010; Rhazali et al., 2016) by considering the MDA approach for MT. The research from (Suchenia et al., 2017) shows BPMN's transformation into USD and intends to support BPMN in the representation of time issues. It proposes a transformation algorithm for manually converging BPMN into the USD that is general and useful for a wider audience (analysts, software engineers). In (Bouzidi et al., 2020), BPMN is transformed into a USD by proposing semi-automatic transformation based on MDA. It uses standardized BPMN as a source model (CIM*) and the UML standard to retrieve the targeted model (PIM). The transformation approach is based on the MVC design pattern, and it maintains the trace between the source and the target model. The transformation from BPMN to USD is performed in each canonical fragment of the BPMN model.

We propose transforming the BPMN model into the USD, based on the transformation rules in the QVT (OMG-QVT, 2016) jargon, shown in Table 7.2. The left-hand column presents BPMN elements, the middle column presents the USD components corresponding to the BPMN elements, and the right-hand column presents the functional exchange diagram (FED) elements. This transformation is specific to our use case (TMMW), while a general approach for transforming BPMN to a USD might be found in literature (Suchenia et al., 2017; Shi and Lin, 2019; Nassar et al., 2017).

We call the newly developed USD a "functional exchange diagram (FED)". The FED describes how and in what order the system functionalities are organized by depicting the interaction between system components and further considering the time required for each sequential step. The FED illustrates a dynamic behavior by drawing the exchange of information between system components to complete a given process (task). The FED

[7]On page 518, more detailed information is provided regarding the UML sequence diagram and time constraints.

TABLE 7.2: The transformation rules (semantics) for BPMN to the UML Sequence Diagram (USD).

Source model (BPMN Components)	**Target model (UML Sequence Component)**	**Description in technical terms (Functional Exchange Diagram (FED))**
Stakeholders\objects Lane Sub-process	Lifeline	Core system component
Sequential Flow Message Event Exclusive gateway Start Event	Message flow Return message	Function input/output and condition
Parallel gateway	Parallel message flow	Function input/output and condition
Data object	Return Message	Function (self) input/output and condition
Inclusive gateway	Single or multiple message flow	Function input/output and condition

presents how the system will behave according to the received task. Using the FED, we intend to understand the *inner event (messages) exchanges in the system*, *impose constraints*, and add additional *components* that we might discover while building scenarios.

7.5 L3: Platform-Independent Smart Contract Model (PISCM)

The global process of TMMW is composed of three different sub-processes: the "process before the TMMW (P0 and P1)", the "process during the TMMW (P2)" and the "process after TMMW (P3)" as detailed in section 5.3.5. This layer shows the platform-independent model for the specification of the SC. We apply this model to each of the processes (P0 - P3) of the TMMW.

This layer offers the system designer the opportunity to express all possible scenarios required for TMMW (TDG) and, beyond that, to improve the entire process. Besides mapping the requirements extracted from the regulatory framework (L1 - L2), the system designer might use this layer to create scenarios that advance the current TMMW by adding the necessary components to the given process. For illustrations such as scenarios, we propose using the FED (7.4.1) to allow us to present the interaction between system components, which are further intended to be SC functions. We refer here to SC as an algorithmic component that executes some business logic (rules, constraints, process flow) that is technology-independent at this stage. The PISCM offers the specification for the offered services intended to be part of the targeted system. At the PISCM level, we do not show any technology components so that any platform can implement our PISCM in the future (beyond BC technology).

7.5.1 PISCM: Smart Contract Specification Model for the Process Before TMMW

These processes are mainly administrative, forcing the stakeholder to comply with the regulatory framework. Figure 7.10 presents the FED schema of interactions between stakeholders (actors) and the stakeholder-system in the context of "stakeholder certification". From this schema, we obtain the functions of the SC, according to the exchange scenarios between stakeholders and system components. Each lifeline's allocated function symbolizes an SC function (or, in a specific case, a condition inside the same function). We present the platform-independent, smart contract (PISC) algorithmic model (pseudo code) representing SC model, functions, and algorithmic flow elements for each FED schema.

PISCM: "Stakeholder Validation": This SC model aims to check and automatically validate the credentials (digital account) of the stakeholders. The validation can be done if the stakeholder is registered in the system according to the requirements described in the regulatory framework and expressed in the BPMN (7.3.1.1). If the stakeholder, e.g., Waste Provider (DG_Provider), is not already registered on the system, the SC will deny any activity and provide instructions for new registration and validation opportunities by giving the option of registering with the system. Further, it has a mechanism to validate documents, request additional documents, and finally complete the registration in cooperation with the "Authorities" as described in the legal documents (7.3.1.1). The interaction schema for the stakeholder validation is shown in Figure 7.10 and the algorithmic representation is shown in 1. The algorithmic representation specified for this case presents the core model of the future SC, and may be further supported by additional components (SC functions) in order to achieve the desired functionality for the process of TMMW.

The main global parameter of this PISC algorithmic model is *start_TDG_activity*, which covers any new TDG activity. In our model, we start any process by considering that the stakeholders are already registered (*registered = true and registration = false*), until notice to the contrary, which denies further activities in the system by using *deny_TDG_ativity = false*. In this case, the variable *registration = true* becomes true, and the procedures for the registration of stakeholders are initiated. A stakeholder is identified by a unique number (*stakeholder_ID*) completes a *dossier* which is then transmitted to the local competent authorities (*authorities_ID*). This also enables an exchange of information between the stakeholder (*stakeholder_ID*) and authorities (*authorities_ID*) until the registration is completed or entirely denied.

In general, the processes before the TMMW (TDG) are mainly concerned with the stakeholders obtaining authorization from the authorities to allow the transportation of the DG. The authorization process is one of the most complicated, and includes much interaction between the involved stakeholders. Figure 7.11 presents just[8] some characteristics of the authorization process expressed in the FED schema. In 2, we present the PISC algorithmic model for the notification process. In addition to the provisional presentation of the global parameters, in this algorithm, we have *notification_Doc* and *movement_Doc* parameters, which presents key documents that need to be filled out by the stakeholder (*stakeholder_ID*). The authorization application is possible only for the registered (validated) stakeholder (line

[8]For brevity reasons, in the image, we are not able to show the entire FED schema for the process of authorization. It is shown entirely in the following link: https://git.list.lu/adnan_imeri/thesisaimeri

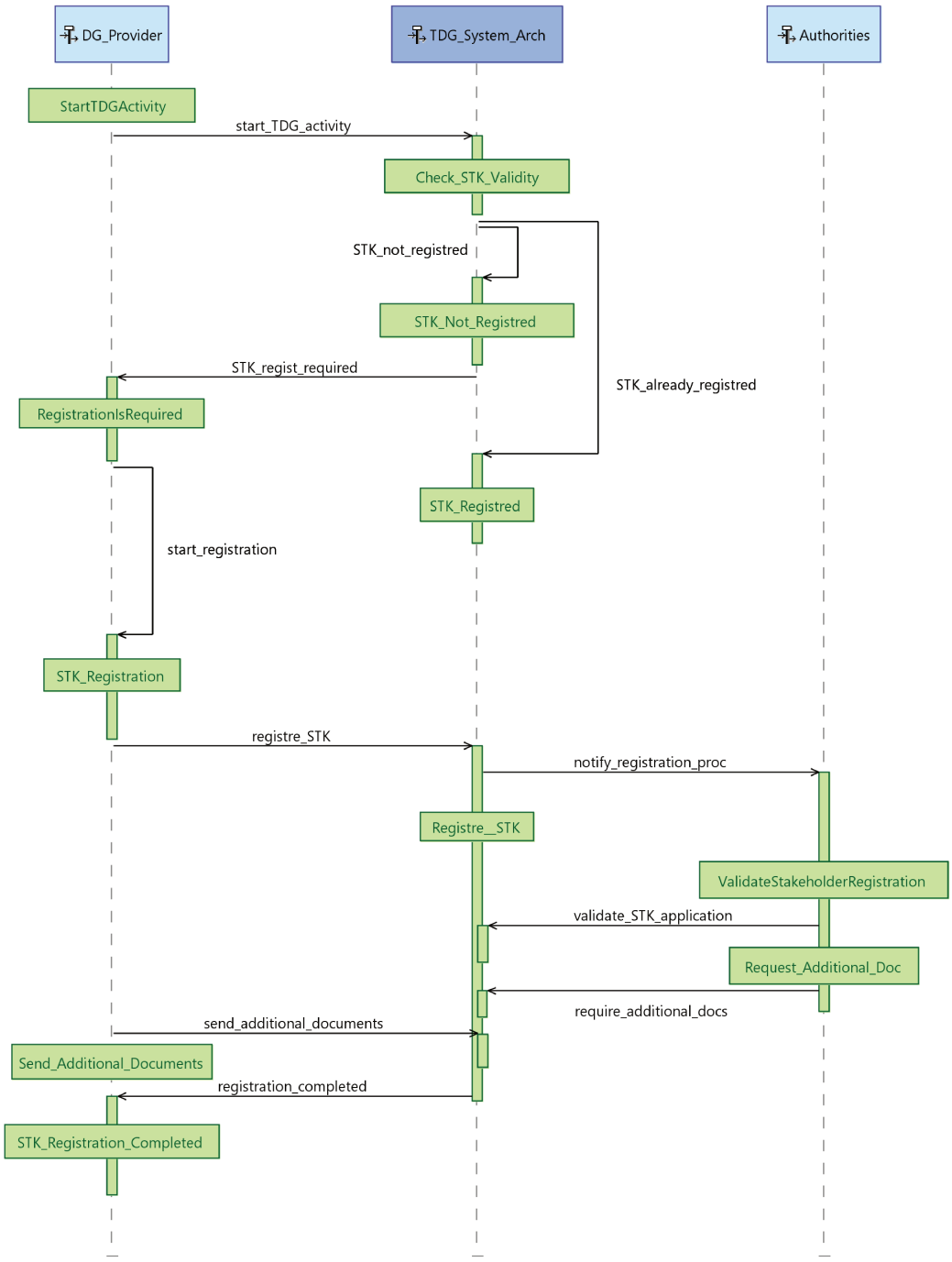

FIGURE 7.10: The FED for the stakeholder registration and validation.

10). If the stakeholder intends to apply for authorization (and is not already registered), the system will deny any activity and inform them of the registration procedure (line 12). Otherwise, the stakeholder might start completing the authorization dossier and then sends it to the competent local authorities (line 14-15). If any document is missing in the dossier evaluated by competent authorities, it is required to be provided by stakeholders. Once the dossier has been completed and approved by the competent local authorities, it is shared with the transit and destination competent authorities (line 20). Furthermore, once the dossier has been received by the competent authorities involved, they might require additional information (lines 23-26) otherwise, the authorization is given (line 31).

Algorithm 1: PISCM: PISC algorithmic model for Stakeholder registration, verification, and authorization into the system by Authorities.

```
1  Initialization:
2  start_TDG_activity;
3  registered = true;
4  deny_TDG_ativity = false;
5  registration = false;
6  stakeholder_ID;
7  authorities_ID;
8  systemSearchForSTK ← start_TDG_activity(stakeholder_ID);
9  if (registred == false) then
10     deny_TDG_ativity = true
11 else
12     registration = true;
13     while (registration) do
14         stakeholder_ID ← composeDossier;
15         sendDossier (authorities_ID);
16         if (!completedDossier) then
17             stakeholder_ID ← requireAdditionalDocuments;
18         else
19             stakeholder_ID ← registration_completed
20         end
21     end
22     abort_registration
23 end
```

Related to the authorization process, we specify an SC model directly linked to the DG characteristics, named *SmartContract_DG* . For any stakeholders that intend to transport or manage DG, they should introduce (register) the DG, e.g., MW. Once the DG is introduced, the competent authorities (and DG provider) highlight the main risk characteristics such as type of DG, risk level (sensitive parameters, e.g., high temperature, humidity, disturbance), quantity, and many others relating to the DG.

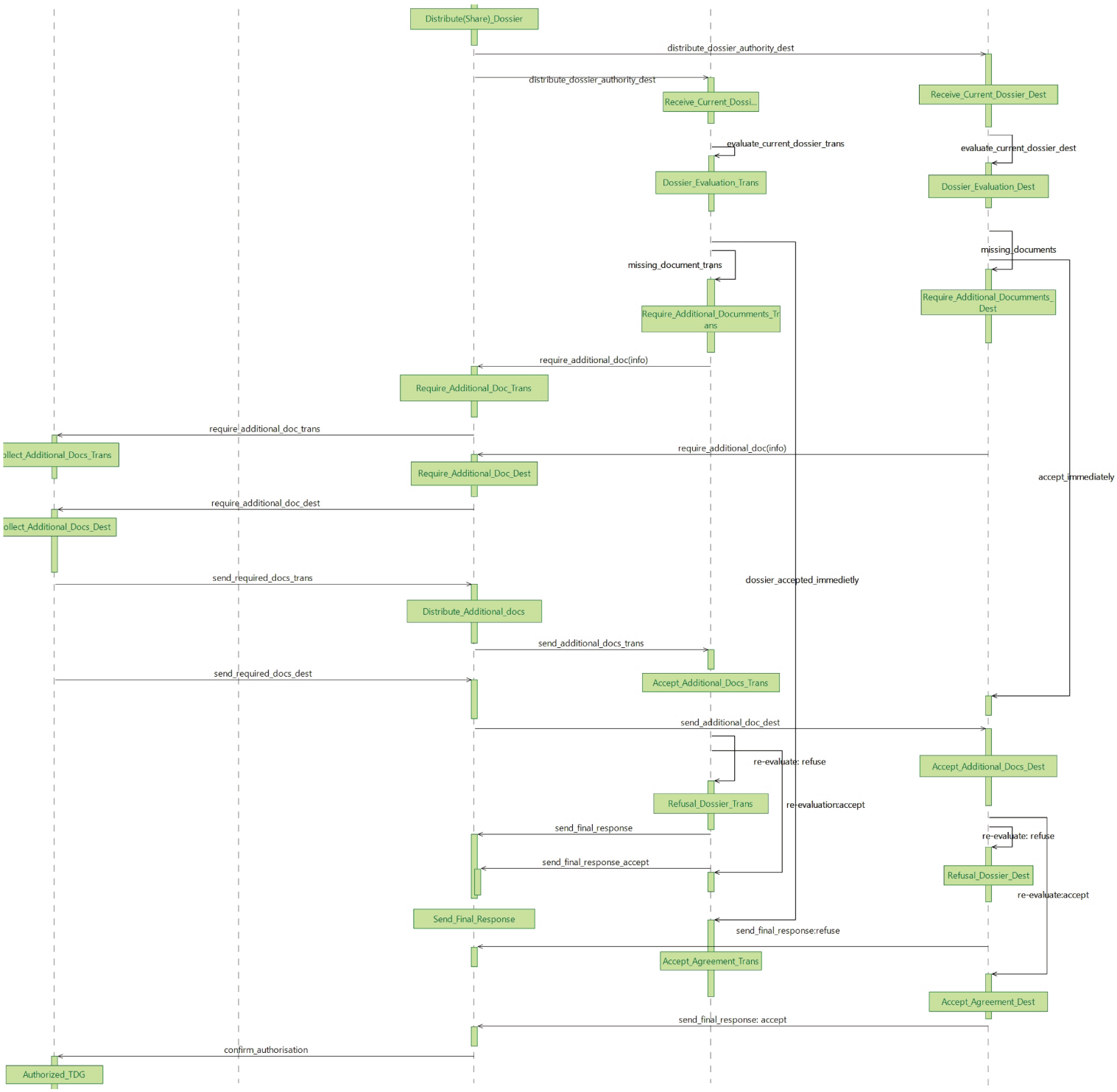

FIGURE 7.11: The FED for the TMMW (TDG) authorization (notification) process.

Algorithm 2: PISCM: PISC algorithmic model for process of authorization (notification) for TDG (TMMW).

```
1  Initialization:
2  start_TDG_activity;
3  registered = true;
4  deny_TDG_ativity = false;
5  registration = false;
6  stakeholder_ID;
7  movement_Doc;
8  notification_Doc;
9  systemSearchForSTK ← start_TDG_activity(stakeholder_ID);
10 if (registred == false) then
11     deny_TDG_ativity = true;
12     registration ← Algorithm1(stakeholder_ID) ;
13 else
14     composeDossier_ID ← (notification_Doc)&&other_docs;
15     sendDossier (stakeholder_ID, notification_Doc, movement_Doc,..) ;
16     LocalCompetentAuthorities ← EvaluateDossier;
17     if (!completedDossier) then
18         stakeholder_ID ← requireAdditionalDocuments;
19         LocalCompententAuthoritis ← requireAdditionalDocuments
20     else
21         ShareDossier (TransitCompententAuthoritis,
            DestinationCompententAuthoritis);
22         TransitCompetentAuthorities ← EvaluateDossier;
23         DestinationCompetentAuthorities ← EvaluateDossierc;
24         if (!completedDossier or AdditionalDocumentsRequired) then
25             TransitCompententAuthoritis ← requireAdditionalDocuments;
26             DestinationCompententAuthoritis ← requireAdditionalDocuments;
27             requireAdditionalDocuments(LocalCompententAuthoritis,
                stakeholder_ID);
28         else
29             notifiy_for_evaluation_dossier(LocalCompententAuthoritis,
                stakeholder_ID)
30         end
31         stakeholder_ID ← Get_authorisation
32     end
33     abort_authorisation_request
34 end
```

7.5.2 PISCM: During Transportation Smart Contract Specification Model

Following the same approach as in the previous section, we present the PISC algorithmic model shown in 3 for the process model during TMMW. This algorithmic model uses its functions for the advancement of the TMMW digitalization process. The *DG_Process_Initialization_SC (process_Initialization)* function presents one of the main functionalities of this SC model: the initialization of the TDG process. For each TDG, an *identified (ID) process* will be initialized. This function also informs the involved stakeholders about the start of the process. This process (ID) remains open, and all interactions, for example, the exchange of information with authorities or between stakeholders, will be identified with the process (ID). There might be a situation in which the TDG process can start while DG might be stored in a Warehouse (an intermediate stop) and remains there for a certain time, e.g., several weeks, before it is delivered. In this case, the process remains open until this DG is eventually delivered to the contractual destination, i.e., "DG Receiver premise". This SC model maintains the TDG process workflow effectively. The *SC_DG_Transport* is responsible for authorizing the start of the process of transport of DG. It provides all the information about the transport modalities, such as the type of DG to transport, the itinerary information, and the truck-related information. This information is valuable for the authorities responsible for monitoring the movement of DG and identifying transport means. For the international context of TDG, a cross-border situation is evident. There are plenty of scenarios that trigger a cross-border situation, and for the management of the TDG process, we consider SC components that highlight such situations. The *SC_DG_Cross_Border* presents a checking point when the truck arrives at the border of a country, and it automatically informs the stakeholders, i.e. *Authorities* of both countries and the *DG Receiver*. Further, we propose a specific function called *SC_DG_Cross_Warehouse*, which provides the necessary information about the warehouse facility, such as the warehouse's location, current capacity to host the DG to be stored, information on the arrival date (and time) of the DG, and availability for maintaining the state of the DG with the safety conditions required (expressed by *SmartContract_DG*). The *SC_Dist_Info*, shares information for the involved stakeholders, if it has been correlated for this purpose, with *SC_Dist_Info (certificate)*.

In keeping with the digitization of the process and improvement of the monitoring aspects (track and trace), a significant role is given to the *SC_IoT_Device* function, whose role is to provide real-time information. The *SC_IoT_Device* is employed (invoked) from other different SC such as *SC_DG_Transport*, *SC_DG_Cross_Warehouse*, *SC_DG SC_Cross_Border* and *SC_DG_Process_Destination*.

The SC *SC_Alert (Notification)*, presents a mechanism that notifies stakeholders according to their role. Each stakeholder will receive a message (notification) according to the event that occurred. The alerting or notification might take place in an emergency or for other events happening when in the TDG.

This SC alerts stakeholders when an emergency occurs:

- when the risk parameters are matched, e.g., high temperature and a high accident probability.

- when an accident has happened (detected by a combination of information from temperature (humidity) and disturbance sensors), the emergency alarm is raised immediately with the responsible stakeholders, e.g., "Authorities","Emergency Responses".
- when business contractual rules are violated

Algorithm 3: PISCM: The PISC algorithmic model for process during the TDG (TMMW).

```
1  Initialization:
2  start_TDG_activity;
3  registred = true;
4  deny_TDG_ativity = false;
5  registration = false;
6  stakeholder_ID = true;
7  stakeholder_status;
8  movement_Doc;
9  notification_Doc;
10 authoristaion_proc;
11 process_Initialization;
12 transport_Process;
13 detected_issues;
14 if (registred == false or !movement_Doc or stakeholder_status == false) then
15     deny_TDG_ativity = true;
16     registration ← Algorithm1(stakeholder_ID) ;
17     authoristaion_proc ← Algorithm2(stakeholder_ID);
18 else
19     process_Initialization ←
        DG_Process_Initialization_SC(processID, stakeholder_ID);
20     transport_Process = SC_DG_Transport
       (process_Initialization, transportMeans);
21     distribute_information (stakeholder_ID);
22     process_information ←
        (IoT_data, current_location, SC_DG_Cross_Warehouse (current_location),
        SC_DG_Cross_Border (current_location))
23     distribute_information (stakeholder_ID)
24     if (detected_issues (error_CODE)) then
25         error_CODE ← detected_issues  SC_Alert (error_CODE)
26     else
27         everything_normal
28     end
29 end
```

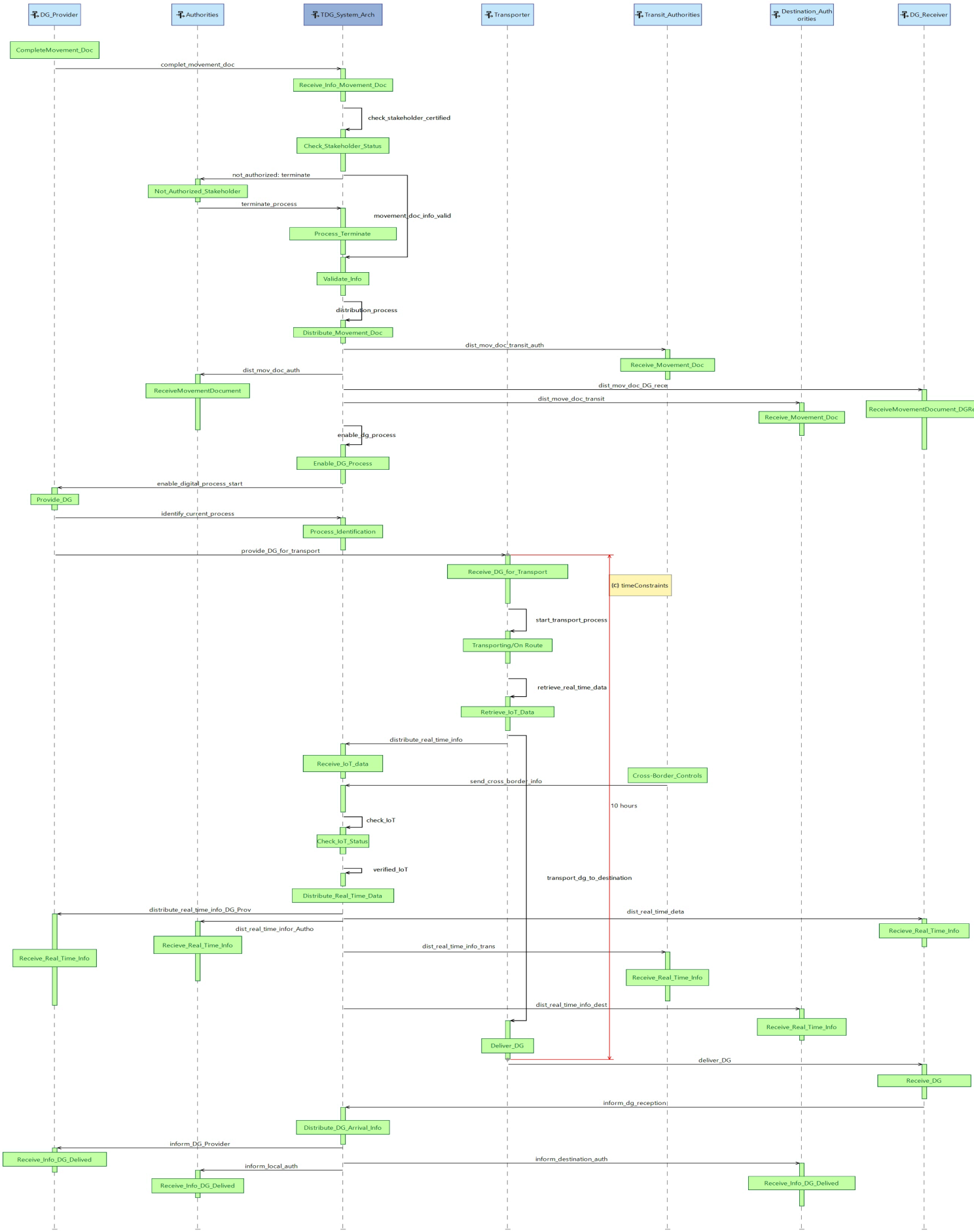

FIGURE 7.12: The FED for the process during the TMMW (TDG).

7.5.3 PISCM: After-transportation Smart Contract Specification Model

The SC model known as *SC_DG_Process_Destination* specifies the process after the DG is delivered. When the *DG Receiver* receives the DG, this SC model automatically informs other stakeholders, e.g., the "Authorities" and the "DG provider". Furthermore, it provides the "DG Receiver" on DG delivery with all available authentication information for the vehicle location, vehicle information and DG package identification (e.g., using RFID), and the driver's digital signature. This process is made possible with the help of information transmitted by various IoT devices deployed in the environment and other information available to the system. After this stage of DG delivery, the procedure continues by checking the DG type, and if the DG is not in accordance with the contract signed by the stakeholders, it might start operations to return the DG to the place of origin. The SC *DT_Treatment* encapsulates several functions that are related to the DG treatment. The FED schema for the process after TDG is presented in Figure 7.13, and the PISC algorithmic model for this scheme is presented in 4. In addition, this algorithmic model specifies parameters, the *contractual_Terms*, which indicate the list of contractual business terms enlisted from the stakeholders (line 16), and *return_procedures*, which indicate the return of the DG that is not in accordance with the initial contract (line 17). The *treatment_DG* indicates the way in which the DG are processed, for example, if they are processed in the same country, or if they are sent to another country for final processing (lines 20-25). The *certificate* is the official document created based on the process of treatment of the DG, and it is shared with the involved stakeholders.

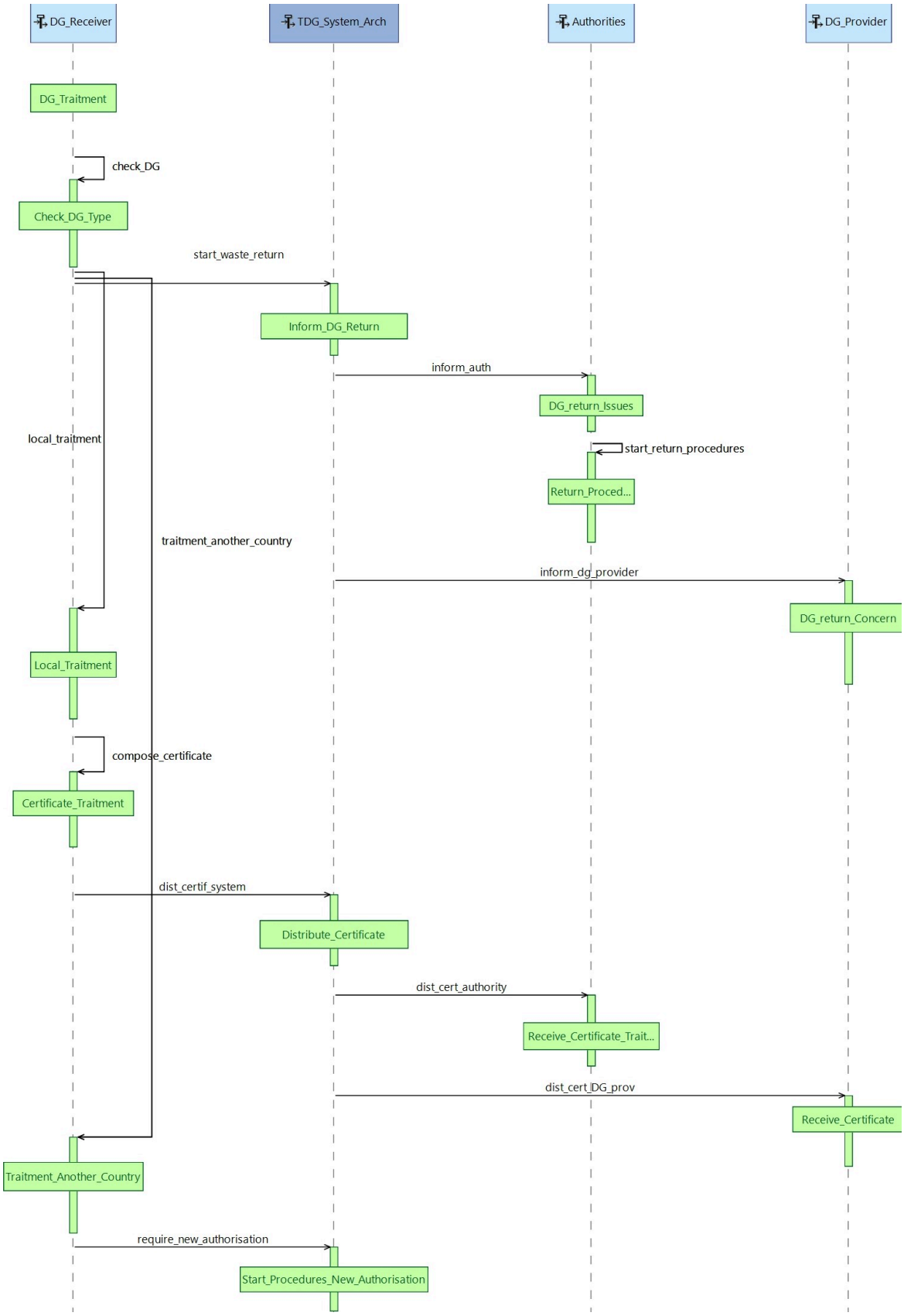

FIGURE 7.13: The FED for the process after the TMMW (TDG).

Algorithm 4: PISCM: The PISC algorithmic model for the process after DG delivery.

```
1  Initialization:
2  start_TDG_activity;
3  registred = true;
4  deny_TDG_ativity = false;
5  registration = false;
6  delivery_DG;
7  contractual_Terms = true;
8  return_procedures;
9  treatment_DG;
10 certificate;
11 if (registred == false) then
12     deny_TDG_ativity = true;
13     registration ← Algorithm1(stakeholder_ID) ;
14 else
15     if (SC_DG_Process_Destination (delivery_DG) == true) then
16         if (contractual_Terms == false) then
17             SC_DG_Process_Destination(return_procedures);
18         else
19             SC_DG_Process_Destination (treatment_DG);
20         end
21         if (treatment_DG == local) then
22             certificate ← compose_certificate;
23             SC_Dist_Info (stakeholder_ID);
24         else
25             Invoke Algorithm2
26         end
27     else
28         Waiting for DG
29     end
30 end
```

7.5.4 Managing Time Constraints at the Design Level

Time management represents a crucial component for the regular and manageable process for the TMMW (TDG). It is related to the organization of the processes in TMMW. The process needs to specify the time needed to transport DG from one place to another, the time needed to arrive at the border of the destination (or transit) country in order to be able to enter that country with DG, and many other time-dependent cases. When designing a TDG-related system, time should be considered at the design level. In the proposed FED, we impose the time constraint that will be necessary for a specific part of the process or end-to-end process. We apply time-related constraints in the transport process showed in Figure 7.12.

The red line indicates the time e.g., 10 hours needed to complete the transport process from *"Receiving_DG_for_Transport"*, until *"Deliver_DG"*. If the transport process is not completed within the time period anticipated, it may indicate a design issue in the system, which needs to be resolved before continuing.

7.5.5 Role-Based Access Control

Access control[9] is a security technique that determines the accessibility to system components and resources. Access control aims to manage access to specific resources in the system by restricting it to certain system users. This restriction is based on specific criteria, which determine the accessibility of system components based on roles, thus forming an access policy. The access policy requires a clear definition at the design level to clarify accessibility in data and other digital objects. The concept of the "Role" was introduced in the "CIM-TDG Activities" meta-model (6.3.3.4). The "Role" concept presents the user role from the stakeholder perspective. Any specific "business activity" requires a particular role to handle activities in the TDG. In general, the stakeholders define "business activities" that are further associated with the role. There is a "treatment" (as a specific activity) for the defined role that they are authorized to perform. The stakeholders determine the set of "treatments", and they may be diverse and large in terms of the activities to be performed by each role. To formalize this, we introduce a specific definition called *"Categories"* that represents the set of treatments that a role can perform over it. Furthermore, we define each role's access policy. This determines the set of treatments the role can perform. With the help of first-order logic (predicate logic) (A.8.1), we formally present the concept of roles, categories, treatments, and access-control policy as follows:

$R-$ *is the set of all roles*;
$C-$ *is the set of all categories*;
$T-$ *presents the set of all treatments*;
$ACP-$ *presents the access-control policy*;

The categories *(C)* are defined by the stakeholders. These categories are defined based on the need of the stakeholders according to the TDG process requirements. A set of categories is composed as follows:

$C = c_1 \leftarrow$ *Process_Administrator*, $c_2 \leftarrow$ *Advisers (controller)*, $c_3 \leftarrow$ *Monitorer,*
$c_4 \leftarrow$ *Evaluator*, $c_5 \leftarrow$ *Business_Contractor*, $c_6 \leftarrow$ *Transporter*, $c_7 \leftarrow$ *DG_Provider*,
$c_8 \leftarrow$ *DG_Receiver*, $c_9 \leftarrow$ *Emergency_Responders*, $c_n \leftarrow$ ...

These categories belong to different stakeholders and have a specific role to perform for TDG activity based on their operations. We propose such categories since for any of them, e.g., $c_1 =$ *"Process_Administrator"*, there might be different roles defined within the same category ($r_i \subseteq c_i$). This means that in the *"Process_Administrator"* category, some roles can

[9]Related work studies for role-based access control and BC are presented in section 4.4.2.

only manage a particular part of the process, for example, application for authorization, but cannot see contractual business information.

Initially,

$$\text{Let } R = \{R_0, r_1, r_2, r_3, r_4, ..., r_n\} \text{ be the set of all possible roles.}$$

$$\exists!\ R_0 \in R \tag{7.1}$$

which presents the *administrator* of the system, thus

$$\{r_1, r_2, r_3, ...r_n\} \subset R_0.$$

The R_0 belongs to a specific stakeholder (or group of stakeholders) based on a *"consortium agreement"* that they may reach while defining future TDG system control. R_0 has the privileges to create, update and maintain the *access-policy*. Furthermore, it is responsible for deploying the SC and performing privileged actions according to stakeholders determination.

For any defined role, it belongs to a category:

$$(\forall r \in R \wedge r \neq R_0)(\exists\ c \in C) \therefore c \overset{belongs}{\leftarrow} r$$

We define the set of treatments (T) (or activities) that determine the group of treatments (activities) that it is possible to execute from the role. It allows the definition of which treatments are executed from a specific role. We recall that in the TDG, we distinguish three kinds of activities that are performed according to the process specification. Consequently, we have a set of activities to be fulfilled before the transport starts (${A_{bf}}^p$), activities during the transport process (${A_{dr}}^p$) and the activities after the transport process (${A_{af}}^p$). All treatments (activities) in TDG belong to one of these global TDG-relented process activities.

For each of these activities (and other sub-activities), corresponding transactions are associated. Suppose, T_a *is an activity for validation of dossier in process of notification*. For such activity in we have an associated transaction T_x:

$$T_x \leftarrow T_a$$

Figure 7.14, visually presents the link between category, treatment (activities) and transactions. Each category may have several roles, and each role may execute (perform) several activities. For each activity, a transaction is associated.

Determining treatments and grouping them into a set $A = \{{A_{bf}}^p; {A_{dr}}^p; {A_{af}}^p\}$, allows us to determine the exact SC that will be associated with each activity. This allows us to determine any SC execution i.e., its functions, by any role (user), by knowing the exactly which roles execute which treatment.

The access-control policy (ACP) is composed of elements that allow the role to perform a specific operation (action) on the system. The $ACP = \{execute, query, view\}$, hence,

"Query q " – can query data transactions in the system (a history of transactions);

"View v" – can monitor, in real-time, the current transactions;

"Execute e" – can execute SC functions in the BC (read and write transaction).

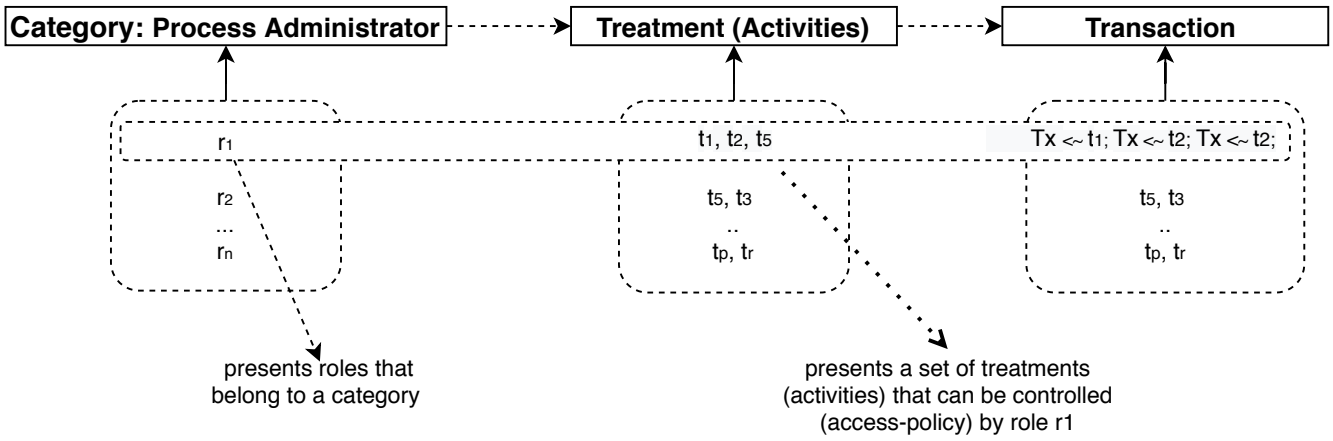

FIGURE 7.14: The diagram for presenting the relationship between category (role)-treatment (activity) and transactions.

The ACP is applied over the SC functions intended to perform a specific action (activity) in the BC. A role can execute an SC function according to its defined ACP.

Therefore,

$$ACP = \{q, v, e\} \text{ then}$$
$$SC_{fn} \leftarrow ACP(SC, r, c)$$

Table 7.3 presents roles-category and their related access policy. For the defined role (r_i) that belongs to a category (c_i) the access-policy is defined according to treatments and stakeholder determinations. The stakeholder maps the access policy (7.3) for all the roles, according to the following criteria:

$$(\forall r \in R) \wedge (\exists t_i \in T) \wedge (\exists\, q, v, e \in ACP) \implies [(q \vee v \vee e) \leftarrow ACP(r_i, t_i)]$$

For example, let

$r_i \in R$ be a role i that is associated with a specific category c_i.

Further, the treatment set determines the activity of the role according to where it belongs in T. The combination of the role and its treatments implies that the role is determined according to its treatments. Therefore, when applying the ACP for role determination, we have:

$$(r \in R) \wedge ((t_i \in T \wedge (1 \leq i \leqslant 3) \in A)) \implies r \leftarrow t_i$$
$$\therefore (q, v, e) \leftarrow ACP(r_i)$$

In particular the intention is to *determine their privileges over the SC functions for any role*. Keeping with this, we determine that, for example $c_7 \leftarrow$ *Emergency_Responders* will be able to read (access) data transactions only at the treatment phase ${A_{dr}}^p$ and ${A_{af}}^p$. This because these services are necessary only during the transport and treatment (warehousing) phase of TDG. Furthermore, it will execute a particular SC, e.g., *Emergency_Response_SC* function (in

TABLE 7.3: Definition of roles, categories and the access-control policy for the TDG.

Role\Category	c_1	c_2	...	c_7
r_1	$(q.SC_{fn_{Name}})$ $\vee$ $(v.SC_{fn_{Name}})$ $\vee$ $(e.SC_{fn_{Name}})$			
r_2				
r_3	$(v.SC_{fn_{Name}})$	$(v.SC_{fn_{Name}})$		$(q.SC_{fn_{Name}})$ $\vee$ $(v.SC_{fn_{Name}})$ $\vee$ $(e.SC_{fn_{Name}})$
r_4			$(q.SC_{fn_{Name}})$ $\vee$ $(v.SC_{fn_{Name}})$	
...				
r_n				

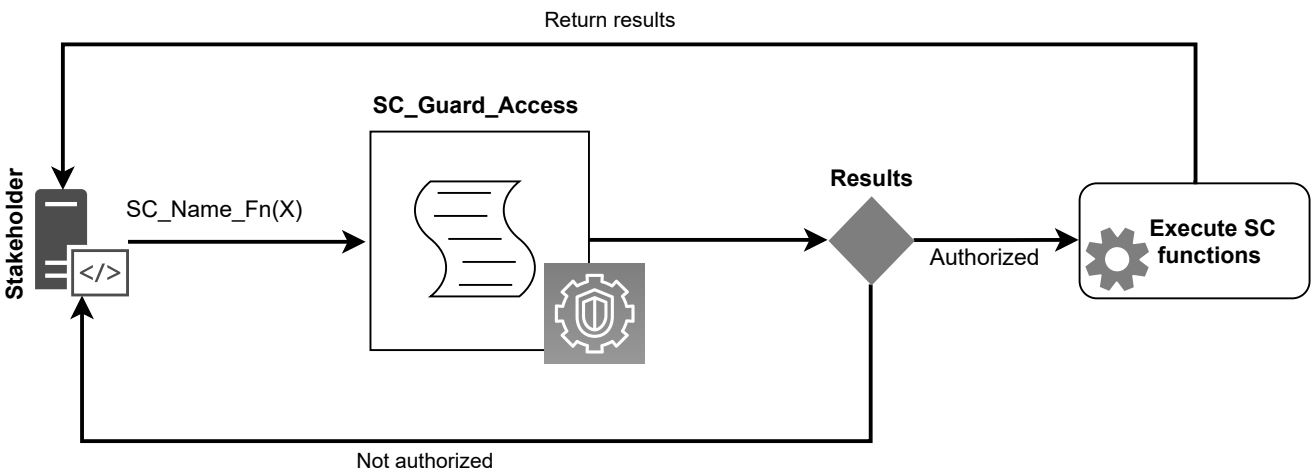

FIGURE 7.15: The role-based access control policy (ACP) and the notion of "guard" smart contract.

treatment $A_{dr}{}^{p}$) only in the case of an emergency and only this particular SC can read and write transnational data for the possible emergency.

In terms of applying the above-defined method for role-based access policy into a BC-based system, we note that for any existing BC address (user) (U), a role is associated, thus $U \subseteq R$. To set the ACP as presented in Table 7.3, we use a specific SC e.g., *"SC_Policy_Update"*, administrated from R_0. This determines the category and aim of the role. Further, to manage the ACP, we use an SC, called *"SC_Guard_Access"* which is used (invoked) when any stakeholder performs an action (activity). In Figure 7.15, we present the notion of "Guard" SC, called *SC_Guard_Access*. Initially, it checks and validates the targeted SC by verifying its name, e.g., *"SC_Name_Fn(X)"* and additional information (owner, purpose, function ids). Once validated, it further performs algorithmic validation of the ACP by checking if a role attempts to execute an SC function (fn) by controlling the given role's ACP. If the user is not authorized to execute that particular SC function, it returns *false: not authorized*. Otherwise,

it allows the user to execute the SC function and to obtain the results according to its ACP. In general, the "Guard" SC serves as a gateway to determine the role-based ACP for the involved stakeholders.

7.6 L4: Platform Independent System Architecture (PISA)

The interaction schema presented in the previous layer enables an overview of the interactions between system components. At the highest level, the L3 (7.5) exposes the operational interactions between system components. These interactions are carried from the FED schema, which compose essential components of the system architecture. Following this layer, we present the system architecture independently of any technology. It is composed of system components, their interaction and information flow. The purpose of this layer is to define what the system intends to accomplish and how its components are described. It presents the inner part of the system context and explores the stakeholder's needs based on the previous layer. For defining the platform-independent system architecture (PISA) we perform MT according to the transformation rules shown in Table 7.4.

Figure 7.16 shows system architecture[10], which consists of the digital representation of stakeholders as an integral part of the system. These digital representations are known as digital system components. As already mentioned, the digital system components are connected to the *"TGD_System_Arch"*, which presents the core of the system. The *"TGD_System_Arch"* can be further translated into a specific technology, e.g., Hyperledger Fabric. This indicates that any interaction between system components uses *"TGD_System_Arch"* as a communication mechanism. The *"TGD_System_Arch"* system itself remains at the *"center of everything"*, to manage any communication between digital objects, and the storage of information, and remains an active component to support any interaction between system components and system-physical worlds (through IoT).

In the PISA model presented, any component is defined as a digital object or digital twin (DT). We introduced the DT concept in Section 3.7. This indicated that any interaction with the system is based on these DT. The stakeholder can communicate, access shared information, and provide relevant information only if its DT is defined and validated by the stakeholders' consortium. This provides a mechanism to manage the TDG process in an end-to-end manner. The core component *"TGD_System_Arch"* retrieves information continuously from system components, e.g., "Transporter" (with the help of an IoT device) acts independently most of the time, self-reacting by sending information when the DG arrives at the destination. The system reaction is based on the definition given at the design level (FED schema).

The involvement of DT in our study aims to theoretically and practically analyze the TDG by exploiting advanced technologies such as BC and IoT. We propose using DT to design a virtual system that represents the real-world system for SpCDG. The concept of DT is suitable in a situation where many stakeholders cooperate together to fulfill the SpC requirements. It

[10]This figure presents a lighter version of the system architecture for the TDG. The completed version of the system architecture is shown in the following link: https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/_SAB__Completed_TDG_System_Architecture_View.jpg.

TABLE 7.4: The transformation rules for FED schema to platform independent system architecture.

Source model (FED schema component)	**Target model (PISA component)**
Lifeline	Digital system component
Function component	Component inside the digital system component
Message exchange (information flow)	Relationship between digital system components

is a situation where interactions are required to exchange important information (Mandolla et al., 2019), which is one of the core research case studies of this thesis. In this sense, the intended DT aims to facilitate the representation of the processes in TDG in an end-to-end manner, including the processes before the transport of DG, processes during the TDG, and processes after the TDG, which are reflections of the current real-world processes in the TDG. The proposed DT will include the stakeholder's involvements, end-to-end data flow schema and stakeholders interaction, and other related processes, such as monitoring the transport in real-time. Designing such DT allows us to find the current drawbacks of the current running system and to improve the real-world system based on its twin in virtual space. That allows us to design a simulated SpC operation by interacting with the real-world system. In this way, before developing such an extensive complex system, DT enables us to capture the current real-world system's drawback, maximize performance, increase traceability and transparency. Consequently, the DT provides convergence between real-world data and virtual data, thus avoiding scenarios where data are isolated outside of virtual reality and the operations diverge from a real-world system.

7.6.1 PISA: Deployment Architecture

Designing a BC-based solution requires considering many challenges in terms of performance issues, the capability of being able to support the business process entirely or partially, and also of the deployment of the physical architecture. The deployment architecture in the context of this layer indents to clarify the physical deployment of the designed architecture (7.16). The main questions when designing a BC-based system also include "*who needs to have a full BC node?*", "*who needs to have a lightweight node*[11]*?*" and "*who needs to only have access to the BC via some user interface*?" In the context of our design method, we propose that the main stakeholders, i.e., authorities, should have a full BC node. The other stakeholders such as "DG Provide", "Transporter", and "DG Receiver" host a lightweight node. The "Emergency Responders", might have just simple access to the full BC nodes by using the BC portal user interface. Figure 7.17 presents a physical deployment architecture for the TDG (TMMW) system architecture. The "Blockchain Nodes" components are nodes that the respective stakeholder administrates. They represent the full BC node (BC node without any limitation) and may be deployed on the premises of the stakeholders (geographically distributed), thus

[11]The BC lightweight node do not contain the full ledger of transaction or perform consensus mechanism. It is allowed to read transaction on BC and trigger SC (Xu et al., 2018).

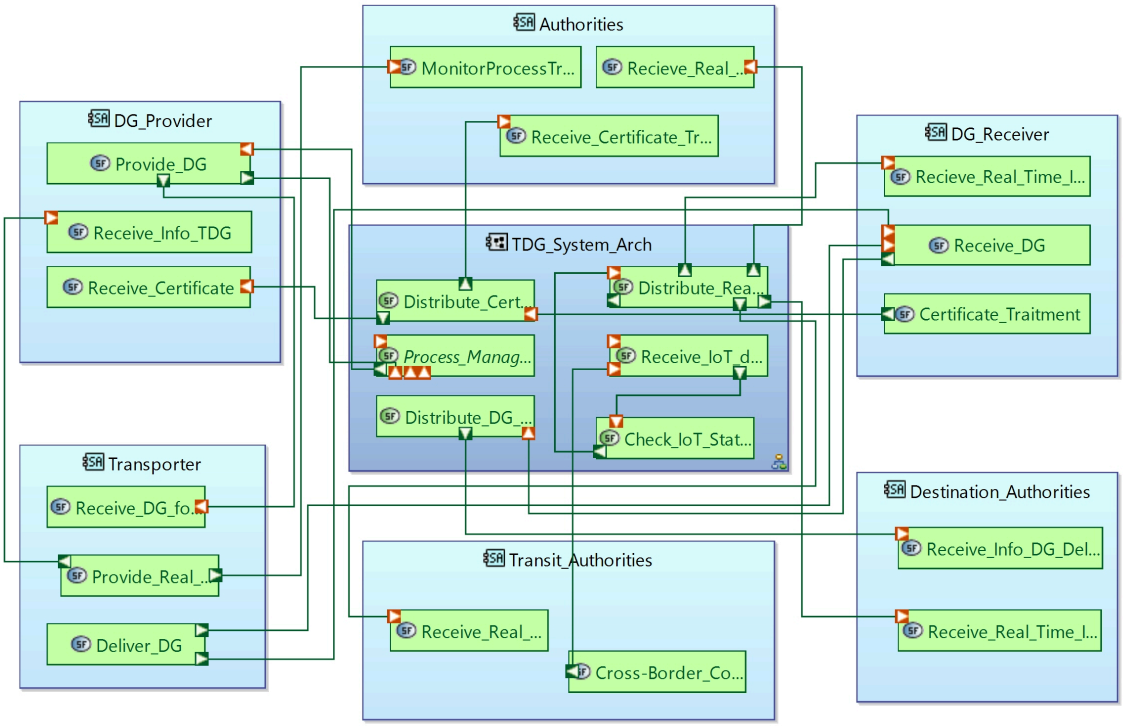

FIGURE 7.16: The platform-independent system architecture for the TDG.

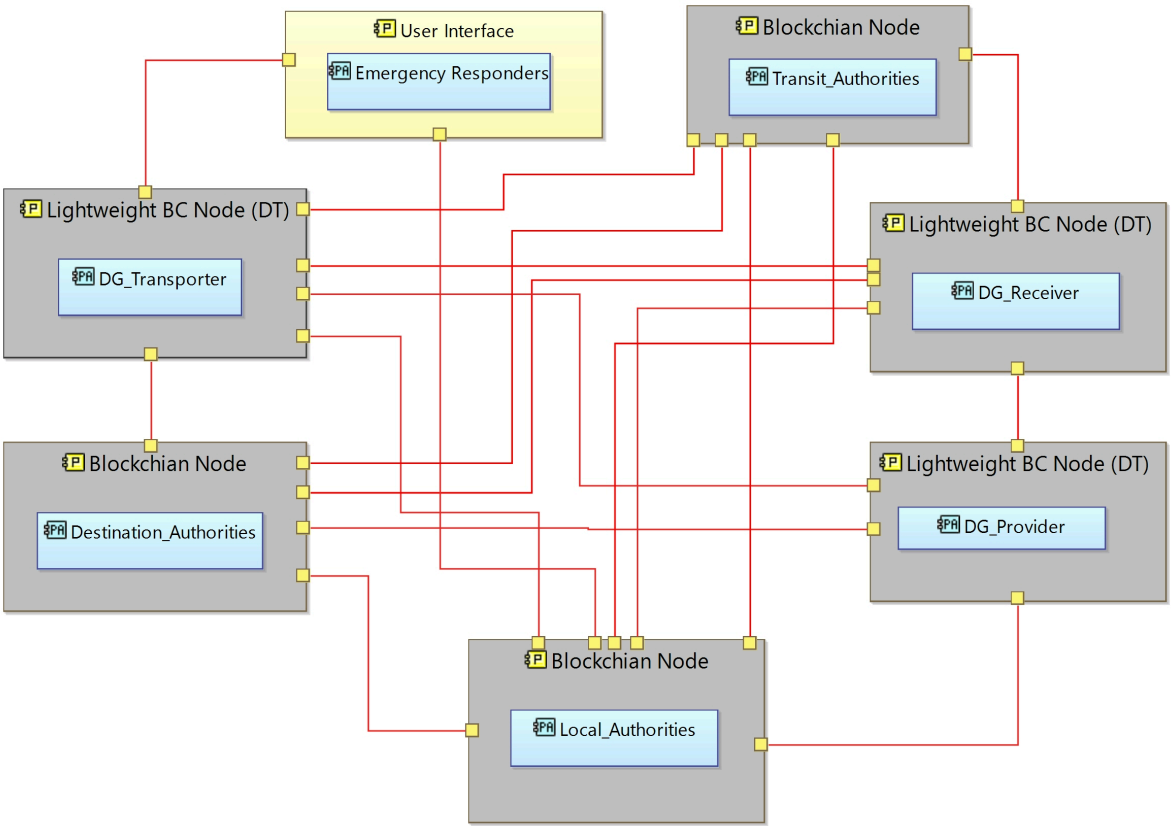

FIGURE 7.17: The deployment architecture for the independent system architecture for TDG.

forming the physical BC network. They share the same communication channel, i.e., BC technology, to share and validate transaction data. The "Lightweight BC Node" accesses the BC physical network through DT and performs authorized operations.

7.7 L5: Platform Specific Model (PSM)

This layer specifies the characteristics of the platform that will be used to apply our design method.

Among the main challenges when designing a BC-based solution is the selection of the appropriate BC platform. Various parameters need to consider before designing any BC-based solution. These parameters are related to BC technology performance or governance aspects (i.e., access rights, privacy, and confidentiality). For the TDG use case, we also consider these parameters. In the proposed design method, we aim to use IoT devices to capture data related to DG transportation and for transaction validation (this latter requires a certain performance level). Furthermore, for governance aspects of the stakeholders involved in the TDG, a certain level of information security and privacy is required to satisfy the "legal contract" or "business agreements".

To select the most appropriate BC framework, we have performed several analyses of existing BC frameworks (shown in 3.4.3) against the required properties for the TDG use case. As a result, we found out that public BCs such as Bitcoin or Ethereum are not appropriate solutions since privacy, confidentiality, and performance are significant issues of these platforms (Zyskind et al., 2018). In the TDG use case, it is required all stakeholders participating in the consortium need to be formally identified (certified identity). For any stakeholder that is requiring to be part of the consortium, initially, access should be required to the consortium and may be part of this consortium only if this access is granted. Granting access means the stakeholder can exchange transactions with other consortium members based on a defined access control policy. Authorities of each participating country may manage this consortium. The BC platform we have identified to satisfy the TDG use case requirements is the *"consortium blockchain"*. It allows forming a consortium of stakeholders with additional properties on privacy and confidentiality. As a result, Hyperledger Fabric (3.4.3) has been selected as the BC framework to develop our solution.

Figure 7.18 introduces the core components of the platform-specific model (PSM). The selected platform-specific model in our design method is Hyperledger Fabric (HF) as a BC platform.

In the PSM presented, the main component is the "Blockchain-Based System," representing the system core. This signifies a BC platform, and in the global context, it may be an instance of any "Blockchain Framework", for example, Hyperledger Fabric (HF) or Ethereum, or any other related technology that satisfies the requirements expressed at the design layer (L1 - L4). The "Blockchain-Based System" is hosted by many "Stakeholders" that are part of the process (as presented in 7.6.1). In such a model, the "Stakeholders" behave as physical BC nodes and are an essential part of the system since they will host and maintain the global ledger. The "Blockchain Framework" component determines the type of BC network, e.g.,

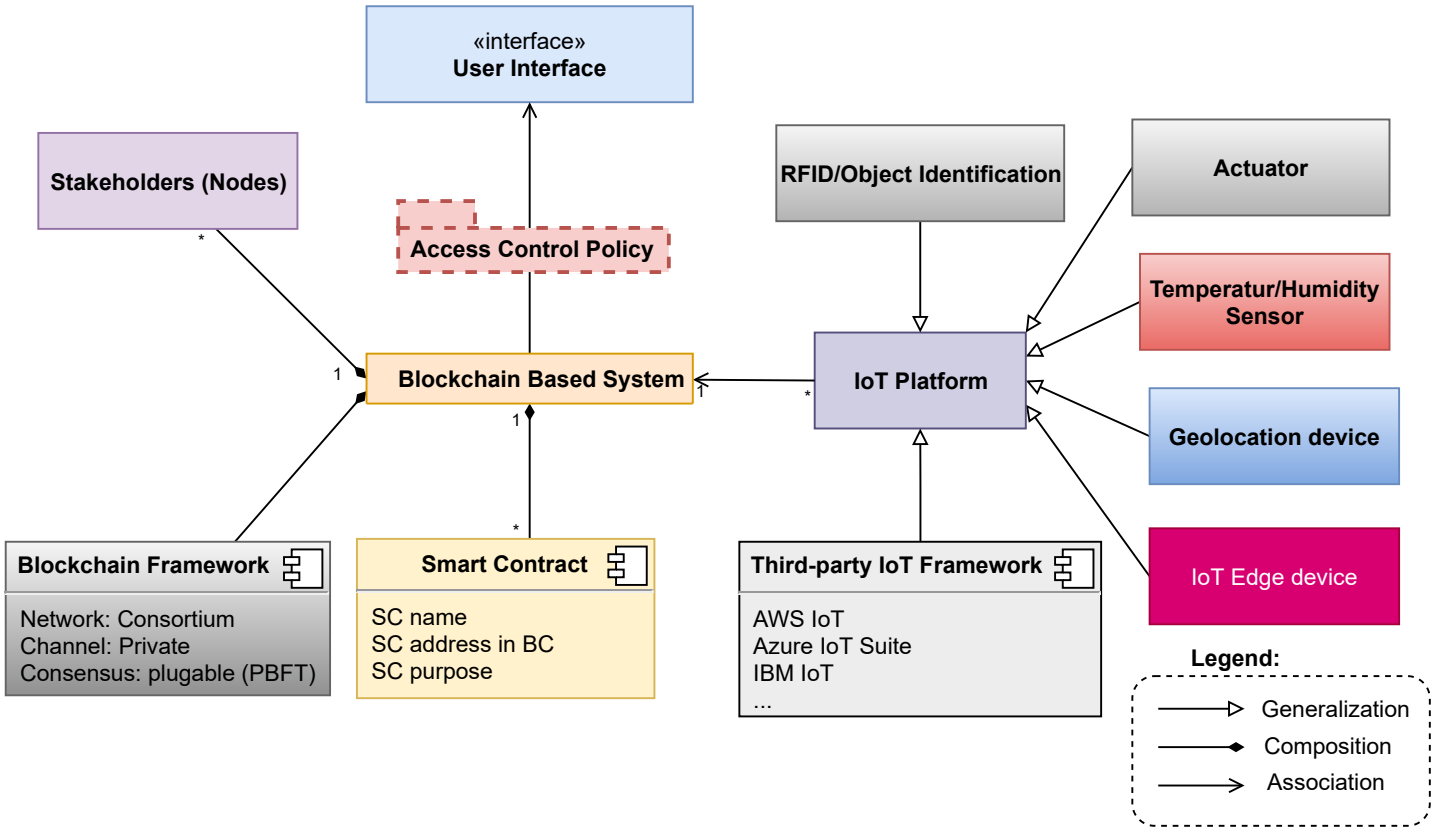

FIGURE 7.18: The platform-specific model and related components for the blockchain-based system.

"consortium", in such a case composed of "invited and certified" stakeholders, and for privacy issues, the notion of channels is used in the HF jargon. Regarding ledger data validation and maintenance, a specific consensus protocol might be plugged, or the system might rely on the built-in consensus mechanism offered from the BC platform.

The "User Interface" components enable the interaction of the "Stakeholders" involved in the process. They use the "User Interface" components to register on the system, apply for certification as an operator with DG, introduce their requests for authorization, register DG, receive information from the transport process, and receive specific information for a particular event in the TDG. The involved stakeholders are consolidated as virtual objects (digital twins (DT), as shown in Section 7.6) and can only interact with the system and other system components by using their DT.

The "Access Control Policy" component presents a software package that should be implemented to enrich the access control mechanism. Besides the authentication mechanism offered by BC and the privacy ensured by HF channels (3.4.3), an enhancement of the authorization aspects could be improved by defining an advanced access policy according to the privacy and confidentiality requirements. In this sense, we defined role-based access control based as detailed in Section 7.5.5.

The "Blockchain-Based System" is combined with the "IoT Platform" to provide real-time information for the events and activities in the process workflow. The "IoT Platform" may contain different IoT devices that have the specific purpose of collecting data at the

process run-time, including geolocation (i.e., warehouse). In our use case, a set of IoT devices are required to support and monitor the TDG process. These IoT devices are mainly used for object identification (RFID tags), geolocation, and the continuous surveillance of the movement of DG, e.g., GPS tracker. For changing the object state in specific circumstances, we use actuators. Further, some specific DG is sensitive to environmental parameters, i.e., higher temperatures or humidity; therefore, we use sensors to measure such environmental parameters. We use an IoT edge device (e.g., Raspberry Pi) as a BC lightweight node to collect information from IoT devices and send it to the BC full node. Another way of retrieving IoT data is by accessing the existing "Third-Party IoT Framework". This allows us to access IoT data that is hosted by IoT vendors such as SigFox, LoRa[12], etc. This is mainly possible by accessing their API that provides IoT data. We present a BC and IoT architecture and its related proof of concepts (PoC) for our design method in Section 8.4.

The "Smart Contract" (SC) component presents the business logic and functionalities required for the TDG, as expressed earlier in layer L3. For data processing, imposing constraints, and performing a particular task, the BC platform offers an SC solution. The SC component is one of the main components of the system. The SC technical capabilities and programming level depend on which BC framework is selected. For example, Hyperledger Fabric allows SC programming in different programming languages such as Go, JavaScrip, Java, and Python. Contrary to this, the Ethereum programming language for SC is Solidity (or Vyper) language only, a purely domain-specific programming language.

7.8 L6: Platform Specific Smart Contract Model (PSSCM)

This layer presents a specific code-model for the SC. The code-model generation is dependent on the selected BC platform. It should follow the specificity offered by the BC framework to fulfill the BC platform's characteristics. This layer is composed based on the previous layer. It follows a selection of the BC technology and its specificity to fulfill the previous layer's requirements. It also refers to the PISCM and PISA layers. The reference to the PISCM examines the inner interaction of prepared SC (pseudo code) with the system. Further, the reference to the PISA intends to know the impact of changes that occurred in PISCM and reflected in PISA.

For code generation, we define a code generation template that addresses our application domain, which allows converting the algorithmic expressions into code. We define transformation rules based on QVT (OMG-QVT, 2016; OMG, 2014), which enables code translation from PISCM. The pattern of transformation rules is given in Table 7.5. For code generation, we use the existing pseudo code from the algorithm composed in layer PISCM. We consider, ideally, an automatic SC code generation, but that requires additional tool development, which is beyond the scope of this thesis. An example of code generation for DG registration in the targeted system is showed in Appendix A.10.

[12] https://eu.mouser.com/applications/sigfox-lora-lte/

TABLE 7.5: The transformation rules from PISCM into code generation.

Source Model Components	Targeted Model Components
Template (Programming Language)	JavaScript
Dependencies for SC	Chaincode dependency; Node version; NPM
Initialization	Variable conversation
Pseudo-function	JavaScript function

7.9 Agility and Openness of our Design Method

We refer here to agility as a process of adapting and reflecting on new changes that occur in basic levels of our approach (L0 and L1) and their reflection in final level L6 without changing the entire system structure. As presented in our design method schema showed in Figure 6.1, the left-hand column is intended for supporting automatic processes and agile maintenance of future systems.

Our design method enables us to incorporate new changes in the regulatory framework (L0 and L1) into the code. These changes are validates through different layers. It manages changes and evolution, adding new concepts, impacting the final system components. These changes in our design method are managed in such a way that the main system is not disturbed, and these changes do not change the entire system. Changes in the law articles that are made in level L1 are reflected in L2 by presenting the new law artifacts. After verification (compatibility with the regulatory framework), the new artifacts are mapped into L3, from where they are directly recognized as new artifacts that need to be encoded. This flow does not intend to disturb the entire system architecture. It merely reflects specific changes, for example, law articles that need to be applied in certain aspects in TDG. Also, it allows us to perform reverse engineering, meaning that we can identify which code pairs reflect a particular law article (s).

We consider our approach open in the sense that it might be applied in different use cases and in different contexts while designing BC-based systems. The composition of the layers in our approach, which goes from the ontological level to the coding of the proposed system, gives the system designers the flexibility to use the same methodology to design the targeted system. Significantly, regulated domains might benefit from such a design method.

7.10 Tools Used for Developing our Design Method

The MDE offers a wide range of modeling tools. (Kahani et al., 2019) shows a detailed survey study on the tools that are used in MDE. Two different tools are used in the context of our approach, the AtomPM (Syriani et al., 2013) and Capella (Voirin, 2017). The AtomPM is used on layer L1 to design a "platform-independent meta-model". Capella is applied for mapping the interaction between the system and its components (in PISCM) (L3). To keep the semantic of the defined model at a certain point, we switched between these tools. In general, there are many tools used in MDE, and choosing the appropriate tool remains the preference of system designers.

7.11 Conclusion

This chapter presents the design method layers (L1 - L6). Initially, we present a model that enables the definition and mapping of the business rules into a meta-model. In the context of TMMW (TDG), business rules mainly emerge from the regulatory framework. The business rules are extracted once using human expertise. We use the notion of a meta-model (L1) for mapping these business rules. The business rules stand as the core of our design method since we use them to verify the process flow. Based on the knowledge extracted for defining business rules, we define the user model, i.e., BPMN (L2). With the BPMN definition, we intend to determine the entire process flow (end-to-end) as well as the requirement to fulfill an end-to-end process for TMMW (TDG). These requirements are expressed in terms of the interaction of the stakeholders and general process flow. We verify these interactions, i.e., the BPMN model based on the previously defined meta-model (L1). This verification enables us to design a regulatory framework-based system by further transforming the BPMN model into a UML Sequence Diagram (USD) (L3), and to enable the discovery of targeted system inner interaction and the expression of the business logic. Following this process, all these interactions are based on the regulatory framework based on the meta-model (L1).

Furthermore, in the Section 7.4.1 we present model transformation from previous layer showed in Section 7.3. We use this transformation to build our new design method components, showing the targeted system component interactions, and to examine the targeted system's inner behavior. We called this model a platform-independent smart contract model (L3) (PISCM), and it operates independently of any technology. Moreover, to enhance privacy and access control, we propose a general approach that allows stakeholders to determine their users' roles and deploy the access control policy. We further define the design and management of the access control policy conceptually with the help of the smart contract. Continuing in this way, we use the components defined in the L3 layer to establish the platform-independent system architecture (PISA). It presents the targeted system's digital components and stands as the main artifact for developing future software artifacts representing digital objects. In the sense of the deployment of the targeted system, we designed a deployment architecture to facilitate decisions on deploying and managing the future system. Further, we define a platform-specific model (PIM) (L5) to determine the technology chosen for developing the artifacts designed in the previous layers. This allows us to justify the technology choice and the choice of the other components used. Finally, on the last layer, a smart contract technology model (L6) is presented. This layer presents a template for developing a smart contract code based on the previous layers. Our method maintains agility over the new changes in the requirements, which determine the targeted system by using a specific layer on the design method to reflect the new requirements (changes) in the targeted system.

Chapter 8

Smart Contract Improvement in Terms of Semantics to Capture Specification of Supply Chain Management for Dangerous Goods

8.1 Introduction

In this chapter, we present some advanced concepts in the SCM of DG supported with the help of SC. We specify aspects for obtaining the semantics and improving the process of TDG. The semantics captured concerning TDG are implicitly associated with the SC. That, in a sense, is how SC can support such concepts for a reliable TDG process. These concepts aim at improving the process of TDG from the security and safety perspective, management of the process, and governance aspects of the process with the regulatory framework. The proposed concepts are designed to be useful in other use cases beyond the TDG (other TDG use cases or similar application domains), in combination with BC-based systems and SC. The presented concepts are introduced in terms of constraints and advanced TDG control system features. We list the constraints, such as obligations from meta-rules sourced in the regulatory framework, time-related and geographic localization constraints. We present the conceptual design approach composed of 3-layer architecture for integrating BC and IoT. Furthermore, we consider advanced features in the TDG, such as digital certificates, managing emergencies, anomaly detection, and shared responsibility, with the help of SC. The proposed concepts are formulated in association with SC functionalities, and the conceptual formalization presents mathematical-logic expressions mainly by using the first-order logic (A.8.1). We have identified a potential problem in managing the SC's immutability in a dynamic environment, thus proposing a conceptual solution to handle such problems.

Another objective of this chapter is to formally illustrate the semantics of SC in terms of formal specification of business rules for the TDG. The formal specification of SC allows expression of the SC properties, while the formal verification (model checking or theorem providing) verifies if the SC model behaves according to that specification.

8.2 Specification of Smart Contract to Capture the Semantics of SCM for DG

The SC specifications aspects are elaborated in terms of constraints that enforce the system to act according to the specifications derived from the regulatory framework, business requirements, and process management.

8.2.1 Meta-Rules from Regulatory Framework

We present essential concepts that are considered immensely important for designing a system that relies on the regulatory framework, including parts of business contracts. To design a compliant system with the regulatory framework, in addition to applying meta-rules on the meta-model (7.2), we also define the source of the constraint from the regulatory framework needs to be validated by the system through the specification of the SC. Essentially, the main challenge is *"How to adapt SC to react in line with the regulatory framework?"*. We have already determined the regulatory framework as a source for the SC in Section (7.2), particularly in Table (7.1).

To respond to the question above, we extract some meaningful legal parameters. The extracted parameters are further defined as meta-rules. These meta-rules have the form of *legal statements* that should be further mapped into SC (or functions of the SC that return values (Boolean) based on conditions). These meta-rules will be used throughout the process of TDG, depending on the need for applying any legal-rule verification in the TDG process. The example of **meta-rules** (MR) are defined as below:

- *MR 1: Stakeholder should not operate in more than θ class of DG;*

  By law, there may be a restriction on operating with DG. For example, a particular stakeholder, e.g., "Transporter", is not allowed to operate with different DG classes due to DG's risk in contact with other materials and the physical capacity to maintain these DG correctly. Therefore,

  If $C = \{c_1, c_2, ..., c_9\}$, and $S = \{s_1, s_2, ..., s_n\}$ presents the set of DG classes, respectively stakeholders. Then,

$$C \cap S = \{\{s_i, c_i\} \mid \textbf{card}\,\{C\} = \theta \wedge \textbf{card}\,\{S\} = 1\} \tag{8.1}$$

  The expression in (8.1), allows determination of which stakeholders is allowed to operate in θ[1] different classes of DG, and not more than θ. Further, it might be extended according to stakeholders' needs and constantly relying on the regulatory framework. In (8.1), we may impose more rigorous constraint, in the level of DG, by specifying DG that might be managed (governed) only by a specific set of stakeholders[2] or only the stakeholders that can transport and manage these DG. Therefore,

[1] θ - number of DG classes, and $\theta \in \{1, 2, ..., 9\}$. The θ determines the number of DG classes for which a stakeholder can operate. Usually, it is determined by the competent authorities of countries based on their regulatory framework for TMDG.

[2] The decision to determine the stakeholders based on the DG classes belonging to each country.

$$C \cap S = \{\{s_i, c_1\} \mid s_i \in \{s_1, s_2, s_3, ...\}\} \wedge C \cap S \neq \emptyset \tag{8.2}$$

The definition in (8.2) determines i) DG belonging, for example, to class c_1, the responsible stakeholder in terms of competent authorities, and ii) the list of stakeholders that are possibly qualified for the transport and management of that particular DG, e.g., radioactive material. This definition (8.2) is based on the regulatory framework that defines which authority, e.g., Ministry in a particular country, is responsible for specific DG (c_i), as we have described in the introduction of DG (2).

- *MR 2: Drivers should be certified;*

  For the transport of DG, a trained crew is required to manage DG, especially during transport. Lets be $D = \{dr_1, dr_2, ..., dr_n\}$ set of drivers. Thus, $D \subseteq S$. The driver is allowed to drive the vehicle carrying DG if and only if it is certified by the competent authority in the local country.

$$(\forall dr \in D)\ (\exists\ certificate_{authority}(D)) \tag{8.3}$$

- *MR 3: Drivers are not allowed to drive more than π hours;*

  For safety reasons, when transport DG, there should be limited time the driver is allowed to drive. Such a condition may originate from the regulatory framework or business contracts. Let T presents the total time, where

$$(\forall \pi \in T)\ \ (\exists\ divingTime_T(D)) \wedge (divingTime_T(D) \leqslant \pi) \tag{8.4}$$

  The definition in 8.4, also determines that the driving time may be determined based on the DG that is subject to transport.

- *MR 4: Specific DG is not permitted to be transported into a particular country;*

  This highlights the geographic constraints, that might be imposed by specific authorities. In the following sections we will presents a concept for geographic localization constraints.

- *MR 5: Impossible to transport and manage DG without authorization;*

  This meta-rule determines the need for authorization for any stakeholders that intend to hold, transport and manage DG.

$$(\forall\ TMDG_{activity}) \implies authorization_{required} \tag{8.5}$$

- *MR 6: All stakeholders should be certified;*

  In the line with previous rule (8.6), any stakeholder that intends to transport and manage DG should be certified from the competent local authority. Therefore,

$$(\forall\ TMDG_{activity} \wedge s \in S) \implies (s_{certified} \wedge authorization_{required}) \tag{8.6}$$

These meta-rules are applied across SC as support for the reliable system. For example, (8.1 and 8.2) are used by the SC function when stakeholders intend to transport DG in a specific country. The involved stakeholders, particularly competent authorities at the international level, might verify the necessary information about drivers by using any SC function that validates conditions showed in (8.3). The statement presented in (8.4) may be used in a different context, i.e., in an early estimation of the needed transport time.

The following section presents time-related constraints, which play a significant role in the TDG.

8.3 Time Constraint Smart Contract

The concept of time represents one of the main perspectives to consider in the process management related to the SCM in general. Many SCM processes are validated by time, such as a measurement to complete specific tasks or activities. Likewise, in the TDG, time plays a significant role in managing and organizing processes. It serves as an indicator to have an efficient and reliable process, affecting the safety and security aspects in TDG. Time may be used as a constraint on the organization and measuring the quality of the process.

In terms of TDG, we present two time concepts: "strong time constraint" and "weak time constraint", which are further applied to SC.

Definition 1. ***SC Strong-time constraints for process P***

A global process $\mathbf{P}^G$ that is composed of several sub-processes (or tasks) p_i , executed in sequential workflows, and expressed in the SC in terms of functions, $F_n = \{f_1, f_2, f_3, ..., f_n\}$, is considered valid if and only if the total time T required to compose the process (in an end-to-end manner), is lower or equal to the predefined time variable H.

Therefore,

$$P^G_{Time} = T \leq H \tag{8.7}$$

In the formal representation of the equation (8.7), we have:

Process $P^G \triangleq (p_0, p_1, p_2, ..., p_n)$, where p_i are sub-processes in the TDG, jointly form the end-to-end process (P^G).

Time T is composed of partitions $t_0, t_1, t_2, ..., t_n$. T presents the total time $T = \sum_i^n t_i$ needed for the processes P^G. Any sub-process $p_i \in P^G$ is associated with time partition $t_i \in T$.

The variable H is defined according to the maximal time required for the end-to-end process. For example, when planning the transport, one should calculate the time for DG preparation, DG loading, and DG transportation. In the case of international transport, calculate the cross-border time and DG unloading. Therefore,

$$H = max\_time(P^G) \tag{8.8}$$

The definition (8.7) has two major impacts in the TDG system: (*i*) at the design level and (*ii*) at the run time level.

The design level indicates that during the design of the TDG process, time should be considered as one of the system's primary properties. Thus, while designing such a system, a designer that maps the stakeholder interaction should consider the time constraint to avoid time management issues. To manage the time issues at the design level, we consider the constraint added on the FED schema presented in Section (7.5.4).

At the run-time level, time impacts service quality and potentially violates the Service Level Agreement (SLA), where stakeholder requirements may not be fulfilled. Further potential issues may arise at the compliance management level, where the regulatory framework terms (from business or legal aspects) may not be fulfilled. For some TDG, the time frame is crucial. For safety reasons, there may be an imposition on the transport time-frame. For example, specific DG, such as "infectious waste" or "radioactive material," may be required to be transported starting late at night and arriving early in the morning to avoid rush hour traffic.

Definition 2. ***SC Weak-time constraints for process P. The impact of tolerance parameter***

A global process $\boldsymbol{P}^G$ that is composed from several sub-processes p_i, executed in sequential workflows, and expressed in the SC, in terms of functions, $F_n = \{f_1, f_2, f_3, ..., f_n\}$, is considered valid if and only if the total time T ($T = \sum_i^n t_i$) required to compose the process (in an end-to-end manner) is lower or equal to the predefined time variable H + α.

Therefore,

$$P^G_{Time} = T \leq H + \alpha, \text{ where} \tag{8.9}$$

α - is tolerance parameter:

$\alpha \approx 0$ - if and only if no delay,

$\alpha > h$ - where h is a specific value that triggers a notification event (emergency alert).

In the TDG, unpredictable situations can happen during the transport process, such as random path deviation (in case of accidents in the road), traffic congestion, or truck failure during transport (wheels or other problems). For tolerating a level flexibility in managing such a situation, we introduce a tolerance parameter α. This parameter has the role that arbitrarily collects the "delay time" in a specific case. It plays a role in maintaining the process of TDG as a way to collect and calculate delay time, and to perform further action based on that calculation.

The calculation of α is performed continuously. We propose at least two ways to calculate α. The first method of calculation we propose is by using IoT devices, which send real-time information during the transportation process. For example, if there are *"p minutes"* delay in the cross-border (or traffic congestion), then this delay is associated with the α. Therefore,

$$\alpha \leftarrow p$$

The value of α that is continuously updated with p value received from IoT devices is tolerated until some thresholds are crossed. In a specific case, the value of α is tolerated for h hours. Therefore,

$$[0 \leqslant \alpha \leqslant h]$$

The value of h is an indicator of another *action* in the transport process, indicating safety issues and triggering an alert situation.

$$if\ \alpha \notin [0 \leqslant \alpha \leqslant h]$$
$$Stakeholders \leftarrow Action\ Required$$

In such a situation, if the α is not on the given interval, meaning that the interval parameter is violated ($\alpha > h$), then this clearly indicates it is not possible to reach the destination in the predefined time. This scenario covers the situation where several countries (cross-border) allow DG (specific DG) to enter their territory only within the predefined time-frame. These are situations where the border is closed at a certain time. Beyond this time-frame, entering with DG is not possible and the situation may be complicated since the DG loaded on the truck may present safety and security issues. Using the α parameters, we intend to provide additional information for the stakeholders, thus informing them before the issues arise to allow them to take the necessary pre-cautions and reorganize the TDG to avoid this situation.

Analogy of applying α in other domains. An example from the construction domain

In the construction domain, when constructing buildings, usually a company is contracted by the project manager to complete the work, i.e., construct the building. To complete the work, there is a certain period of "time", i.e., deadline (corresponding to our H). The end-to-end delay (deadline) is known, and that is one (1) year contracted ($h = 365\ days$). The process of building has several phases (corresponding to our p_i). Some of the processes have to occur in a specific order (sequentially), e.g., ground excavation, building the structure, until completing the building exterior. Once the building template is completed, several processes, e.g., plumbing, electricity, and many others, can run in parallel. We recall that the end-to-end has to be finished in contractual time (h). In the case of inadequate weather conditions, for example, if there is heavy rain or a storm which prevents normal workflow, the company will not work during these days. So there is a "justified delay" (corresponding with our α, which in the contractual time is equal to zero) according to the law (regulatory framework) for construction, the law articles of unexpected events. That means the company will not pay any penalty for not finishing work on contractual time. All the events impact the value of α (increasing the number of days). Consequently, the contractual time H will increase for a value of α, therefore have *"end-to-end construction time"* = $H + \alpha$. Furthermore, in the end, the core issues remain if the α is **reasonably justified**. There is the question of whether the company should or should not pay the penalty. For measuring such parameters, there are competent "authorities" that make a decision on the delay (α), for example, if rainy (storm) days, $\alpha = 30\ days$. Depending on the decision of the authority, the company will (or will not) pay the penalty.

The value of α in the BC

Similar to the authorities that calculate the value of α in the construction domain, we consider BC as such an "authority" in the context of TDG. Instead of having calculation authority

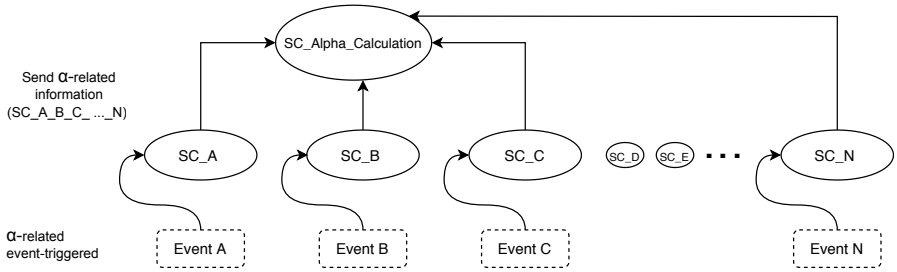

FIGURE 8.1: The SC schema for updating the value of alpha (α) based on different events captured with the help of SC.

to decide such issues, we place a distributed-decentralized consensus-ruled mechanism to perform such a decision. The value of α will continuously be updated based on the event observed in TDG. The events that impact α are pre-defined by the consortium of stakeholders. The calculation of α remains unchanged and transparent for all members of the consortium.

We specify SC (*SC_alpha_calculation*) to update and calculate the α continuously, based on the value received from IoT devices (in the case of transport) and other SC that maintain different events related to the α dependencies. The other SC maintains events such as border crossings, traffic congestion (or pre-defined roadwork), and severe weather conditions that are official events which might impact the TDG. Figure 8.1 shows the association of SCs with the *SC_alpha_calculation*, where for any event captured in relation to α, the *SC_alpha_calculation* is invoked by sending α-related information.

8.3.1 Applying Time Constraints on the Blockchain and Smart Contract Level

For the possible application of time constraints in such a scenario, the current BC protocol is considerably limited. In general, the concept of time, and moreover the global clock, is not defined (does not have any exact meaning) on the BC, and this concern remains for major BC frameworks such as Ethereum and Bitcoin. Because of the BC decentralized principles, there is no central point to store and manage the global clock (time). This means that it is not possible to schedule transactions to execute in the future[3]. Furthermore, querying the time for the specific transaction yields the miner clock timestamp at the time of mining. Moreover, at the Ethereum level, the SC execution (calling) requires a private key engagement, thus excluding scheduling SC invocation by a time-based transaction. SC does not have any private key. Therefore they need to be triggered by a private key based account. Nevertheless, to support and better manage the time constraint in BC, we propose some design principles for managing the BC time constraints.

8.3.1.1 Ethereum-based Design

The design principles of Ethereum offer a built-in function called *now* that gets the "current time" when the SC function is executed. With the process initialization, we use the approach

[3]For additional information, see Ethereum Alarm Clock's documentation, available in the following link: https://ethereum-alarm-clock.readthedocs.io/en/latest/

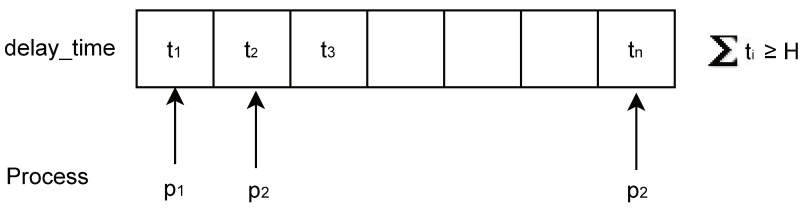

FIGURE 8.2: The conceptual representation of the time-constrain in SC.

shown in (Mutunda, 2017), which is based on the *now* function to get a starting point for the time counter in our process. Furthermore for each SC function that is executed, we associate a variable *delay_time* (t_i) at the end of function in order to capture the time for the process p_i, therefore ($t_i \leftarrow p_i$). Thus at the end of the process, we verify if the sum of t_i (counting hours) is higher than H, as shown in Figure 8.2.

Relying on the miner timestamp is not among the best design principles. The timestamp is not immutable at the miner level, and the host can manipulate it (see SC vulnerabilities in (Kolluri et al., 2018)). Whereas deploying a consortium network, in which nodes are certified and trusted (authenticated and authorized) is more reliable, this design principle may be acceptable and more easily manageable.

Compared to the current miner-based timestamp, we consider other design principles to manage time in BC with SC's help. In the following section, we present the concept of a shared "time-based" variable.

8.3.1.2 Shared Time-based Variable for Time Management and Constraint

The time-constraint should be applied at the design time and run-time level, as mentioned earlier. We propose a novel formal approach for managing time constraints in TDG in design time by introducing it earlier in the FED schema and following at the run-time level.

We propose a conceptual approach for managing time constraints in TDG at the run-time level. The process in TDG is executed in an orderly way (sequentially). The process p_n can be finished only after p_{n-1} as presented in (8.10). That allows us to manage the way processes are executed and to maintain the execution flow. Therefore,

$$p_n \leftarrow p_{n-1} \leftarrow p_{n-2} \leftarrow ... \leftarrow p_2 \leftarrow p_1 \tag{8.10}$$

For managing the time constraints, we propose the concept of a shared variable that stores the time (t_i) for each process (p_i). The variable is managed by a specific SC, named *Time_Constraint_SC*, which is invoked by other SCs (or functions of the SCs), based on process flow, as shown in Figure (8.3). Our conceptual approach is composed of the following steps listed below, and the algorithmic representation is shown in (5):

1. Deploy SC (*Time_Constraint_SC*) to manage the shared variable. This SC stores and calculates time, and notifies stakeholders in case of time-constraint violations. Besides the main functionalities specified for this SC, the deployment phase allows specification of different parameters for managing run-time execution of the TDG process. Among

the main parameters is the threshold (*Out_of_time*) used to indicate the unexpected delay time of a particular process (p_i).

2. Time initialization. When the process (p_i) starts, the time (t_i) is initialized. That is mainly achieved by using the function *now*, or any other programming primitives that yield time.

3. An SC (or function of the SC) that is related to the process (p_i) of TDG and its objective is to calculate the time for this given process, initiated by getting the time at the start of the process and at the end of the process. An SC function (or local variable) is used to get the time and send it to the *Time_Constraint_SC*.

4. An SC, e.g., SC_X (or SC function F_n_X), related to TDG which has to count the time, is used to manage the time. The events that happen in SC (or function of SC) indicates the time. Time management is performed in such a way that, when the process starts, the time is captured by a variable (*start_time* = *now*). A transaction is immediately executed which sends the *start_time* to *Time_Constraint_SC*. The *Time_Constraint_SC* receives the transaction and stores it (including the timestamp received from the same transaction) in the BC. Then the process in SC_X continues in *"work to do mode"* until it finishes. When the process is completed, another variable *end_time* = *now* indicates that. Furthermore, it sends a transaction to the *Time_Constraint_SC*. The *Time_Constraint_SC* receives the transaction and stores it (including *end_time* = *now*, the timestamp received from the same transaction).

5. The *Time_Constraint_SC* reacts according to the instruction given in the deployment phase. An SC function in *Time_Constraint_SC* initially stores the received transaction (from SC_X) with its unique parameters. It stores again (in the second received transaction) the information received from (SC_X) with its unique parameters. From the stored transaction data, it performs the time-constraint calculation. In case the SC_X does not send back the *end_time* = *now*, transaction, *Time_Constraint_SC*, will inform the relevant stakeholder that the given process (p_i) is never achieved, thus the threshold *Out_of_time* is reached.

6. Protecting the shared variable (*Time_Constraint_SC*).

   The *Time_Constraint_SC* plays the role of a shared variable from the point of view of other SCs. Since the *Time_Constraint_SC* is deployed in a decentralized environment, any user can invoke *Time_Constraint_SC* at any time, which raises the risk of not maintaining the values correctly.

   To avoid such scenarios, which may lead to inconsistencies, we propose an approach for protecting *Time_Constraint_SC*. Whenever the *Time_Constraint_SC* is invoked, set values are stored on the BC for that invocation. This set of values mainly presents the *start_time* whose purpose is to store it, the *SC_ID* which indicates the SC from which it was invoked, particularly the transaction sender, e.g., the IDd of SC_X, *caller_ID*, the owner of the SC (SC_X). Furthermore, it stores the details of a process p_i, a short

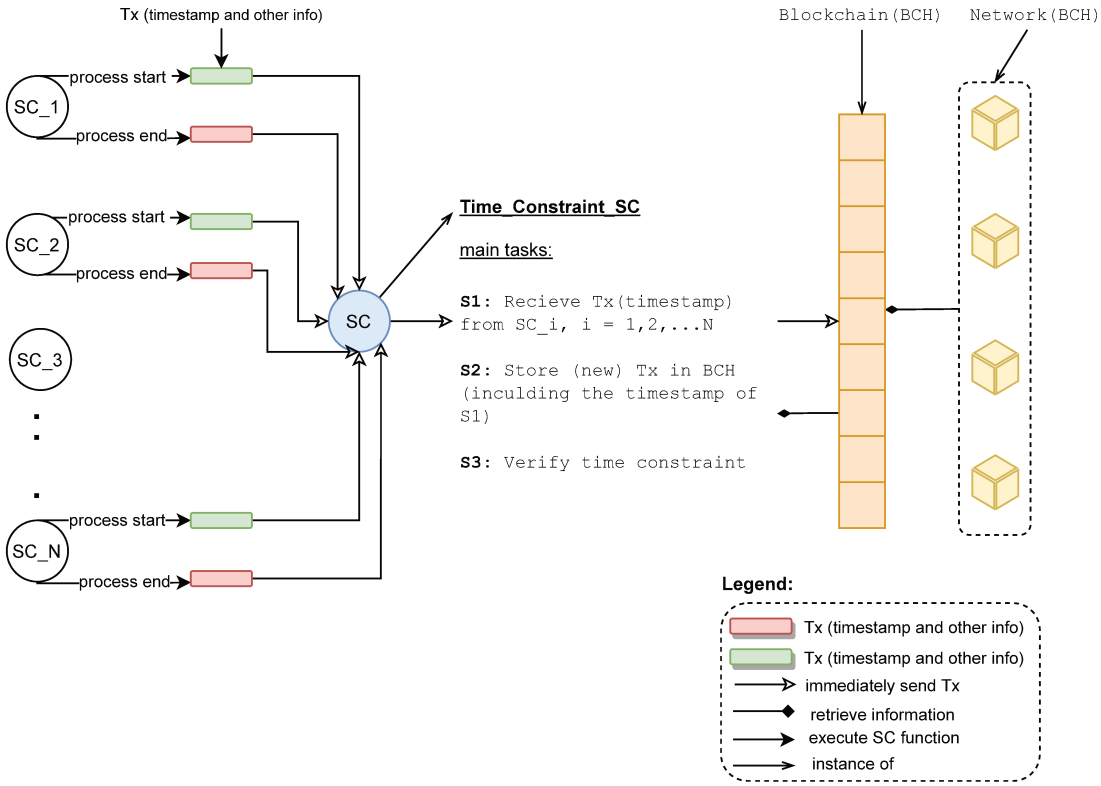

FIGURE 8.3: The concept of SC for time management at run time level. The conceptual schema propose the time management by using on-chain blockchain capabilities.

description, and auxiliary data needed for authentication of the data records. When the response time is requested from *Time_Constraint_SC*, it gives a response based on such parameters that are stored uniquely for a process p_i. The stored data serves as a barrier to avoid any overwriting of such parameters. Comparably to the shared variable from classical programming primitives, it is diverse because it can be invoked only by a process in the given time. This is because all the nodes in the BC should be aware of the ledger changes and agree on them (in general, the consensus rules allow a single change to be proposed at the time).

Algorithm 5: Algorithmic representation of time constraint and management for TDG.

```
1  Initialization: deploy an SC that stores and manages time constraints.
2  Time_threshold_p_i // time defined by stakeholder as threshold for specific process in TDG;
3  Counted_time // presents the total time needed for the process p_i to be completed;
4  Out_of_time // presents a specific threshold for forcing the end of specific process p_i;
5  while p_i > 0 do
6      //Any SC can count time and send it to Time_Constraint_SC;
7      SC_X do;
8      send tx_start (p_i, short description, start_time = now, SC_ID (SC_X), caller_ID,
         auxiliary data) to the Time_Constraint_SC;
9      Time_Constraint_SC ← tx_start;
10     //store transaction in BC
11     SC_X do;
12     send tx_end (p_i, short description, end_time = now, sc_ID (SC_X), caller_ID, auxiliary
         data) to the Time_Constraint_SC;
13     Time_Constraint_SC ← tx_end;
14     //store transaction in BC
15     Time-constratint calculation from Time_Constraint_SC;
16     Counted_time (p_i) ← Time_Constraint_SC (count(tx_end - tx_start));
17     if (Counted_time > Time_threshold(p_i) or Counted_time < Out_of_time) then
18         Notify stakeholders for time constraint violation;
19     else
20         Continue to work normally
21     end
22 end
```

The proposed approach does not rely directly on the transaction timestamps nor programming primitives. We propose this solution as an on-chain capacity to maintain time for any process. Formally, we introduce some additional constraint mechanisms to verify the timestamp of the transaction execution and avoid relying entirely on the miner. The *Time_Constraint_SC* writes (new) transactions upon receiving any invocation from another SC. We recall the fact that the *Time_Constraint_SC* is distributed over many nodes on the BC; thus, it bases its timestamp of writing (new) transaction on their local clock (the miner, where the *Time_Constraint_SC* is actually executed). That clearly indicates that the *Time_Constraint_SC* is not a centralized mechanism. When writing the (new) transaction, it also includes the timestamp (*start_time* and *end_time* respectively). That allows for this SC to provide a comparative mechanism over the timestamp received (expressed in *start_time* and *end_time*), and the timestamp the *Time_Constraint_SC* writes the transaction on the ledger. For example, when receiving the transaction, among other information, it also stores the following:

$$\begin{aligned} SC\_X\_rec\_tx\_timestamp\_start &= p; \\ SC\_X\_rec\_tx\_timestamp\_end &= p + Counted\_time; \end{aligned} \tag{8.11}$$

where *Counted_time* is shown in line 3 (and 16), in Algorithm (5).

These are the parameters (8.11) that are compared with the timestamp of writing the transaction from the *Time_Constraint_SC*, with a small degree of standard deviation (network latency or transaction delay tolerance). If high differences are noticed, then the transactions will be rejected, and all the involved stakeholders should be notified. The transaction refusal means that further process will be terminated, and further procedures may be engaged. This approach helps to avoid any fraudulent behavior from the miners.

8.3.1.2.1 Evaluation of the Proposed Approach SC deployment in a public BC such as Ethereum is costly and is not the most appropriate solution when many SC are proposed. We have explored the cost of deploying the SC in Ethereum and its execution in Section (3.4.1.1), where we performed BC technology studies. The results indicated that any time a transaction is executed, an amount of wai (gas) is required to proceed with the transaction. The cost may be high and not reasonable in multiple transactions to maintain such a solution.

The proposed approach is more suitable for the consortium BC, where the execution cost of the SC and network maintenance is not based on the miner incentives. As we have presented in the architecture Section (7.6.1), the deployment schema for the TDG system control is based on authorities and the involved stakeholders which maintain the network. That shows that the miners are not rewarded for any transaction execution, and involved stakeholders ensure the network maintenance.

8.4 Blockchain and IoT Integrated Approach for a Trusted and Secured TDG Process

This section presents the conceptual approach for BC and IoT integration for a trusted and secured TDG process (Imeri et al., 2020b).

8.4.1 3-Layers Conceptual Architecture

The approach we propose a smart, secured, and trusted process of TDG aims to adapt the process of managing the TDG based on available information about security, integrity, and availability, i.e., accessibility. We propose a new approach to exchange (share), manage, and store information between stakeholders using BC technology. The core advancement behind our proposed solution is the decentralization mechanism, supported by a combination of BC technology and IoT devices. The proposed solution aims to respond efficiently to any security concerns in the TDG as presented in section 5.5.2.1. Figure 8.4 shows our conceptual architecture. The proposed conceptual architecture is composed of three layers, organized in a top-down manner as follows:

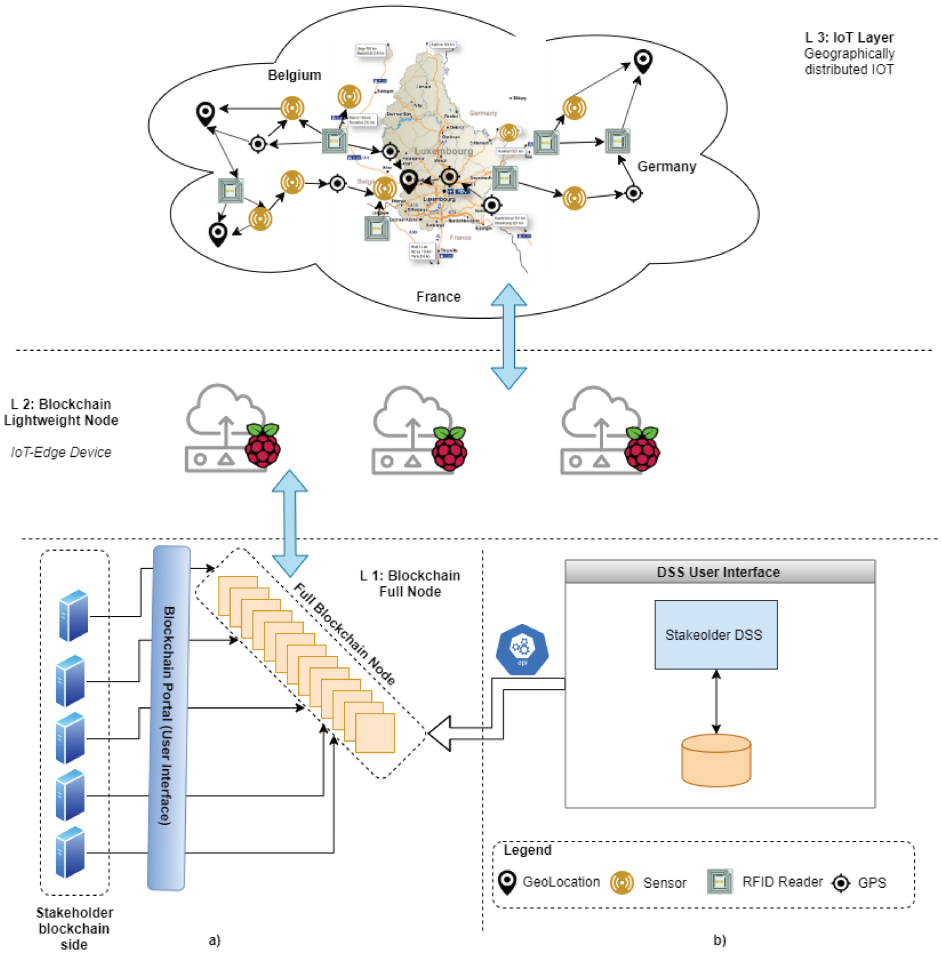

FIGURE 8.4: The conceptual architecture for blockchain and IoT integration to support the TDG.

- *L3: IoT devices layer*

  This layer is composed of IoT devices that are deployed in the target geographic area (e.g., for object identification during mobility) and embark in the transport facilities (e.g., sensors and GPS trackers into the trucks, etc.). These IoT devices are not intended necessarily to have powerful processing capabilities

  All these IoT devices are identified before deployment (using the CA authority of Hyperledger Fabric (HF), which releases public key for IoT devices) and authenticated. These IoT devices are registered on the BC (L1) using their hardware identification. This registration allows storing the IoT devices *public key* on the BC, for their identification (Huh et al., 2017). This allows to avoid the BC to receive information from an unauthorized IoT device and therefore securing it from potential attacks.

  Regarding the communication protocol between IoT devices, we recommend using peer-to-peer communication. Indeed, while this is optional, it might be chosen by the system designers to improve the performances of the system (e.g., extending communication

range using P2P IoT communication protocol)[4].

- *L2: BC Lightweight Node*

  The second layer comprises IoT edge devices with higher capabilities to processing data (transactions) captured by other IoT devices (hosted at L1). These devices have the necessary storage, processing power capacity, and operating system to perform authentication and signing (confirm) transactions. These IoT edge devices are known as BC *"lightweight node"*[5], which means that they do not contain the full BC stack. Their primary task is to sign transactions (confirmations) using the BC mechanism. When the *"lightweight node"* receives a transaction, it first checks if that transaction is from a registered IoT device, and further, it signs the transaction. After signing (verifying) the transaction, the *"lightweight node"*, use the appropriate communication channel to send the transactions to the full BC node (L1). Furthermore, the IoT edge devices serve as transaction (data) aggregators (collector) when the connection with the full BC node is missing. The received transactions from the IoT device (L1) are stored and kept until the connections with the full BC node are retrieved, and then these transactions are transmitted to the full BC node.

  The communication channel that we propose is a connection over a cellular network 2G/3G/4G provided by a mobile operator or over a wireless WAN (Wide Area Network) provided by a LPWAN (Low Power WAN) providers, e.g., SigFox, LoRA, etc., to properly transfer transaction data from L2 layer to L1 layer and vice versa. Since the communication channel could be subject to security issues in this segment of the network (Voas, 2016), we propose to always encrypt the transferred transactions.

- L1: *Stakeholder BC side*

  This layer belongs to the stakeholder's domain. It comprises several BC nodes deployed in different stakeholder premises, thus creating a geographically distributed networked system. These nodes have the capabilities to add new blocks into the BC. These transactions are received from the previous layer (L2) and transactions from other stakeholders. The business logic required for TDG is implemented in this layer with the help of *Smart Contract (SC)*. The SC that is intended to express the workflow of the process of TDG is deployed on this layer to fulfill the TDG business logic. Furthermore, all other components such as IoT devices (L3) and "lightweight nodes" (L2) are registered on this layer. When a block is added on the BC, the corresponding SC is executed to trigger the specified tasks in the business model in conformance with its logic.

  This layer also serves as the user interface for the stakeholders. The ones involved in the process might use the API provided by layer to insert immutable information (using the *BC portal*) and share them with other authorized stakeholders (as presented on the

[4]There are several communication protocols for IoT devices such as ZigBee, WiFi, Bluetooth, etc.
[5]The definition of the "Full Node" and "Lightweight Node": https://www.mycryptopedia.com/full-node-lightweight-node/

left side of the Figure 8.4, (L1 (a))), exchange information with other stakeholders and monitor the lifecycle of the TDG process (Imeri and Khadraoui, 2018).

The proposed approach permits also to existing business applications such as DSS, or other ERP systems, to connect to the BC (as full BC nodes) using specific API, as presented on the right side 8.4, (L1 (b)) (Imeri et al., 2019a).

This proposed approach aims to provide a new way of managing, storing, and sharing information in the process of TDG. It allows stakeholders to connect their applications to the system while eliminating the need to use third-party or centralized systems (e.g., clouds or centralized databases), as shown in Figure 3.3. Using BC technology for storing and managing the information brings to the system the required level of confidentiality. As a matter of fact, only the certified parties are enabled to perform actions in the SpC of DG. Furthermore, this system enables the authentication and authorization of any stakeholder accessing the system. All the users are authenticated and a full authorization control is performed on any of their actions. For example, a driver is only allowed to load DG if he is successfully authenticated by the host, e.g., "DG Provider", and at the same time the location of the driver should be one of the "DG Provider" premises. In case of violation, a notification alarm for non-compliance is sent to "authorities" and "DG Receiver".

In such a system the information remains immutable, thanks to the BC properties. The nodes hosting the main ledger are fully decentralized, and the system remains sustainable since none of the end-users (stakeholders) is able to shut-down the whole system.

The information sensed by IoT devices provides real-time tractability information about the actual state of the TDG process. The user interface allows authorized stakeholders to monitor the process and securely store their data in the system. The ability of immutable record-keeping of BC enables auditing of processes and operations for TDG.

For evaluation of the BC and IoT integration approach, we have implemented a proof of concepts (PoC), showed in Section 9.2.9, also extensively described in showed in (Imeri et al., 2020b; Imeri and Lamont, 2019).

8.5 Geographic Constraints

It is widely known that DG exposes immense risk for humans, living organisms, environments, and properties. To avoid an adverse situation with DG, many countries impose regulations to forbid DG movement through populated areas. In this sense, we discover the need for a geographic location constraint that imposes rules over the movement of DG in specific areas. The purpose of the geographic constraint is to maintain the movement of DG in regulated transport geographic space, thus avoiding any violation of local or international rules. Furthermore, it shows a level of adaptation with new regulations emerging with the change in geographic areas, i.e., the city's geographic layout. We define the geographic localization constraints as follows:

1. *Limited geographic area where truck should not deviate from given path(s).*

These are situations where the given area is populated and any other path have higher risk (calculated in advance by stakeholders) in the case of accidents involving DG.

We have that,

$$\exists\, \rho \;\wedge\; \{\phi_1, \phi_2, \phi_3, ..., \phi_n\} \in \rho \;\text{ - set of all paths available for TDG,}$$

$$\exists\, d \in D \;\text{ - where } d \text{ is dangerous belonging to a specific DG class,}$$

$$\exists\, \Omega \;-\; geographic\;\; area\;\; in\;\; [X_{positions}, Y_{positions}],$$

where X and Y presents the boundaries of the path in terms of geographic localization.

Therefore, we have

$$(\forall\, \phi_i \in \rho)\;\; (\exists\;\; \Omega) \;\Longrightarrow\; (\phi_i \cap \Omega) \neq \emptyset \tag{8.12}$$

The definition in (8.12) indicates that, in any circumstance, the path expressed by specifying values in Ω should not deviate. This presents a strict constraint where the transportation route includes areas that are densely populated.

2. *Limited geographic area through which specific DG should not be transported.*

   This presents cases where the transported DG has a high risk level, such as radioactive material or infectious waste, such as medical waste. For such material, some areas may strictly restrict the transport of such materials. In such a situation, we have the given area defined on the map:

$$\exists\, \Delta = \int_a^b f(x)dx \tag{8.13}$$

   Definition (8.13) presents a function that determines the specific area over the given map (paths) through which transport of the DG is not allowed.

   In TDG, $\exists\, \lambda$, which determines the current location of the truck.

   Therefore,

$$Transport\;\; process\;\; requires\; (\lambda \leftarrow \phi_i \;\wedge (\lambda \notin \Delta)) \tag{8.14}$$

   The definition in 8.14 indicates that in any circumstance, the truck that is moving on path ϕ_i must not travel through the restricted area.

8.6 Digital Certificate for Traceability Management of DG

The TDG stakeholders, particularly the competent authorities, require surveillance of DG movement across the geographic area under their jurisdiction in and cross-border context.

The stakeholders and even the end customers require information for the physical flow of goods from the departure point up to the destination point. To offer such information, the establishment of a traceability mechanism is required. Traceability is the possibility to track and trace the history, administration, or location of the DG located in the warehouse or during transportation. Tracking and tracing the information of the active and passive processes in the TDG enhances monitoring and auditing aspects. The active traceability makes it possible to know the exact location of the DG that are in transit. The passive traceability enables inquiry of any possible information regarding the completed process in TDG.

To manage the traceability aspects, we present the concept of the *"Digital Certificate"*. The *digital certificate* is established at an early TDG planning stage, before transport starts, by gathering the necessary information. The *digital certificate* remains valid during and after the transport process. The *digital certificate* contains significant information articles for the TDG. Instances of such information include "ID_DG_Process", "ID_DG_Provider", "ID_DG_Receiver", "ID_DG_Transporter (Sub_Contractors_ID)", "ID_DG_Good (loading, qu antity at departure (or arrival), risk level (sensitivity))", "Truck_ID", "Container_ID", "ID_IoT_ Devices", and "Timestamp"). Most of these information articles present on the *digital certificate* are already introduced from our design method development in the previous chapter.

Additionally, we introduce the article "ID_IoT_Devices" representing the set of all IoT devices that are part of our TDG control system. This IoT[6] device allows for capturing digital information from physical objects (truck, containers), provides real-time information for the geographic location of DG, and measures the DG state inside the truck. Furthermore, the "Container_ID" identifies any container (or other type of the load of DG), while "Truck_ID" presents the identification of the truck that moves the DG. In a single transport process, there might be several trucks involved. The "timestamp" identifies the date and time of any activity involving the DG. The aforementioned information articles remain available and are updated during the process flow. New values capture and append in the *digital certificate* articles based on "local information push" and "real-time" information flow. At any time, the authorized stakeholders may retrieve the digital certificate with all the information during an end-to-end process. The information retrieval is further shown in the Merkle-tree style (BC data structure). At a high-level view, the *digital certificate* concept presents a virtual *sub-ledger* formed from the global ledger. It presents an interactive component that allows new information articles to be added based on the need for that information. We propose using SC to gather and store this information as a segregated sub-ledger, maintaining the transaction history for an end-to-end process.

Figure 8.5 illustrates the concepts of a *digital certificate* in an end-to-end transport. As shown here, the certificate is established with significant information articles at the departing point in the transport process. It gathers previous information known from the certification of the stakeholder process and authorization, and combines additional detailed information for the transport process. There might be intermediate stops[7] during the transport process in

[6]We present an extensive study for the BC and IoT integration architecture in Section 8.4.

[7]Contrary to the warehouse where DG is stored for a longer time, the intermediate stop is used for driver exchange or rest, and in terms of time it takes several minutes until to H hours.

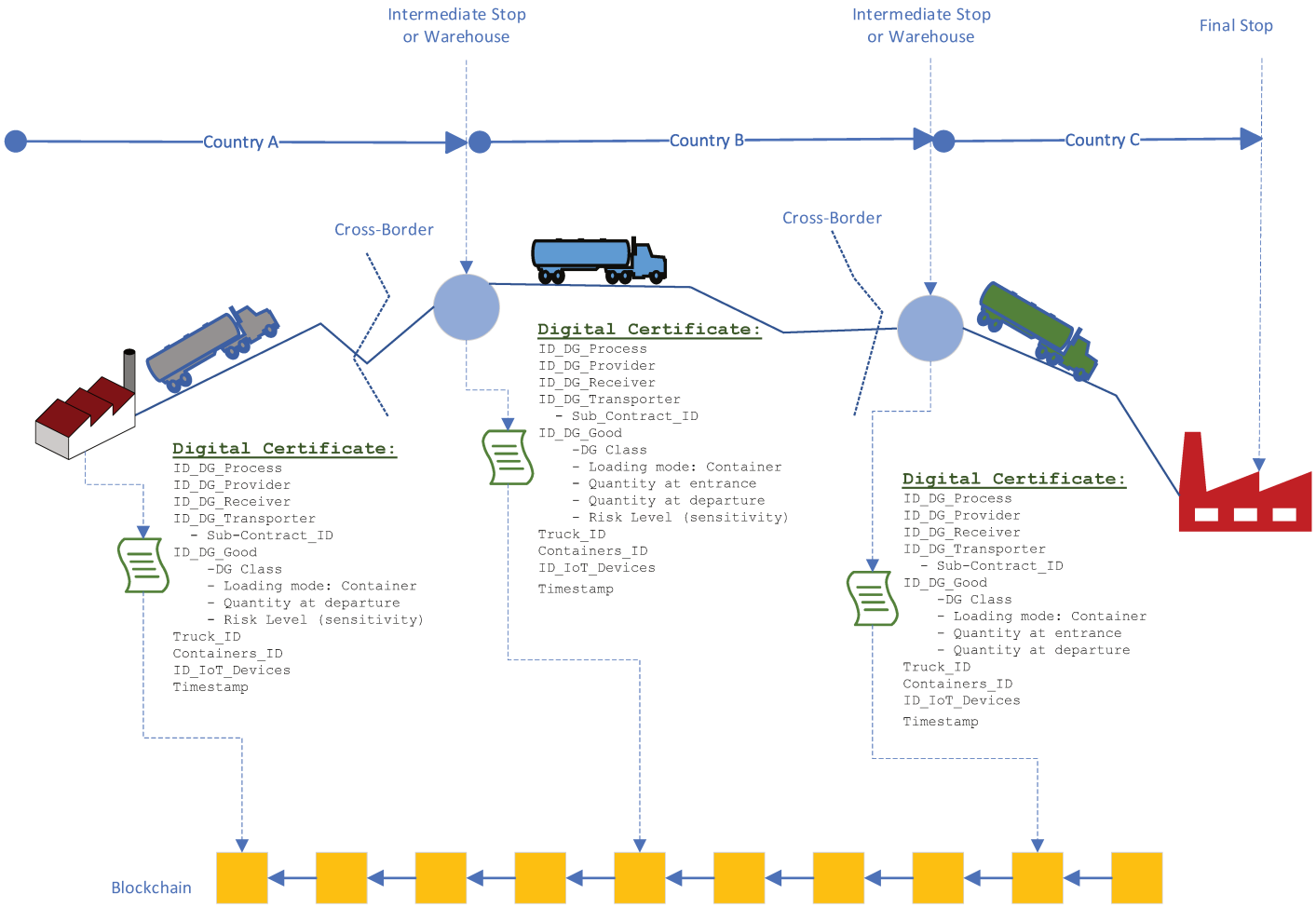

FIGURE 8.5: The concept of Digital Certificate for digital traceability and management of DG.

different countries (e.g., from country A to country B). At any entry on the intermediate stop, the *digital certificate* is updated with the last (on the push and real-time) information. The intermediate stop might play the role of warehousing the DG, meaning that the transport will not continue immediately, and the DG received remains stored there for a certain time. The *digital certificate* remains open for this process, identified by articles "ID_DG_Process". After a certain time, the same DG may be moved to another warehouse or directly to the destination point. It uses the same *digital certificate* to continue the process and update it accordingly.

We formally present some aspects that we consider continuously in the *digital certificate* with the help of SC.

Consider we have the set of DG noted D.

$$(\forall d \in D)(\exists\, quantity(d) = \epsilon) \xrightarrow{transportation} (\forall d \in D)((\exists\, quantity(d) = (\epsilon' \ \vee \ \epsilon) \wedge (\epsilon \neq \epsilon')) \tag{8.15}$$

In the equation (8.15), the ϵ signifies DG quantity. After transporting and warehousing DG, the quantity may differ from its initial measure. This signifies that either the DG is separate in other quantities or used partially (if the warehouse is the destination point). The *digital certificate* calculates these quantities and keeps the ledger updated. Even in the situation in which the separated part is transported to other stakeholders (may be located in other countries), which might be repeated several times (by subcontracting other certified transporters), it still keeps that information until the end of the life-cycle of the DG. That highlights aspects of monitoring and control for DG even if they are separated into smaller quantities. The final step on the digital certificate counts k-parts as the sum of entire quantity of DG ($\epsilon = \epsilon_1, \epsilon_2, ..., \epsilon_k$, thus $\epsilon = \sum_i^k \epsilon_i$) from its departure until its treatment. Formally and empirically, that gives the first indication that the DG is treated according to the regulatory framework and is not misused (thrown in open land or sea).

The digital representation of DG and its characteristics through *digital certificate* enhance the management aspect in TDG. We refer to the ability to manage some characteristics of DG digitally as *digital management*. The *digital management* aspects provide stakeholders with extensive information for the current capacity, type, and related stock information for the DG in the warehouse. Based on that information, they might decide if they possibly host additional quantities of the DG or not. Furthermore, the digital information for the DG distributed in several warehouses allows stakeholders to have the most relevant information about their DG capacities circulation under their ownership. In the context of *digital management*, the information is received digitally. This is unlike paper-based approaches, where we measure the temperature (or humidity) of the arriving DG and write it on the paper. After a certain time, there is not only the disadvantage of one-time temperature measures on arrival, but also there is no mechanism to prove that the temperature was as it is written on the paper. Moreover, there is not a way to come back on that particular day (an hour) and verify the process flow with empirical data. The *digital management* provides digital information in the end-to-end process. Monitoring the state of DG is during the entire end-to-end process enhances quality control aspects and improves the management aspects of the process, and in addition, and in addition, can be verified at any time.

Awareness of the current location of the DG and its condition is managed through the TDG control system. This tracking and tracing feature allows quick response in case of emergency situations identified autonomously by the IoT devices or by manual alerting (by information push). In both cases, the system provides information to the involved authorities and the emergency response teams. In the following, we present details on managing emergency situation (8.7) and advanced features on anomaly detection (8.8).

8.7 Managing Emergency Situations

An emergency is currently an enormous challenge in the TDG. Emergencies and critical situations are not well managed in the existing system, or are very complex to put in place. In an international context of transport of DG, emergency respondents and rescue teams do not have a clear overview of the how to respond in case of any intervention (Magnusson, 2015). The marking and labeling of the DG as described in ADR[8] provides a certain level of information but is considered insufficient for quick, efficient, and safe intervention.

The relevant data that provides exact information describing the DG inside the truck or container is not directly available for the emergency respondents and rescue teams. This information is only visible for the rescue team on arrival at the spot of the accident and not in advance. Rescue teams could make different decisions about the equipment they need to carry to the site if they are aware of all the relevant details of the DG involved. This information might be the type of DG, the risk level including the reaction to various environmental parameters, e.g., water or air, and accident causality (explosion, DG flow, fire, truck failure). Knowing this information in advance would allow emergency respondents to know which team needs to intervene, thus avoiding non-relevant rescue team intervention. Furthermore, it allows them to take the necessary precautions to avoid any emergency crew safety issues. Additionally, the determination of the current DG accidents' location allows the closest and probably most relevant teams to intervene on the accident site.

To enhance the management of the emergency, we propose an approach based on data availability and accuracy.

Let E be an indicator of an emergency situation. An emergency is defined as a consequence of an accident during the transport or storage of DG.

Let $(d \in D \wedge d_{info})$ be the DG that is transported. For any DG (d) registered on the TDG-system control, the d_{info} contains all relevant information regarding the type of DG (class) and the risk level.

Let $\aleph$ be the set of information received from IoT devices deployed in the DG transport vehicle or warehouse. The $\aleph$ contains information like temperature, disturbance, humidity, geographic location, and other relevant information for the DG. At the stage of process initialization, we consider $\aleph \equiv d_{info}$, indicating that the range of nominal situations.

We define an emergency situation as follows:

$$E = \xrightarrow[time\_limit((d,\, d_{info}) \not\equiv (d,\, \aleph))]{\text{emergency situation}} \tag{8.16}$$

[8]The ADR Hazard Identification Number and UN Code give some basic information for DG class.

The formula in (8.16) indicates an emergency situation (storm, fire, accident), while continuously, in a range of time limit, e.g., m minutes, it receives alarm parameters that are not equivalent to the pre-defined parameters for the DG (d). We use SC to store such parameters for managing the emergency in the *"digital certificate"* as previously introduced (8.6). It receives continuous information from the IoT devices and checks if these parameters are in the range of the nominal situation or an indication of an emergency. If an emergency is triggered, it invokes another SC (or function), which further alerts the relevant stakeholders. Based on estimation in (8.16) if the time limit passes, where the alarm is triggered, then it generates a particular emergency event, e.g., *event code: 800*, which is then submitted to a specific stakeholder (emergency respondents), including the geographic location, driver contract, other authorities contact, and all information of d_{info}.

8.8 Anomaly Detection in TDG via Blockchain and Smart Contract

The complexity in transporting DG rises when specific DG are sensitive to different environmental parameters. Several DGs are reactive to environmental parameters such as humidity, temperature, or disturbances. During the transport process, these parameters should be kept constant in the range where they are not considered as a threat. The TDG should be moved within the range of such parameters. If not, it increases the potential for disastrous accidents. To prevent adverse situations, we introduce *anomaly detection* concepts for managing and maintaining TDG under surveillance based on specific parameters emerging from the physical states in the TDG. Our focus on *anomaly detection* is on helping detect and avoid emergency and critical situations. The anomaly detection allows comparing the actual data events received from deployed IoT devices with the "regulated" data pattern. The regulated data pattern presents the "normal" baseline on the process of TDG.

In literature, anomaly detection is described as a pattern recognition process from the data events that are gathered into a data set. The data events that exhibit different behavior from the expected ones are considered an anomaly. The main goal is to detect irregularity while maintaining a low rate of false alerts (Sample and Schaffer, 2013; Demertzis et al., 2020). A system for anomaly detection compares the receiving data set with the "normal" baseline (Patcha and Park, 2007). The literature on BC and anomaly detection is limited. The most significant and recent research is presented in (Demertzis et al., 2020). This study introduces an innovative digital security architecture for securing network communication in the Industrial Internet of Things (IIoT), purposing to solve individual issues of the business environment in Industry 4.0. In this approach, SC is used to implement the bilateral traffic control agreement for detecting anomalies based on the trained DANN[9] (Demertzis et al., 2020). The approach allows the creation of a distributed platform for controlling and completing transactions in critical infrastructure networks without using a central point. The study applies mainly to industrial applications, and it can improve the security and functionality of such applications.

[9]Deep Autoencoder Neural Network.

Beyond the existing research, we use a different approach to detect anomalies. We propose timely behavior monitoring for detecting anomalies during the process of TDG. In the context of the TDG, when an anomaly is detected during the transport process (or at facilities of warehousing DG), the information is received from the IoT devices, which are deployed on the physical transporting object and DG warehouses. The devices send information that is not matching the predefined "regulated" data pattern. We define a data pattern based on the given risk parameters for DG. In such a pattern the risk parameters are *"temperature"*, *"humidity"* and *"disturbance"*. The data pattern is defined before the transport begins and specifies the *"authorized"* range of temperature, disturbance degree, and humidity level[10].

Therefore,

$\exists\ \delta \in [T_{min}, T_{max}]$, where δ is temperature variable and T_{min} and T_{max} determine the temperature interval allowed for specific DG,

$\exists\ \psi \in [H_{min}, H_{max}]$, where ψ is the humidity variable and H_{min} and H_{max} determines the humidity level allowed for specific DG,

$\exists\ \varphi \in [D_{min}, D_{max}]$, where φ is the disturbance variable and D_{min} and D_{max} determines the disturbance degree allowed for specific DG.

We specify our algorithm to capture the misbehavior aspects for the given model for the defined data pattern above. The algorithm compares the current data pattern with the received IoT data streamline.

Let variable Γ be the IoT data streamline. Then for detection of any misbehavior on the TDG process and warehouse, the formal process for anomaly detection is defined as follows:

From the contentious IoT data streamline, represented by $\Gamma\ (\delta)$, $\Gamma\ (\psi)$, and $\Gamma\ (\varphi)$, we receive different measurements:

$$\forall\ (\Gamma\ (\delta) \vee \forall\ \Gamma\ (\psi) \vee \forall\ \Gamma\ (\varphi))(if\ ((\delta \notin [T_{min},\ T_{max}]) \vee \psi \notin [H_{min},\ H_{max}] \vee \varphi \notin [D_{min},\ D_{max}])) \implies Anomaly\ (alarm\ triggering)$$

For all cases where the condition presented in (8.8) is satisfied, an alert message is transmitted to the relevant stakeholder, including the truck driver, as a first-line responsible in TDG.

To avoid false alerts, we specify a condition that guards alerts. It is based on the time interval of the received information from the IoT device.

Let μ - be the defined variables that count the frequency of satisfying situation that verifies condition in (8.8)

$$\text{If } \mu > R_{[temp,\ hum,\ dist]}, \text{ where } R = [R_{temperature}, R_{humidity}, R_{disturbance}]$$

is the corresponding reaction time which is composed based on specificity of DG, and introduced by stakeholders. R is further translated in the time-parameter, meaning that if $\mu = 3$, then the anomaly (8.8) has happened three times, with each time counting s seconds.

In this way, we determine the level of anomaly detection to perform better management of the process of TDG.

[10]These are official parameters that are described on (UNECE, 2017), page 18-34.

8.9 Shared Responsibility

The safety and security of the involved parties in the TDG should be given high priority. In TDG, among other involved parties, some stakeholders provide (hand over) DG, and other stakeholders deliver those DG to the place of destination. In our approach, any interaction between these stakeholders is specified in an SC, and all data shared or exchanged is stored on the BC. This means when stakeholders claim that the handover DG is *"medical waste"*, then this information is immutably stored in the BC. In case, for specific reasons, it is discovered that the same DG is not *"medical waste"*, but something else which presents a much higher degree of risk, e.g., *"infectious substances"*, than the responsibility remains on the stakeholder that handed it to the transporter. The non-declared risk degree of DG may expose higher-level risk for the transporter, particularly to the driver (and DG receiver), by exposing them to potential infection risk.

On the other hand, the transporter should deliver DG as agreed on the business contract to DG receivers. There may be a situation when DG is required to be delivered on time, precisely before the closing time at the DG receiver premises, in order to be able to proceed with such DG immediately (avoiding high contamination or other exposed risks). In any unjustified delay, the transporter cannot claim that they have delivered the DG at the given time while the door to the DG receiver's premises was closed. This is due to the geographic location information received from the IoT device in close real-time and stored immutably on the BC.

Our approach intends to maintain the shared responsibility between stakeholders, thus providing immutable information for any future auditing process, significantly helping to solve any dispute between stakeholders. Each involved party remains responsible for their actions since we intend to store all evidences (track and trace) of any step on the TDG process. To achieve this, we deploy an SC which gathers such existing information. The semantics of this SC brings the necessary information for any process of TDG upon request by stakeholders, i.e., authorities.

8.10 Multi-Party Smart Contract and Business Contract Management

In the TDG, many stakeholders are involved in fulfilling the end-to-end process and have different associated resources according to their operations and activities with DG. There may be a situation where the involved stakeholder may not completely fulfill the transport process, nor the management aspects (in terms of warehouse DG or treatment DG[11]) for a specific DG. To overcome such concerns, cooperation with other stakeholders might be an option for performing end-to-end process management. The TDG management systems may integrate virtually all the stakeholders into one digital environment, representing them as digital objects. In such a digital environment, they can manage processes, securely and confidently share

[11]These are situation where stakeholder has a DG treatment contract but for specific reasons is not able to provide such operations, and proposes to share its beneficial contract with other stakeholders.

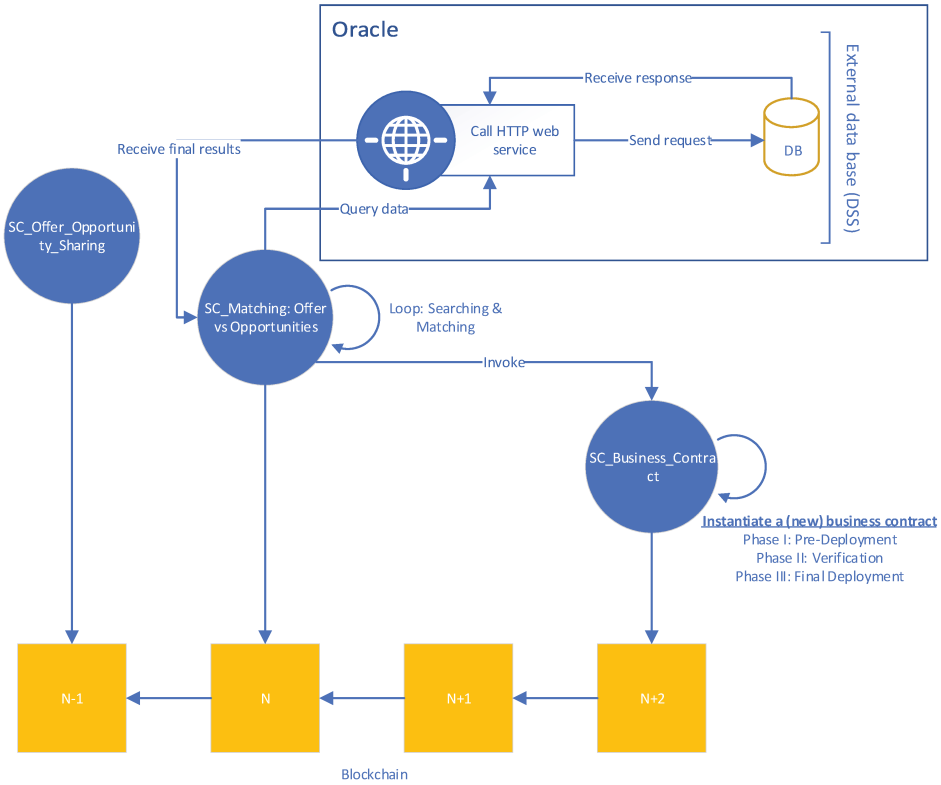

FIGURE 8.6: The concept of multi-party SC in TDG process.

valuable information, and cooperate with involved stakeholders. We introduce the concept of *multi-party SC* to enable different stakeholders to share their resources in terms of operational capabilities and their availability to perform specific TDG related operations. The rationale behind the *multi-party SC* is to allow stakeholders to share their resources on the BC digital environment, which will then be used for digital matching. Digital matching is considered an initial point for the renegotiation of a business contract between stakeholders. Figure 8.6 illustrates the concept of *multi-party SC*. The semantics behind this concept is to enable the deployment of the offer[12] through the *SC_Offer_Opportunity_Sharing*. Another SC, named *SC_Matching_Offers_Opportunites* performs matching between offer and opportunities[13]. When matching the offer and opportunity, the *SC_Business_Contract* is invoked to notify stakeholders and initiate the digital business contract. At this point, stakeholders may exchange additional information and reach an offline agreement. At the final stage, the *SC_Business_Contract* deploys the agreement and all related information on the BC. The deployment of the business agreement by *SC_Business_Contract* is possible after passing three different phases: 1) Pre-deployment phase, which initiates the business contract; 2) Verification phase, which verifies if the pre-deployed contract complies with the regulatory framework; and 3) The deployment phase, which deploys business contracts and associates it with stakeholder and other processes (authorization and process_ID (identification of TDG

[12]Offer contains: DG class, quantity, loading type (tanks, truck), departing point, warehouse, destination, time-frame for transport, and other possible characteristics.

[13]The opportunity characteristics: possible to transport DG, in a specific destination, in specific time-frame, with specific operation vehicles and quantities, and many other opportunity characteristics.

process)). The association with other process allow the monitoring of activities under that business contract.

We consider the matching process challenging and hard to cope with it. Its objective is to find the most relevant opportunity for the given offer. There may be a situation where offers may be specific, and the opportunity may not respond entirely; therefore, a combination of opportunities is required. For example, the specificity comes from conditions to perform TDG, e.g., early in the morning, specific DG class, geographic constraint where one transporter may transport up to the border crossing and another transporter may continue from there (additionally counting other constraints on DG exchanges, authorization process, and many others). In such a situation and probably many others with their specific conditions, it is extremely hard to work with the current BC's design principles. Thus, to achieve this and use the added value of the BC for process workflow maintenance, we consider using external services to match offers and opportunities. The external services enable searching for opportunities for stakeholders to transport and manage DG. Any certified stakeholder may propose its operational capabilities as an opportunity in the TDG for stakeholders that need them. This service is considered trusted since it is hosted by stakeholders and administrated by authorities.

Algorithm 6 expresses a matching possibility based on specific criteria of the offer and opportunity characteristics. Initially, it enlists the offers (lines 2), which are already deployed on the BC. Then, while the offers exist (line 2), the extraction of their main characteristics is performed (line 7). These characteristics are extracted in terms of keywords which are used to elaborate the query. The search from opportunities is performed on the external service, known as "oracles" (Al-Breiki et al., 2020) in the BC jargon. The *SC_Matching_Offers_Opportunites* uses the keyword to query in external services to find the possible opportunities. Based on these keywords, the external services are used to perform matching and check if matching is satisfied (line 8). The matching is performed by querying, comparing, and pairing these characteristics (based on SQL Query (Egenhofer, 1994) or Elasticsearch query (Kononenko et al., 2014)). Once the matching is achieved, the stakeholders are notified (line 9); otherwise, there is no matching between offer and opportunity.

The purpose of using external services to perform matching comes in two folds. First, the matching is extremely hard and specific, and it is almost impossible to express it with the design principles of the current blockchains. Secondly, by performing external querying, we reduce on-chain computation, which further reduces the entire blockchain's performance and might be costly if the computation is paid on the blockchain framework, e.g., public Ethereum.

The *multi-party SC* is used in the following specific cases:

- Business contract association

  A stakeholder publishes its criteria for transporting a specific DG (class, quantity, loading type (tanks, truck), departing point, warehouse, destination, time-frame for transport, and other possible criteria) in terms of an offer. The other stakeholders (in the role of DG Transporter) may publish the opportunity (possible to transport DG,

in a specific destination, in a specific time-frame, and numerous others) to transport DG and their related resources. The opportunities are published on an external service (Oracle). Then the Algorithm (6) matches the offer and opportunity for the TDG by using external services for the matching process. Stakeholders can check and evaluate and probably match with it. They enter into an agreement based on these conditions and consider the condition from authorities and other involved parties.

- In case of enhancing business contract

  This allows the stakeholder to sub-contract other stakeholders to perform TDG operations, e.g., transportation. Once the selection criteria match, it enables them to negotiate offline the possible agreement and execute it online on blockchain as a single witness authority. The agreement should be in line with the regulatory framework since our design method will deny activity (including stakeholder interaction) if they do not comply with the regulatory framework.

- Emergency situation

  In case of an accident (or truck failure) with DG during the transport process, the *multi-party SC* searches on the ecosystem for new alternatives to transport the DG from the point of the accident the destination. The search is performed over the shared resources by stakeholders. This operation is pushed by the driver (or stakeholder who monitors the process) or triggered by an IoT device after receiving the same geographic localization information. We recall that the DG characteristics (type, quantity, dangerous level, departing point, destination points, etc.) are already provided on the blockchain before the TDG starts.

Algorithm 6: Algorithm for matching opportunities to offers in multi-party TDG environment.

```
1  Initialization:
2  list_of_offers [] // presents any available offer;
3  list_of_stakeholders [] // presents any registered
     stakeholder;
4  while (count(list_of_offer []) > 0) do
5      Search for possible matching between:
6      ψ ← characteristics_of_offers(list_of_offer[] )
7      if (matching: ψ ≡ Oracle (characteristics_of_opportunity (ψ))) then
8          // notify stakeholders and assign offer with opportunity
9          SC_Business_Contracts (opportunity (list_of_stakeholders [i]) ← offer
             (list_of_stakeholders [j]))
10     else
11         No matching possible for the actual offers and opportunities
12     end
13 end
```

8.11 Dynamic Smart Contract for Permissioned Blockchain

In this section, we intend to focus on SC design, verification, and execution of SC. The research addressed in this section is related to the maintainability of SC. The SC code cannot be modified (patched) once deployed on BC due to BC's immutability properties. For any modification on the existing SC, there is a need to deploy a new SC on BC. This means a new reference (ID) is generated on the BC for this SC. This situation leads to complicated *maintenance* tasks according to the number of SC to update and the eventual static cross-references. The proposed model aims to enable dynamic behavior into SCs without deploying them again. We present some related works towards designing maintainable SC in Section 4.4.1.

8.11.1 Problem Definition: The Issues of Maintaining the Immutability of the Smart Contract

Immutability: For permissionless or permissioned BC, a SC remains immutable once it is deployed on the BC. This means, all the terms and the logic implemented behind the SC remain unchanged over time. Thus, if we need to make some changes in the SC's logic (add or remove events), we should redeploy it, and a new hash will generate as an *ID* for that SC. The user who needs to use this SC must know the newest address of SC before being able to invoke it. Otherwise, the user will call the old one since its hash value is already mapped on the current SC. Figure 8.7 illustrates the issues of changing to the SC address.

The new address of the contract should be distributed to all stakeholders that are invoking this SC. The concern is that all the other SC that have invoked this SC (now with a new *hash address (ID)*) should be changed. That means that we have to reconfigure the entire system (i.e., all objects need to have a new address). For instance, if there are thousands of SC that call the changed SC, this would be extremely difficult to reconfigure (cf. maintenance), and the performance will become a concern for the BC-based applications. Furthermore, this implies the automation capability of the process decreases in this sense.

This problem is a concern for stakeholders intending to move some of their business logic over the BC. To maintainability, we need modularity, which means that a complex problem should be divided into several minor problems to solve them much more easily. Similarly, with the code libraries, using several inter-connected SCs becomes useful for high-level business processes. Thus, we can assume a complex system using several SC with cross-references and automation. However, using static addresses for the SC that are hardcoded in the SC logic leads to uncomfortable maintenance, as explained above.

Cross-references: Involving several SC and cross-communication to perform high-level tasks still exposes similar problems. Figure 8.8, shows the issues of logic flow for cross-reference SC. There will be another SC address to know if a SC logic updates either from the caller side or from the executor side. For instance, considering two SC, where SC1 calls SC2, and if we only update SC2 (to SC2′), we must change SC2 address reference into SC1 to point to SC2′. That also implies changing SC1 to SC1′. These administrations are not ideal for a system in production. However, it is even worse when we have cross-references, i.e., if SC1

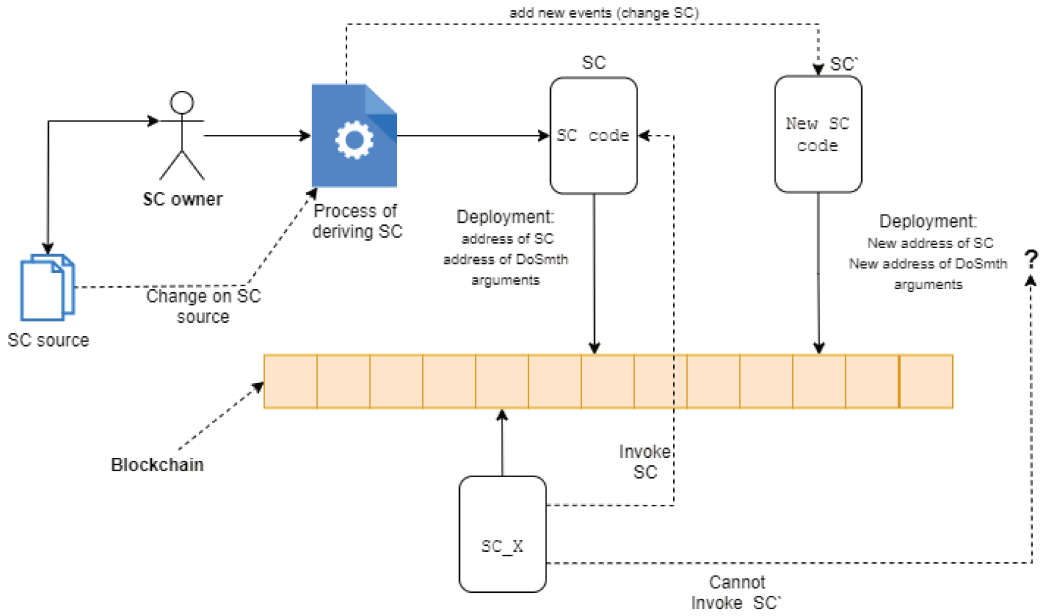

FIGURE 8.7: The schema for presenting the static referential SC addresses issues.

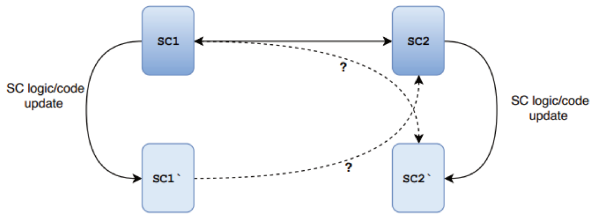

FIGURE 8.8: The problem of cross-references for smart contract.

calls SC2 and SC2 calls SC1, then we fall into a deadlock situation because SC1 and SC2 have a hardcoded address of each other. That comes from the fact that we cannot guess the future SC address to hardcode it in advance in the logic. In such a case, there is no workaround to do except avoid bidirectional cross-references for maintainability.

8.11.2 Use Case of Temperature Checking. The Issues of Smart Contract Maintainability in a Dynamic Environment

Temperature checking: Considering a SC that aims to check the temperature collected by one IoT device and storing these data directly into the BC. This SC is checking on demand the current temperature (from the IoT device), and if this temperature exceeds a specific threshold, a set of users (stakeholders) should be notified based on the pre-defined condition in a particular use case, e.g., for DG that is sensitive to a temperature degree (Imeri et al., 2017).

Detailing the features of this SC, it has three core functionalities. The first is to **collect** the temperatures provided from IoT sensors. This IoT device is authenticated (based on the approach showed in section 8.4) and transmitting its data directly in the BC. The second

functionality of this SC is to **set** and change the temperature threshold. The third functionality is to **notify** all involved stakeholders when the temperature threshold is exceeded.

The issue in such use case is regarding the **threshold** and the **set** transactions. The common approach is to define the threshold as a static constant into the SC code (i.e., as a hardcoded and immutable variable). Consequently, we must update the SC each time the constant need to change, and this exposes enormous problems, as we mentioned above.

For overcoming these issues, we propose a new way of managing the SC by storing the threshold constant as a variable into a SC *asset*. The SC asset is an editable variable that allows to update it dynamically (as required by the use case) and avoid the change of SC code. The *temperature checking* use-case does not deal explicitly with cross-references because there is only one SC, but the solution mechanism remains the same. For both SCs, storing the target SC address as a variable (same as in *threshold* case) allows change that addresses and then unlocks the deadlock situation dynamically. We present the approach of dynamic SC for permissioned BC in Section 8.11.3.

8.11.3 Dynamic Smart Contract for Permissioned Blockchain Based on Dynamic Parameterization

In this section we describe the propose approach for managing SC dynamically in permissioned BC.

8.11.4 Dynamic Parameterization

Our approach allows defining SC, specific to a use case, that will have a static code deployed on the BC, but it will run dynamically. Mainly the dynamic part of these SC remains the parameters of their transactions. We propose the usage of the BC technological features to store data, which further enables the possibility to store "dynamic parameter" (*DynParam*[14]) into it. The "dynamic parameter" is considered a variable or an *asset* following the HF terminology (3.4.3). This variable leads to relying the SC code on that internal data (i.e., constant) in order to have a dynamic behavior for the cases when the *DynParam* is updated in an SC that has immutable (static code). Emphasizing that providing this *DynParam* as parameter of the SC transactions is an external input e.g., from the external API call, or "Oracle" (Al-Breiki et al., 2020) in the Ethereum (ETH) community.

8.11.5 State Machine Representation

Figure 8.9 shows how *DynParam* works for the two functionalities *Set* and *DoSmth* as part of the SC. *Set* corresponds to the ability to set (if it does not exist) or update (if it exists) the dynamic parameter that will be stored in the BC. That variable can be of any type, even though it is usually a string or integer. The *DoSmth* transaction can be of any purpose while using this *DynParam* to adapt the code's behavior according to its value. Further, if the *DynParam* is not set (defined) the SC cannot run (Locked state). If *DynParam* is set, we can

[14]The dynamic parameter term presents a constant (which is static for the time being), and it will change when a specific SC is called for updating its value, then globally it turns to be dynamic for long-time point of view.

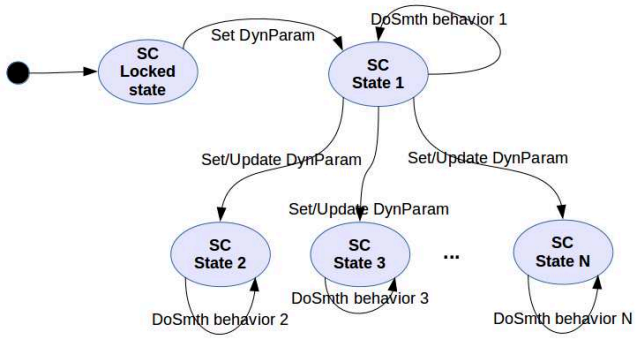

FIGURE 8.9: The state machine schema for the dynamic SC for the permissioned blockchain approach.

run the *DoSmth* functionality, which will be one behavior (or state, for example state 1). If we update by setting another value in *DynParam*, then the *DoSmth* transaction will change in consequence (i.e. behavior (states 2, 3, ... N)), still based on the same static code (logic). This solution assumes that the *DynParam* is given by one authorized user from the outside (off chain) of the BC and checked by the transaction itself to accept or revoke it (Imeri et al., 2019e).

Moreover, for automation purposes, we can easily extend that solution by substituting the user by an automated call of the SC to an external database to getting back the new value for the *DynParam* while the user identity is still known and allowed by the SC.

For the proposed approach, we have implemented proof of concepts (PoC) solution for temperature checking in TDG use case, showed in (Imeri, 2019), and some of the results are shown in Chapter 9, particularly in Section 9.2.10.

8.12 Formal Specification of Smart Contract for capturing the semantics in SCM for TDG

This section formally illustrates the semantics of SC in terms of formal specification of business rules for the TDG. These semantics means the SC are behaving in accordance with the defined business rules. SC can be expressed as a finite state machine (FSM) model. We showed a formal definition of the SC with the help of FSM in Section (3.5). For better handling of the functionalities of the SC, the FSM is extended with guard conditions on transition labels. The guard is a Boolean expression that enables conditions of initiating (firing) a particular transition. The semantics of the initiating a transition in FSM is affected by a guard: the transition is enabled if the guard condition is satisfied (being true). The current most usable SC platform, such as Ethereum, recommends designing the SC as FSM (Buterin, 2017; Wood, 2014). A single invocation of the SC function generates specific output transaction data from a given input transaction data. That signifies, it update the shared ledger after each successful transaction. This invocation may update the ledger state according to the SC owner's intention or, conversely, generate some unexpected output, leading to a disadvantageous situation. To avoid a disadvantageous situation in cases when

the SC generates the unexpected output or SC behaves in opposite with our expectations, the formal specification and verification of the SC are considered necessary.

The formal specification of SC allows expression of the SC properties, while the formal verification (model checking or theorem providing) verifies if the model behaves according to that specification. There are different methods for formal specification and verification of the SC. The research from (Murray and Anisi, 2019; Tolmach et al., 2021; Singh, 2020) shows SC formalization methods such as "first-order-logic", "temporal logic" and "symbolic expression". In (Murray and Anisi, 2019) the *Theorem Proving* and *Model Checking* are presented for the formal specification and verification of SC. The formal specification and verification of the SC enable *reliability* of the system, meaning that the system will not fail at a certain time based on its formal specification.

We propose using *Temporal logic* as an appropriate formal specification for SC. We present an extensive study for mathematical logic and temporal logic in Appendix A.8.

8.12.1 SC Formalization in Linear Temporal Logic (LTL)

The formal specification avoids ambiguities and double-meaning in the system requirements specification. The system models should behave according to their specifications. To achieve that, it is necessary to have a precise specification that the model relies on and behaves accordingly. A potential precise specification is mathematical-logic, which allows us to express specification unambiguously and concisely (Rozier, 2011). For the SC formal specification, we consider Linear Temporal Logic (LTL) (A.8.2.1), which allows us to express the desired behavior for the SC.

In the context of our system design, SC composes the core of the targeted system. SC carries the main business logic functionalities supported by well-defined business rules. On the targeted system for the TDG, we intend to specify aspects that we consider critical and over which we intend to rely TDG process flow and business contracts. These aspects are considered to be significant steps in the TDG process workflows. We propose a formal specification of SC based on the business rules. We recall that in the context of TDG, business rules are mainly determined in the regulatory framework (as shown in Chapter 7). This specification determines how the SC will behave by imposing specific constraints, thus ensuring a regulated (complaint) TDG process.

8.12.1.1 Formal Specification of Business Contracts Applied in Smart Contract

This section presents some aspects of business contracts that need to be formally specified to avoid ambiguities and misbehavior with regard to the regulatory framework. TDG is maintained by the stakeholders who operate this process. These stakeholders enable this process with their cooperation. Usually, the cooperation is performed through business contracts that should be in line with the regulatory framework. TDG is a regulated process, and business contracts should rely on a regulatory framework to enable the correct functionality of the targeted system. We present SC formal specification, procured from the regulatory framework, to manage several aspects of the business contracts.

1. `business_contract` cannot happen before `consent` from authority;

$$(\texttt{business\_contract}\ \mathbf{R}\ \neg\ \texttt{consent})$$

   That indicates the authority should be aware of `business_contract` between stakeholders and this enables the authorization process.

2. `authorization` cannot happen before `business_contract`;

$$(\texttt{business\_contract}\ \mathbf{R}\ \neg\ \texttt{authorization})$$

3. `business_contract` cannot happen after `authorization` has happen;

$$\mathbf{G}\,(\texttt{authorization}\ \rightarrow \neg\,\mathbf{F}\,\texttt{business\_contract})$$

   That indicates that `business_contract` should be part of an authorization request and, it is not possible to have an authorization without a business_contract between stakeholders, i.e., DG_Provider and DG_Receiever

4. `business_contract` with stakeholder for given process (authorization) can happens only once;

$$\mathbf{G}\,(\texttt{business\_contract}\ \rightarrow \mathbf{X}\neg\,\mathbf{F}(\texttt{business\_contract}))$$

5. `transport` can happen only if `business_contract` has happen;

$$\mathbf{G}\,(\texttt{transport} \rightarrow \texttt{business\_contract})$$

   These are situations, where stakeholders prefer to select only specific stakeholder, in a way to fulfill the business requirements. This specification may give stakeholders i.e., DG_Provider to select particular transporter through the `business_contract`.

8.12.1.2 Formal Specification of Smart Contract for Governing TDG

We present a SC model for the TDG process flow in an end-to-end manner. This SC plays the role of a guard to maintain the process flow. Figure 8.10 presents a simplified model developed in FSM for the SC. This model is sourced from the user model (BPMN) showed in Figure 7.9. It is composed of S-States (for brevity reasons inside the state circle, the corresponding states are as follows: A-DG_Provider, B-Authorities, C-Transporter, D-Transport (process), E-DG_Receiver, F-DG_Traitment, and St_ P initial state). The transition in the model is enabled if certain conditions are fulfilled. Guards on the transition enable such conditions. The successful transition execution provides changes in the SC states, thus changing the global ledger (BC). The FSM and its states consist of all possible computations of the SC. The SC computational aspects expose an infinite sequence of SC states that correspond to the

SC behavior. Following, we present a traversal example for the model shown in Figure 8.10. We start on state St_P which indicates the process started. Following, we traverse to state A. If A intends to hand over DG, we go initially to state B to request authorization. Once authorization is approved, it brings us to state A. With the authorization at hand, state A hands over DG, and we are placed on state B. Once the transport process starts, we are now in state D. From state D, we can reach state A according to the conditions. If the process of transport flows normally, we reach state E. From E, we can reach state A according to the conditions. When the DG treatment is performed, we are in state F. Once the certificate of treatment of DG is completed, we can reach states A, B, C, and we close the cycle by reaching the initial state (St_P).

A compliant system with a regulatory framework requires applying meta-rules to base the compliant system behavior. As presented in the previous section, these meta-rules are procured from a specific regulatory framework (8.2.1). Furthermore, we present a formal specification of the SC that guards the process workflow. It identifies each process in an end-to-end manner and covers each step, including its constraints. We recall the main process flow to enable SC specification.

In TDG, the "DG Provider" hands over DG to the "Transporter", after administrative and physical preparation. It is not possible to transport DG without authorization and movement documents, as described in sections (7.3.1.3 and 7.3.1.4). Even those for which a financial guarantee is required for the transport process (Article 6 in Regulation (EC) No 1013/2006 (EC), (Commission and Parliament, 2006)). Once DG is delivered ("DG Receiver"), then DG treatment can be performed. If the DG treatment is performed, that enables the formation of the DG treatment certificate. The treatment certification should be sent to "DG Provider", "Transporter", and "Authorities". This allows starting procedures for releasing the financial guarantee and indicates the completion of the process of TDG, and closes it, if everything goes well, archiving information regarding the process.

The statement enables us to refer to the business rules (showed in Chapter 7). Furthermore, we derive several rules that change the SC states and translate the SC's informal specification as desired properties for the targeted system. For the given model shown in Figure 8.10, we present the formal specification of SC based on the LTL formulas (showed extensively in A.8.2), for an *end-to-end* example as follows:

1. `authorization` cannot happen more than once;

$$\mathbf{G}\,(\texttt{authorization} \rightarrow \mathbf{X}\,\neg\,\mathbf{F}\ \texttt{authorization})$$

2. `financial guarantee` cannot happen more than once for the authorized process;

$$\mathbf{G}\,(\texttt{financialGuarantee} \rightarrow \mathbf{X}\,\neg\,\mathbf{F}\ \texttt{authorization})$$

3. `process_close` cannot happen more than once;

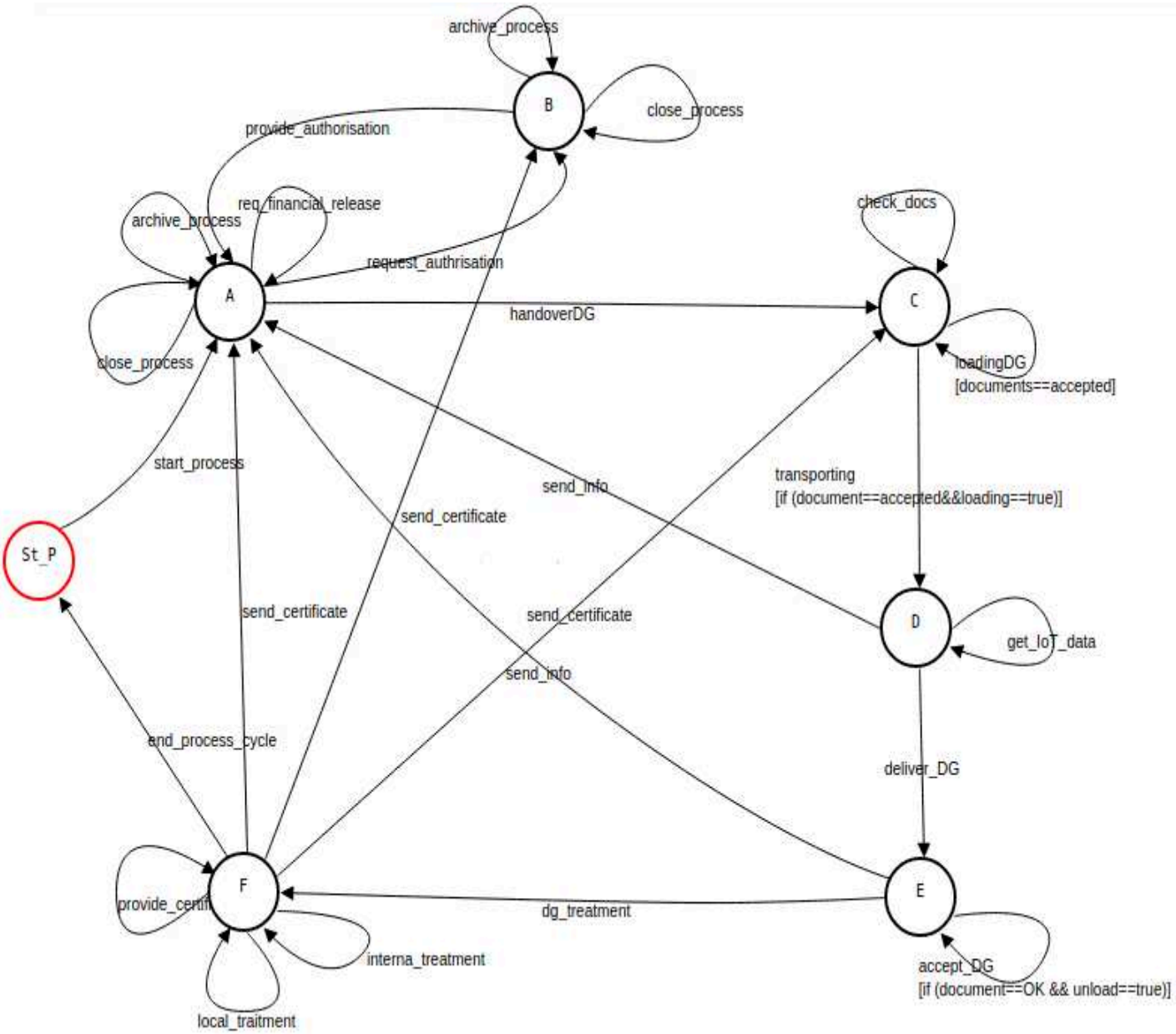

FIGURE 8.10: The FSM model for formal specification of SC.

$$\mathbf{G}\,(\texttt{process\_close} \rightarrow \mathbf{X}\,\neg\,\mathbf{F}\ \texttt{process\_close})$$

4. `archive_process` cannot happen more than once;

$$\mathbf{G}\,(\texttt{archive\_process} \rightarrow \mathbf{X}\,\neg\,\mathbf{F}\ \texttt{archive\_process})$$

5. `handoverDG` cannot happen before receiving `authorization` (notification);

$$(\texttt{handoverDG}\ \mathbf{R}\,\neg\ \texttt{authorization})$$

6. `DG_treatment` cannot happen before receiving `authorization` (notification);

$$(\texttt{DG\_treatment}\ \mathbf{R}\,\neg\ \texttt{authorization})$$

7. `movement_doc` cannot deliver more than once;

$$\mathbf{G}\,(\texttt{movement\_doc} \rightarrow \mathbf{X}\,\neg\,\mathbf{F}\ \texttt{movement\_doc})$$

8. `handoverDG` cannot happen before receiving `movement_doc` (notification);

$$(\texttt{handoverDG}\ \mathbf{R}\,\neg\ \texttt{movement\_doc})$$

9. `handoverDG` cannot happen before receiving `financial guarantee` (notification);

$$(\texttt{handoverDG}\ \mathbf{R}\,\neg\ \texttt{financialGuarantee})$$

10. `handoverDG` cannot happened after transporting DG (`transport = True`)

$$(\texttt{transport} \rightarrow \mathbf{F}\,\neg\ \texttt{handoverDG})$$

11. `transport` cannot happen before delivering `movement_doc, authorization and handoverDG`;

$$(\texttt{transport}\ \mathbf{R}\,\neg\ (\texttt{movement\_doc} \wedge \texttt{authorization} \wedge \texttt{handoverDG}\ ))$$

12. `receiving_DG` cannot happen before transport;

$$(\texttt{receiving\_DG}\ \mathbf{R}\,\neg\ \texttt{transport})$$

13. `handoverDG and transport` cannot happen after delivery DG (`receiving_DG = True`) ;

$$(\texttt{receiving\_DG} \rightarrow \mathbf{F}\,\neg\ (\texttt{handoverDG} \vee \texttt{transport}))$$

14. `receiving_DG` cannot happen more than once;

$$\mathbf{G}\,(\texttt{receiving\_DG} \rightarrow \mathbf{X}\,\neg\,\mathbf{F}\ \texttt{receiving\_DG})$$

15. `DG_treatment` can happen only after `receiving_DG` has happened;

$$\mathbf{G}\,(\texttt{DG\_treatment} \rightarrow \texttt{receiving\_DG})$$

16. `certificate` cannot release before treatment;

$$(\texttt{certificate}\ \mathbf{R}\,\neg\ \texttt{DG\_treatment})$$

17. `certificateSubmission` can happen only after treatment;

$$(\texttt{certificateSubmission}\ \mathbf{R}\,\neg\ \texttt{DG\_treatment})$$

18. `financial_guarantee_release` can happen only if `certificate` is `received` `(cert_received)`;

$$\mathbf{G}\,(\texttt{financial\_guarantee\_release} \rightarrow \texttt{cert\_received})$$

19. `process_close` can happen only if `certificate` is received and the and financial_guarantee_release = *True*;

$$\mathbf{G}\,(\texttt{process\_close} \rightarrow (\texttt{cert\_received} \wedge \texttt{financial\_guarantee\_release}))$$

20. `archive_process` cannot happen before closing the process;

$$(\texttt{archive\_process}\ \mathbf{R}\,\neg\ \texttt{process\_close})$$

8.12.1.3 Time Constraints Specification

For safety and security reasons, the transport and management of DG, especially for the particular DG class (e.g., infectious or radioactive), requires additional constraint to the transport process by imposing a specific time-frame. For example, the transport process should not start before midnight (e.g., 00:00H, starting time) and not after 00:30H to arrive at the destination or intermediate stop, e.g., warehouse, before 06:00H (arrival time). In case of any delay on the departure side, the targeted system must deny any start of the transport process since the travel time is calculated in advance. The departure time might be affected by different factors, e.g., preparing DG for transport, DG loading, or weather conditions.

For formal specification of such SC with the help of LTL formulas as follows:

1. `transport` cannot happen after `starting_time` has happened;

$$\mathbf{G}\,(\texttt{transport} \rightarrow \neg\,\mathbf{F}\ \texttt{starting\_time})$$

2. `transport` can happen only if it arrives before the `arrival_time = True` ;

$$\mathbf{G}\,(\texttt{transport} \rightarrow \texttt{starting\_time})$$

8.12.1.4 Formal Specification of Geographic Constraint at Smart Contract Level

We present a formal specification of the SC for applying geographic location constraints according to travel time and geographic area. A traveling geographic area is determined by stakeholders using the formula given in (8.13). Through that geographic area, there might be several traveling paths. The geographic area and transportation time are strictly considered in the path determination where some constraints are imposed for safety or contractual business reasons.

In the TDG, there are several traveling paths for the transport process, e.g., `Path_1, Path_2, Path_3, Path_4,..., Path_N`. Along these paths, a transporter might transport DG according to the agreement for TDG. The agreement specifies that for TDG through a given area, paths, there is a time frame for DG transportation through it. Paths compose an end-to-end road, and all the paths are connected, for example, $Path_{J+P}$ cannot be reached before completing $Path_J$.

We formally specify SC in a way to limit vehicles (truck) moving to specific paths in the given time:

1. Only one path can be selected. This specification determines that only one single path can be the selection at the time.

$$\mathbf{G}(\texttt{(Path\_X)} \rightarrow \neg\ \texttt{(Path\_Y)})$$

2. `Path_2` cannot reached before completing `Path_1, Path_3` cannot reached before finishing `Path_2` and Path_J+P cannot reached before Path_J ;

$$(\texttt{(Path\_1}\ \mathbf{R} \neg\ \texttt{Path\_2)} \wedge\ \texttt{(Path\_2}\ \mathbf{R} \neg\ \texttt{Path\_3)} \wedge (\ \texttt{Path\_J}\ \mathbf{R} \neg\ \texttt{Path\_J+P)})$$

The concept is to determine that path $(J + P)$ cannot be reached before path J. For example, if the area has multiple paths and DG movement is performed from path "J" to path "J + P", where P $\in \{1, 2, 3, 4, ..., N\}$, then the formal specification determines that these paths should be reached accordingly.

3. `transport` operate in paths Path_1, Path_2, Path_3 and not in `(Path_X, Path_Y,...)` if the `arrival_time` is between (18:00H-21:00H)

$$\mathbf{G}(\texttt{transport} \rightarrow (\texttt{arrival\_time} \wedge ((\ \texttt{Path\_J}\ \mathbf{R} \neg\ \texttt{Path\_J+P)} \wedge \neg\, \mathbf{F}\ \texttt{(Path\_X} \vee \texttt{Path\_Y)}))$$

The formula above expressed with temporal operators indicates that some paths are forbidden for vehicles to pass with DG in a certain time interval. The transport might pass through any other available path (Path_J+P, where P varies), but transport should not pass through paths `Path_X and Path_Y`. Indeed applying different time intervals is possible by using guards in FSM. That allows the application of different time intervals (hh:mm - hh:mm) for specific paths.

8.12.1.5 Constraint-based Path Determination

We present an algorithmic specification, showed in 7 for the path determination under time and geographic localization constraints in TDG. These are dynamic constraints, meaning that they evolve continuously. Initially, we present the main parameters for initializing this algorithm, and there are imposed conditions for determining specific paths to follow in TDG. The parameters such as *starting_time, arrival_time, accepted_traveling_time*, allows determination of specific time-related conditions, and are specified from stakeholders, while *geographic_area* determine the allowed geographic area for TDG. These restrictions, geographic areas, and travel time are imposed to select the "next" path. The *transportation_time* (line 7) indicates the required time for an end-to-end transportation. The process of TDG must not start, indicated by *TDG_process_start = false* (line 9), in case the *transportation_time* does not fulfill the required *accepted_traveling_time*. The selected path must belong to the geographic area, then the process of path determination begins (line 15). Otherwise, a specific DG is not allowed for transport in that area (line 33). It searches for any possible path available on the list of paths (lines 18). If the selected "next" paths do not fulfill the condition of being in an accepted geographic area, and the traveling time in that path is not acceptable, it selects another path (line 20). The algorithm checks continuously for the possible paths if they fulfill the condition (geographic and time constraints, line 24) and store them in the paths list (line 25). Finally, the algorithm will return the list of paths available for TDG in a given geographic area and strictly time-frame imposition.

Algorithm 7: The algorithm for path determination under constraints in TDG.

```
Initialization:
starting_time; arrival_time; accepted_traveling_time;
geographic_area[][]; list_of_paths[]; current_path; next_path;
TDG_process_start = true; TDG_transportation = false;
determined_listOfPaths[]; available_listOfPaths;
travel_time_path[];
transportation_time ← (arrival_time − starting_time);
if (transportation_time !== accepted_traveling_time) then
    TDG_process_start = false;
else
    //start path determination:
    TDG_transportation = true;
    travel_time_path[] ← travel_time_path(list_of_paths[]);
    available_listOfPaths ← count(list_of_paths[]);
    if (current_path in list_of_paths) then
        i = 0;
        j = i + 1;
        while (TDG_transportation & available_listOfPaths >= 0) do
            if (next_path[i] not in geographic_area ∨ travel_time_path[next_path[i]] !==
             accepted_traveling_time) then
                current_path ← next_path[j]
            else
                current_path ← next_path[i]
            end
            if (next_path[j] or next_path[i] in (geographic_area and
             travel_time_path[next_path[i] or [j]] == accepted_traveling_time)) then
                determined_listOfPaths[] ← current_path;
                available_listOfPaths - - ;
            else
            end
            The current paths are not allowed for TDG in given time
        end
        i = i + 1;
    else
        current_path not in geographic_area (list of paths);
    end
    return determined_listOfPaths;
end
```

```
MODULE main
 VAR
   pathDetermination : boolean;
   path1             : process pathSelection(pathDetermination);
   path2             : process pathSelection(pathDetermination);
 ASSIGN
   init(pathDetermination) := FALSE;

 LTLSPEC G (path1.state = Allowed  -> !(path2.state = notAllowed))
MODULE pathSelection(pathDetermination)
 VAR
   state : {netural, PathSelectionIssues, notAllowed, Allowed};
 ASSIGN
   init(state) := netural;
   next(state) :=
     case
       state = netural                      :{netural, Allowed};
       state = Allowed & !pathDetermination :PathSelectionIssues;
       state = Allowed                      :{Allowed, notAllowed};
       state = notAllowed                   :netural;
       TRUE                                 :state;
     esac;
   next(pathDetermination) :=
     case
        state = Allowed    : FALSE;
        state = notAllowed : FALSE;
        TRUE               : pathDetermination;
     esac;
 FAIRNESS
   running
```

Verification:

```
C:\Program Files\NuSMV-2.6.0-win64\bin>NuSMV.exe paths.smv
-- specification  G (path1.state = Allowed -> !(path2.state = notAllowed))  is true
```

FIGURE 8.11: The NuSMV model-checking for specification of path selection.

8.12.2 Model Checking of Smart Contract

This section presents the model checking and verification components for the specification we provided in the previous section.

8.12.2.1 Model Checking

Model checking is an automatic process that allows the verification of models. The model is expressed in a finite state machine (FSM), and the verification mechanism consists of efficient inspection of all possible states described by the model. The verification intends to prove that the models satisfy the desired properties specified with temporary logic. The intended model is formally checked if it behaves according to its specification. If the model does not satisfy the considered properties, the model checker produces a counter-example, highlighting the sequences that violate those properties, thus determining the issues on the model (Rozier, 2011). Among the main properties that model-checker verifies are *safety*, which indicates that something bad never happens; *reachability* which determines the possibility to end up in a given state; *fairness* which checks whether, under specific conditions, an event happens repeatedly; *liveness*, express that something good eventually happens; *functional correctness*,

which indicates if the system does what is supposed to do, and *real-time* properties which checks is the system is acting on time (Pnueli, 1977).

There exists several model checking tools including NuSMV (Cimatti et al., 2000), UPPALA (Larsen et al., 1997), SPIN (Holzmann, 1997), that enables model checking and verification (Baier and Katoen, 2008; Cassez et al., 2001). For performing model checking, we use NuSMV, as it is an open-source, well-documented, and appropriate model checking tool for expressing linear temporal specification (Rozier, 2011). NuSMV is a well-known model checking tool that has been used in several scientific and commercial projects.

Following, we present an example of model checking for specifications for the path selection presented in the previous section. We specify the NuSVM model and further apply model checking. The NuSMV model checker will formally prove (fairness properties) the unique path selection properties, shown in Section 8.12.1.4, point 1. The NuSMV model, including verification, are shown in Figure 8.11. The model checking ensures the correct functionality (or finding counterexamples if the model does not satisfy specification) of the model at the design model. The SC intended for the targeted system must be based on these verified models.

8.13 Conclusion

The objective of this chapter was to present some advanced concepts that are applicable in the TDG and other use cases in the domain of SpC and transportation. Initially, we showed the concept of meta-rules, and their purpose is to provide additional semantics in terms of constraints to be applied in the process of TDG with the help of SC. Further, we show an approach for BC and IoT integration. To highlight the dynamic aspects in the process of TDG, we presented time-related constraints (including strong and weak time constraints), geographic location constraints, the concept of the digital certificate, managing emergencies, the anomaly detection concept, shared responsibility, the multi-party SC, and SC maintainability approach. In general, time addresses distinct challenges in the current blockchains, and for overcoming such challenges, we propose a conceptual solution based on a dynamic variable. The proposed approach uses a shared variable to maintain and apply time constraints over the TDG process workflow.

Because of the sensitivity and risk exposure of the DG, geographic constraints are proposed as a preventive to avoid potentially disastrous situations in specific geographic areas. Its objective is to determine the specific set of geographic location constraints and to manage the process of TDG accordingly within the regulatory framework and specific conditions imposed by local (or international) authorities. Furthermore, to maintain the end-to-end lifecycle of the DG, we propose the digital certificate concept. It presents a digitally computed document for specific DG. It enables storing continuously immutable transaction-data, which enables identification, traceability, and transparency over the DG lifecycle usability and treatment. Similarly, this concept can be used in other related domains where traceability and transparency of goods are required.

The management of emergency situations presents enormous challenges in the TDG. We propose a conceptual solution for managing emergencies based on the information received from IoT devices. This enables an accurate and adequate intervention process for the responsible stakeholders in the event of an accident with DG. To maintain and monitor the process of TDG under certain control, we propose the concept of anomaly detection. We specify SC for monitoring the state of DG and alerting when an "abnormal" situation is detected. For better management of the DG, we present two management-related concepts. Shared responsibility quantifies the stakeholders' responsibility involved in the process of TDG. The multi-party SC intends to improve the business ecosystem in the TDG and provide means to avoid risks in TDG, while for maintainability issues of SC, we proposed an approach based on the dynamic parameter (assets) change on the SC.

Furthermore, this chapter presents the formal specification of the SC for the TDG, with the help of Linear Temporal Logic (LTL). The LTL enables the SC specification based on the temporal time-logic parameters. It allows the expression of desired behaviors of the SC in terms of the business rules for governing the TDG process, applying specific time geographic localization and path determination constraints.

SC implements the business logic that must follow the business rules. The formal specification of SC allows the expression of business rules that determine the TDG process flow. SC must follow these specifications. In the context of TDG, the business contract are determined from the business rules procured from the regulatory framework, and it is required continuous maintenance of its contract (TDG) operations with the regulatory framework. The formal specification and further SC implementation based on this specification enables the management of business contracts in compliance with the regulatory framework. The specification showed in Section 8.12.1.1 can be implemented in a single SC or partially in different SC based on the design-decision taken from the system designer. We express the targeted system business rules and desired behaviors through such specifications. The presented specifications are purposive for the use case of TDG, but they may be used for the other use cases in this function. We consider this formal specification of the business rules as a template for possible usage for any other business case. This indicates the same specification may be used for different contexts or different use cases in TDG.

Another objective of this chapter is to verify whether the specified business rules (or desired properties) and the applied constraints are expressed correctly. To achieve this, we propose using model checking techniques. It allows formal verification that the model which presents the SC is behaving according to the expressed rules. This verification is performed at the design level, and it gives an early estimation of the SC's functionality. The SC is an integral part of the targeted system; thus, its functions are based on these specifications.

The importance of using the formal specification and verification of the SC is on better expression and verification of the desired properties of the targeted system by designing SC-models. The objective is to verify whether the targeted system will perform accordingly to the specifications.

Chapter 9

The Proof of Concept (PoC) Implementation for TDG-control System

9.1 Introduction

This chapter shows details of Proof of Concepts (PoC) implementation for the TDG-control system based on our design method solution, as shown in Chapter 6. Since our proposed method is technology agnostic, it allows us to select any BC platform that permits the TDG-control system development. Initially, we present the technical architecture of the TDG-control system, which describes the associated technological components. Furthermore, the BC platform components that enable the development of the solution are presented. The descriptions of the main functionalities of the TDG-control system are shown, and also we present conceptual validations resulting from our design method. Since we are developing this BC-based solution in cooperation with the Luxembourg local competent authorities for managing TDG, we will show implementation details only as agreed in a "signed non-disclosure agreement".

9.2 The TDG-control System Architecture and Related Components

This section presents the TDG-control[1] system architecture and the associated technological components.

For the PoC implementation, we follow a standard approach that allows developing software components to fulfill specific functionality. The proposed approach allows separating the architecture according to the system functionalities. Figure 9.1, shows the general TDG-control system architecture. It is composed of five main components: 1) "Connection Devices", which enables any user[2] (stakeholder) to connect to the TDG-control system; 2) "Authentication and authorization" mechanism intended to impose security layer of authentication of the user since that will authorize them to use "Core Services" and "Blockchain

[1]The developed system is called the TDG-control system.

[2]In the reset of the section, we may refer to stakeholders and any possible stakeholder representative with the term user.

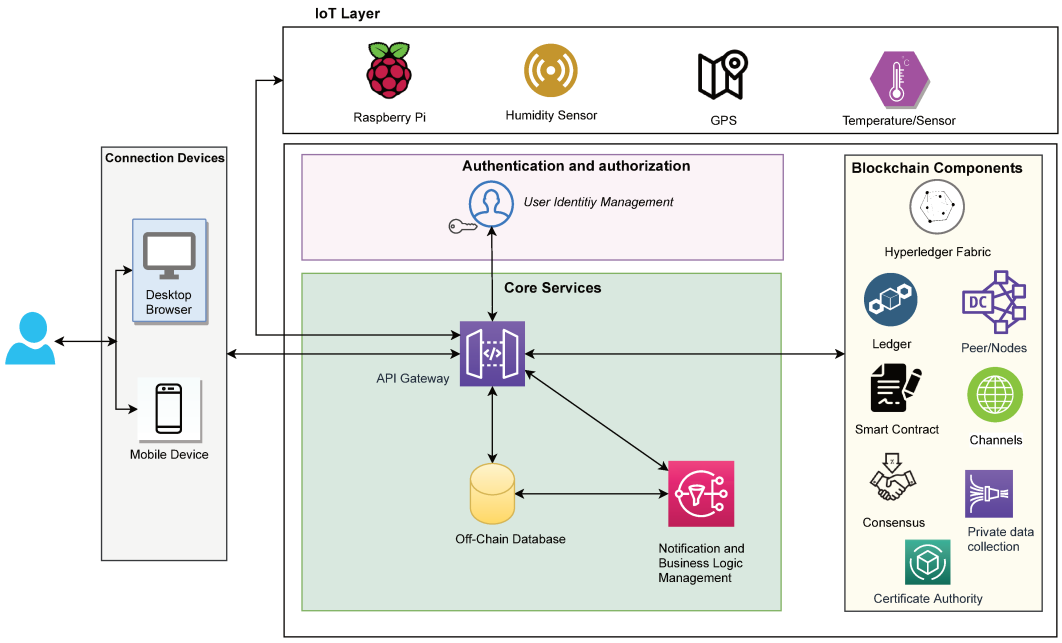

FIGURE 9.1: The TDG-control system architecture.

components". With this component, we intend to avoid any issue with user identity by requiring authentication at the interface layer with the help of an authentication mechanism, e.g., 2-factor authentication; 3) "Core Services", which enables the implementation of different services. It is composed of "API Gateway," which filters and redirects requests from users or external service to "Blockchain Component". The "Notification and Business Logic Management" is a component that enables the system to act according to its business logic which is not implemented in SC. It enables the implementation of the backend that helps managing notifications according to the user request. It also provides specific data and analytic (e.g., reporting and other matrices) for users. The component "Off-Chain Database" serves to store a large amount of data that are not necessary to be placed on BC, and also, it may retrieve data from BC for analytic purposes; 4) "IoT Layer" components present any involvement of IoT devices in the TDG-control system architecture; 5) The "Blockchain Components" presents the BC platform and its related components.

9.2.1 Blockchain Environment Configuration

- **BC platform selection**

  For developing PoC, initially, we refer to our design method (6) by recalling layer L5 called Platform Specific Model (PSM), shown in Section 7.7. This model is entirely related to a specific technological platform, including the BC platform. As indicated on the PSM (7.7), we selected Hyperledger Fabric (HF) BC platform to develop the PoC based on the TDG use case requirements. The general characteristics of HF are presented in Section 3.4.3, and through this section, we present HF technical details used for PoC development.

- **Technical specifications for PoC development**

Several technical dependencies are required for running this full BC node. Following, we present the required technical dependencies. We present some command-line interface (CLI) scripts for installing and configuring these technical components (dependencies) for the HF node.

- *Full BC Node Environment*
  - Server (nodes) characteristics:
    - VM: Ubuntu 18.4
    - RAM Memory: 10 GB
    - Processing Power: Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
  - Node.Js v10.18.0[3]
  - Docker Container v18.09.7 and Docker Composer v3.7[4] (Docker, 2019)
    ```
    sudo apt-get install docker-ce docker-ce-cli containerd.io
    ```
  - Hyperledger Fabric (HF) v2.3.1[5]
    ```
    curl -sSL https://bit.ly/2ysbOFE | bash -s - 2.2.2
    ```
- For the development of the user interface which allows stakeholders to interact with TDG-controls system, we applied:
  - User interface: VuJS[6] and HTML[7]

9.2.1.1 Network and Channels

To deploy the physical BC network, we refer to layer L4 (PISA, shown in 7.6.1) as we have specified the BC deployment architecture. We recall that to compose PISA (7.6), we used MT to design the architectural components. In this context, in the deployment phase, we specified: a) system digital components: *BC nodes/peers*; b) component inside digital system components: *ledger and SC*; c) relationship between digital system components: *channel*. We deploy a BC network comprised of nodes distributed geographically and managed on the premises of the stakeholders. Figure 9.2, shows the deployed BC nodes and the possibility of accessing the BC network through the user interface, i.e., "Blockchain User Portal".

The concept of the channel (3.4.3) in HL enhances privacy and confidentiality. In the context of TDG, stakeholder communication is sensitive and requires information security and confidentiality. To accomplish this requirement, we have composed a BC network that allows stakeholders to manage their communications and exchange information through their *private channels*. The channels are composed of peer nodes representing stakeholders. The *private channels* create sub-ledgers that are accessible only by the invited members. Using *private channels* stakeholders can exchange information privately, and only the channel

[3] https://nodejs.org/en/blog/release/v10.18.0/
[4] https://docs.docker.com/engine/install/ubuntu/
[5] https://hyperledger-fabric.readthedocs.io/en/latest/install.html
[6] The Progressive JavaScript Framework: https://vuejs.org/
[7] https://html.com/

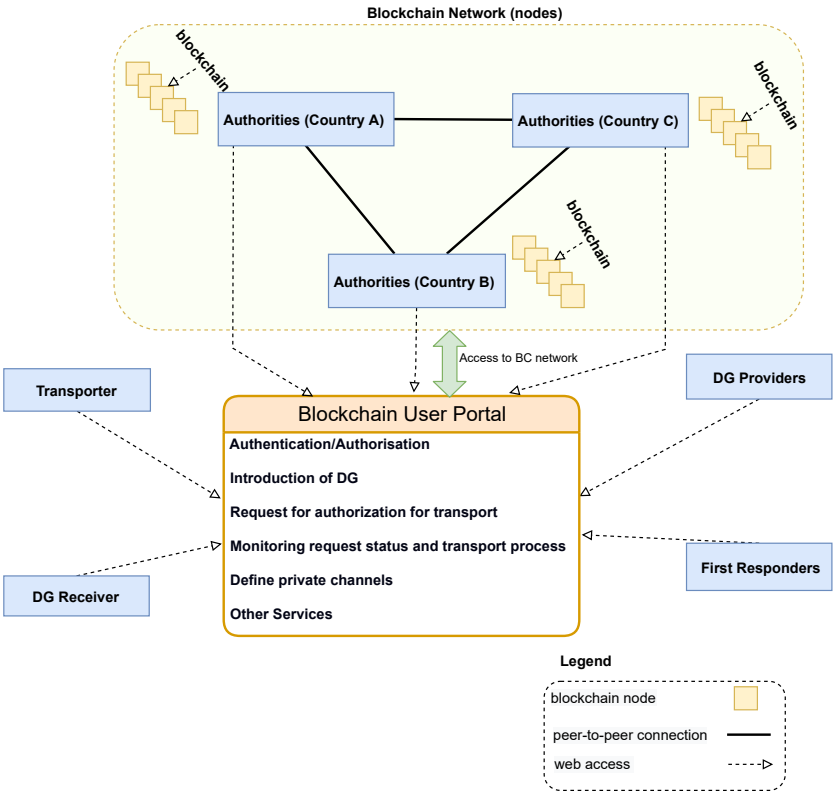

FIGURE 9.2: The access of blockchain network via web user interface.

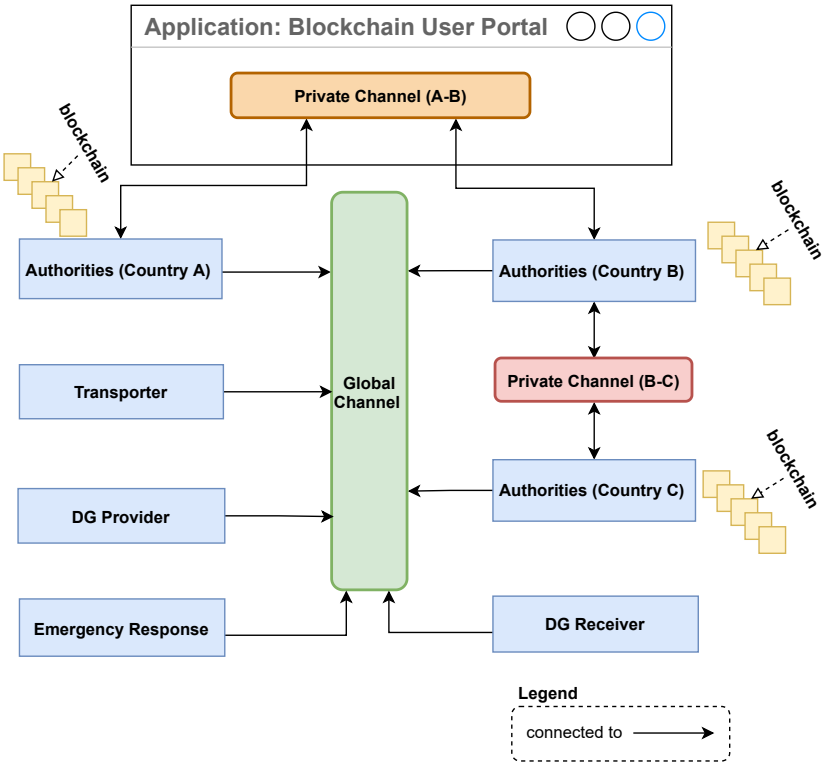

FIGURE 9.3: The structure of sub-networks (channels) in HF for TDG.

participants can access this information. This access is enabled by the application (or client). The sketch presented in Figure 9.3 illustrates the organization of channels. In the "Global Channel", all stakeholders are involved, and any information exchange is available for all participants. In the "Private Channel (A-B)", the stakeholders "Authorities (County A)" and "Authorities (Country B)" exchange information privately, and only they can see this information. This also enables the traceability of information for the involved stakeholders for any operation with DG. The CLI code and procedures for the creation of *private channels* are shown in Figure A.9.

9.2.2 SC Development and Deployment

This section shows the development of the SC based on the specification supported by our design method.

SC enables interpreting the business logic required in TDG, which further is implemented and deployed on the BC. As the main source for expression of the business logic in SC, we refer to the L3 (PISCM) (7.5), which provides the SC specification. The business logic expressed in the models defined in PISCM is express in the SC. Further, with the help of MT, we have defined the platform-specific smart contract model (PSSCM), shown in 7.8) (L6) for coding (translating) the business logic into SC. The PSSCM serves as an initial code template, and then we further might extend the code depending on the application needs. Our design method, by combining its layer, for example, PISCM (L3) and PSSCM (L6), allows the development of the SC code[8]. This may be achieved by mapping the algorithmic specification of the SC (L3) into the programming language choose to develop SC[9] (L6). Beyond the specified SC in the design method layer, we develop additional SC necessary to help manage the TDG process in combination with IoT devices.

Scenario 1: Before any activity with TDG, the stakeholders registration and validation is required.

For the scenario defined for the TDG stakeholder registration showed in 7.10, we specified the SC functionalities. The models showed in PISCM and their algorithmic expressions are the source of the SC development. Based on the PSSCM, we convert the algorithmic expression into SC code. Figure A.8 shows a simplified version of SC for stakeholder registration. The SC function validates each step of the algorithm for the registration of stakeholders.

Scenario 2: Instantiation of the meta-rules for stakeholder management and business contract.

Once the stakeholder is registered, particularly "Authority," they can instantiate meta-rules that help manage the process of TDG. The meta-rules, as shown in 8.2.1, helps in applying semantics on TDG by enforcing SC to react in compliance with the regulatory

[8]The automatic generation of the SC might be an ideal solution which further might allows direct deployment of SC into BC. This is possible with the development of appropriate tools which allow mapping of pseudo-code into JavaScript, as shown in (PseudocodeToJavaScript, 2020; PseudocodeJS, 2017).

[9]The SC codes are shown in GitLab: `https://git.list.lu/adnan_imeri/thesisaimeri/-/tree/master/SmartContractCodes`

framework or business contract terms and conditions management. To instantiate such rules, we use SC, named *"SC_Stakeholde_Mera-Rule"*[10], which enables storing of meta-rules into BC.

Scenario 3: Before starting any TDG operation, an authorization should be required from competent local authorities.

Based on the specification of the SC showed in 7.5.1, an authorization for any TDG operation is possible only when all the required information are completed. We use the algorithmic specification showed in 2 for developing *TDG_Process_Authorisation*[11]. This SC validates in real-time the information provided from stakeholder (requester) involved in TDG. It automatically shares the dossier once it is validated and notifies the "requester" for the final decision in authorizing TDG.

Scenario 4: For any DG that will be transported a registration on the TDG-control system is required.

The *SmartContract_DG* SC[12] allows stakeholders to introduce a new DG that is subject to transport. It determines the properties of the DG, which are prepared for transportation. The main attributes highlighted here are DG identification (class), type of DG, risk level (sensitive parameters, e.g., in high temperature, humidity, disturbance, or others), quantity and many others.

Scenario 5: For any process of TDG we collect any information related to it.

This *SC_DG_Process_Initialization*[13] offers one of the main functionalities TDG-control system. It initializes the process of TDG. For each TDG, an *identified (ID) process* will be initialized. This SC also informs the involved stakeholders about the starting of the process. This process (ID) remains open, and all interactions, for example, the exchange of information with authorities or between stakeholders will be identified with the process (ID).

Scenario 6: For better management of the Warehouse a SC collects and share information.

The *SC_DG_Warehouse*[14] gives the necessary information about the warehouse facility. A piece of information is the location of the warehouse, current capacity to host DG to be stored, information on the arrival date/time of DG, availability to maintain the state of the DG with the required level of safety conditions

Scenario 7: The cross-border with DG requires higher attention than usual goods.

To manage better the TDG and to improve safety while operating with DG, the *SC_DG_Cross_Border*[15] collects information in combination with IoT devices. It presents a checking point, when the truck arrives at the border of a country, and it automatically informs stakeholders, i.e., "Authorities" of both countries and also the "DG Receiver"

[10] https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/SmartContractCodes/SC_Stakeholde_Mera-Rule.js

[11] https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/SmartContractCodes/TDG_Process_Authorisation.js

[12] https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/SmartContractCodes/SmartContract_DG.js

[13] https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/SmartContractCodes/SC_DG_Process_Initialization.js

[14] https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/SmartContractCodes/SC_DG_Warehouse

[15] https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/SmartContractCodes/SC_DG_Cross_Border

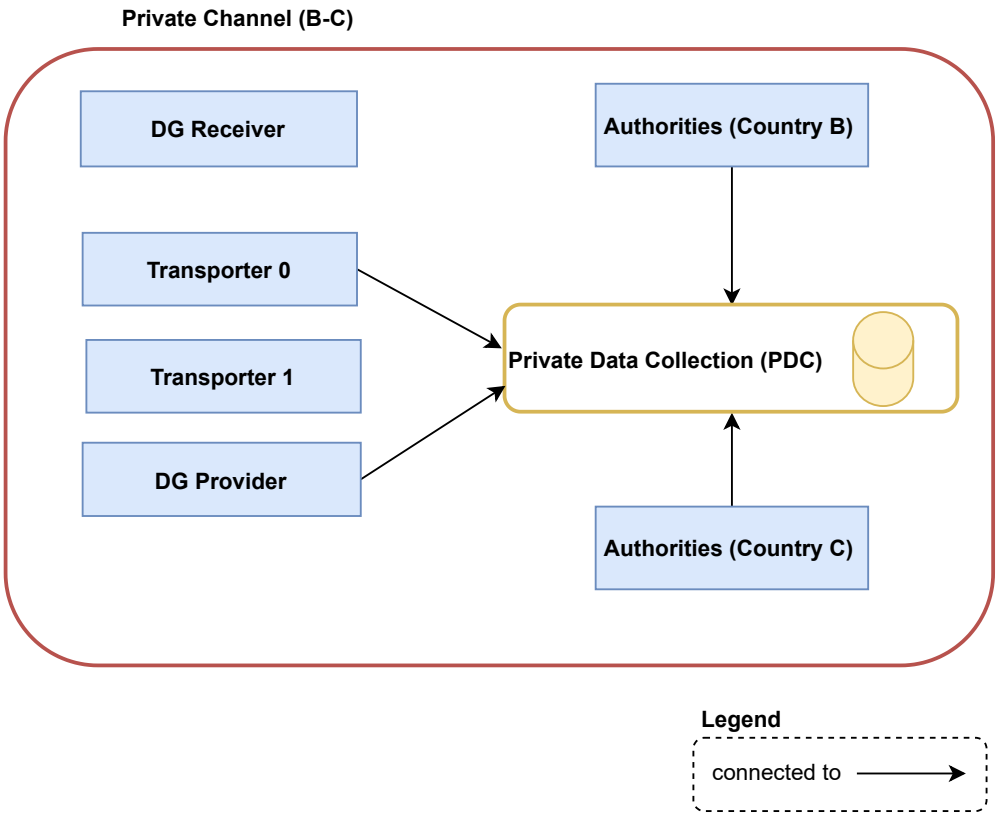

FIGURE 9.4: The example of private data collection (PDC) shared in HF channel.

Scenario 8: For managing the end-to-end lifecycle of DG.

The TDG stakeholders, particularly the competent authorities, require surveillance of DG movement across the geographic area under their jurisdiction in and cross-border context. The *SC_Digital_Certificate_DG*[16], collects all information from TDG in an end-to-end manners, based on the specification shown in 8.6.

Scenario 9: How to enable the exchange of private (confidential) information for stakeholders that belong to the same channel.

The private channels enable only communication between stakeholders (participants of the channels), and all information exchanged is available to stakeholders in that channel. In order for the stakeholders to exchange *private information* between them while being part of the same channel, HF offers the notion of *private data collection* (PDC) (Private-Data, 2019). The PDC enables sharing specific-private data inside channels that can be accessed only by stakeholders invited in PDC. This is the case when a group of stakeholders on a channel intends to keep data privately from other stakeholders that are on the same channel (Private-Data, 2019). Figure 9.4 illustrates the context of PDG shared in channel ("Private Channel (B-C)"). All stakeholders connected in PDC can access the privately shared information, except "DG Receiver". The PDC enables sharing of information at the stakeholder (organization) level. However, PDC does not support sharing private information within organization level,

[16] https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/SmartContractCodes/SC_Digital_Certificate_DG

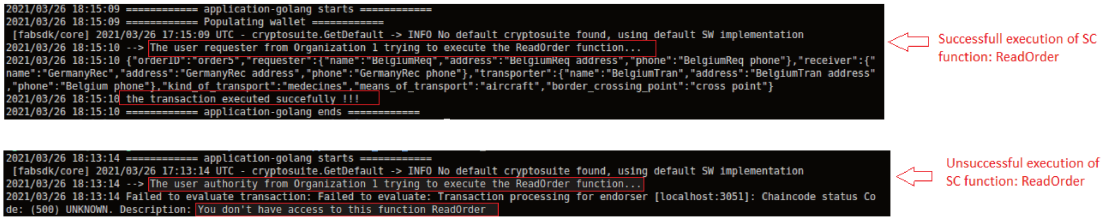

FIGURE 9.5: The results for the execution of SC guard access.

meaning that the "Transporter 0" cannot hide information from "Transporter 1", nor being able to hide information at the role level, e.g., "DG Provider" can read any information that the "Authorities" share in PDC. We consider this is an issue that needs to be addressed in the context of the TDG use case.

To overcome these issues and to enhance the privacy and confidentiality of the application that uses BC, we defied a conceptual role-based access control approach shown in 7.5.5. To support our conceptual approach, we developed SC to implement the conceptual approach that helps overcome access rights issues that the current design principles in HF do not fulfill. The implementation logic corresponds to the *"SC_Guard_Access"*[17] which reads the access right from the access policy (as shown partially in 9.1) and denies/allows any SC function to be executed by role (user) based on the defined access control policy. Figure 9.5 shows the example of implementation of the *"SC_Guard_Access"* that denies execution of SC function called "ReadOrder" when a specific role, e.g., "authority" does not have access right to the information that the function "ReadOrder" will show. The "successful execution of the SC function" means that the role has the right to execute the SC function "ReadOrder" and the "unsuccessful execution of the SC function" means that the role does not have any right to execute the SC function "ReadOrder".

```
var ruleTable RuleTable
ruleTable.Function = make(Function)
err = json.Unmarshal(tableAsBytes, &ruleTable)
if err != nil {
  return fmt.Errorf("failed to unmarshal JSON: %v", err)
}
var newCategoryList []string
var newMSPList []string
if ruleTable.Function[functionName] == nil {
  newCategoryList = []string{category}
  newMSPList = []string{organizationMSP}
} else {
  newCategoryList = append(ruleTable.Function[functionName].CategoryEnabled,
      category)
  newMSPList = append(ruleTable.Function[functionName].MSPOrganizationEnabled,
      organizationMSP)}
```

LISTING 9.1: The short representation of code for SC_Guard_Access.

[17]Source code of SC_Guard_Access:https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/SmartContractCodes/SC_Guard_Access

9.2.3 Template: Generation of Digital Twins Based on the Design Method

We consider the model transformation and the alignment with BC principles to efficiently design BC-based solutions and shortly deploy them in a specific BC platform. The proposed design method allows us to consider high-level concepts, i.e., sourcing from the regulatory framework, performing formal analysis that identifies concepts (stakeholders), performing workflow analysis, and specifying interactions (based on methods) between concepts. Finally, it enables defining low-level concepts, e.g., SC and digital twins (DT) (3.7).

Initially, the proposed design method (6) allows us to identify concepts and validate them against the regulatory framework. We use the layer of our design method, i.e., PISCM (7.5), associated with the platform-independent model (PIM) to express and verify the interaction of concepts (components) of the targeted system. Once the validation is performed, we are able to define (with the help of model transformation) the architectural components of the targeted system (TDG-control system) (7.6). Following, we use our methods layers, i.e., PSM (7.7) and PSSCM (7.8) that are associated with the platform-specific model to define technological components specific to the selected BC platform.

The concepts that we showed on the model (7.6) are now represented as a template (components). Any time we intend to instantiate the TDG-system control, we instantiate all these concepts. For example, if we have three "Authorities", then we instantiate three component-authorities, or if we have two "Transporters", we instantiate two component-transporter in the BC, and so on. This instantiation is performed through the user interface[18] (shown in 9.2.5), and any time someone creates a "Transporter", then there is a component created on the BC, and it will be specified by a set of information, and authorized operation it can perform (as specified in L3). That component presents the real "Transporter" (or other stakeholders in case they instantiate a digital registration of their profiles). The specification of stakeholder registration is presented in L3 (1). For any real "Transporter", we have on the BC its DT (7.6), which is an associated set of information and authorized operations. Figure 9.6 shows a reflection of real stakeholders into DT. The defined DT (component) is entirely based on the regulatory framework; thus, all its activities are in compliance with the regulatory framework specifications. The interaction's specification (authorized operations) indicates DT activities before, during, and post TDG with the help of SC. The authorized operation are strictly maintained by business rules expressed in SC. To maintain the business rules sourced from the regulatory framework for TDG, we have formally specified and verified SC with the help of temporal logic. This enables the alignment of DT with regulatory frameworks.

Figure 9.7, presents a simplified example of authorized operations (shown in green boxes) of DT for the "DG Provider" and "Authorities". The DT of "DG Provider" has authorized operations such as *"Complete Movement Doc"*, *"Apply for TDG authorisation"*, *"Send dossier"* and many other operations specified according to activities of "DG Provider" in TDG (specified in L3). This **DT presents a template** for the "DG Provider", and we can create different instances of "DG Provider" by using this template, which will react according to the authorized

[18]For instantiation and generation of DT, an automatic process would be ideal. That requires additional technical development of tools, and it is out of the scope of this thesis.

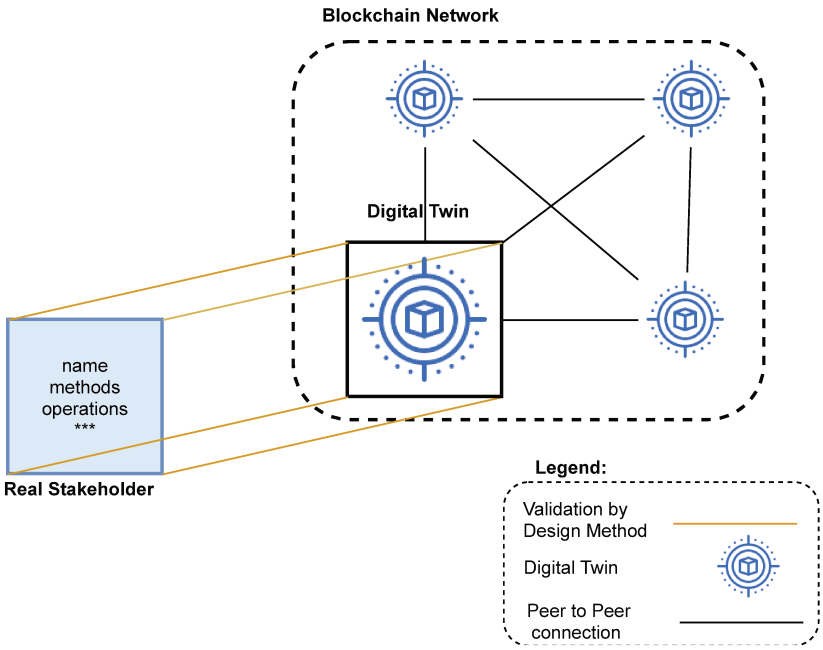

FIGURE 9.6: The conceptual presentation of real concept (stakeholder) into digital twin.

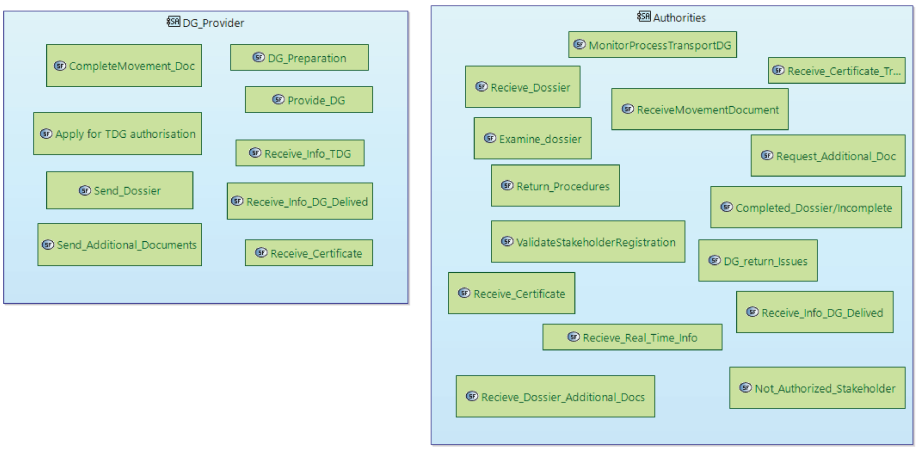

FIGURE 9.7: The authorized operation for the DT of "DG Provider" and "Authority".

operations. The authorized operations are expressed in terms of the function of the SC. Any activity on the TDG-controls system is performed based on these operations expressed on DT. The DT performs authorized operations on behalf of its real twin (stakeholder). Similarly, the DT of "Authorities" perform authorized activities as intended by the "real" twin. In Figure A.6, an extended version of DTs and their main authorized operations in TDG.

The generation of DT enables equivalent management of the TDG among many countries and stakeholders. The DT generated are in compliance with the regulatory framework applied in different countries, e.g., France and Luxembourg. For example, we consider the use case of transport and management of MW, based on a regulatory framework ((Commission, 2008)) applied at the national and international level. In this way, the generated twins and their operation comply with this regulatory framework; thus, facilitating management of TDG at the international level.

9.2.4 The Improvement of Trust and Transparency in SpCDG

At the highest level, the proposed design method enables the generation of the DT of the stakeholders and maintains the dynamic aspect of SpCDG. The DT is defined based on high-level specifications proposed by our design method and further is transformed into an operational component in the BC. This approach enables the improvement of trust in the SpCDG. Initially, this allows the certification (valid registration against regulatory framework) of any involved stakeholder in the TDG. For the static components (digital twins), acting in compliance with the regulatory framework enables correct and compliant operation with DG. The TDG-control system automatically validates concepts, processes, relationships based on the specification shown on the design method and the meta-rules instantiated from the involved stakeholders. Furthermore, information security based on the BC principles facilitates sharing of information among involved stakeholders. To highlight the information management from the privacy perspective, we designed and developed a PoC solution to manage data access control based on roles (9.4).

Another aspect of being highlighted is the ability to enhance transparency in the SpCDG. Our design method enables the identification of stakeholders, specific information for DG, which is under transport, current warehousing, and other related information. To fulfill these features, we proposed and implemented the concept of *Digital Certificate* as shown in 8.6.

9.2.5 User Interface and Scenario Implementation

As the use case in this thesis comes from the real world, the user interface is inspired by the official "paper-works" forms that are also the natural way the TDG works. This indicates, we digitally include any possible component that already exists in the current real-world use case. Figure 9.10[19] illustrates the main components of the user interface. For accessing the application, the stakeholder should log in with its credentials (as one main component shown in Figure 9.1).

Scenario UI 1: Stakeholder registration from the user interface.

[19]Initially, it is empty since we do not have performed any action.

In case the stakeholder is not registered already on the TDG-controls system, the login is denied, and prior registration is required (as described in L3 of PISCM (7.5.1, and specified in Scenario 1 in 9.2.2). Figure 9.8, illustrates the stakeholder registration web form. Recall the L4 (7.6) of our design method, the stakeholders are considered as DT, and any interaction with the system is possible only through DT. The presented web form allows stakeholders to provide specific credential information, and some of them are automatically validated. For example, the national "NationalID", presents the national identification number of any stakeholder. The SC for stakeholder registration automatically verifies this number by reading from the "off-chain" (as shown in TDG-control system architecture 9.1) database, which contains national information for stakeholders. In the registration web form, a drop-down list allows the selection of the type of stakeholder, e.g., "Authority", "Transporter", etc. Furthermore, the registration form allows additional options for the stakeholder when particular events happen during the TDG. These events, e.g., event *CODE: 600*, indicate cross-border arrival of DG. Then stakeholder may provide a specific link (API-POST) where this information will be posted. This enables automatic notification of stakeholders by posting information on their provided URL (API-POST). The posted information helps stakeholders increase DG arrival attention, thus prepare to warehouse DG or taking similar action to host DG. The event *CODE: 700* indicates delays in transporting DG. We specified time constraints and management TDG process where time should be considered in Section 8.3. In this situation, the stakeholder will continuously receive such information, and they may react accordingly. Moreover this is to help to avoid the situation where the "DG Receiver" premises are closed while the DG is on the route to arrive at that destination. The event *CODE: 800* indicates emergency situation. For example, during the TDG, if the temperature of DG is exceeding the normal threshold, the competent stakeholder is notified (as implemented in Section 9.2.10). The successful registration of the stakeholder generates its DT (9.2.3) on the BC. The DT from the BC perspective is presented as a "block", and the stakeholder interacts with DT from the user interface or integrated API Gateway (9.2.7).

Scenario UI 2: Stakeholder accessing (login) to TDG-system control. Definition of meta-rules for TDG process governance and business contract management.

The successful login (9.10) ensures accessibility to the core services to interact with the BC through DT. Once login, the TDG-control system shows another web form that allows the expression of meta-rules (by "Authorities") and meta-business rules [20] (by "Transporter", "DG Provider", and "DG Receiver"). The intention of this web form is as follows:

a) Once the user (stakeholder) is login as an "Authority", a specific web form[21] becomes available. That presents the opportunity for the "Authority" to express specific meta-rules (8.2.1) to govern the process of TDG with additional strict rules. The "Authority" is considered as the host of the BC network, and they initially have privileges to add specific meta-rules to govern the TDG process. For example, a kind of meta-rule indicates that for a particular

[20] We refer with term "meta-business rule" for expression of business preferences of the stakeholders. This also intends to distinguish from the term "business rules" defined in Section 8.12.

[21] The web form for meta-rules and meta-business rules can be adapted by providing different categories on meta-rules (meta-business rules) and providing a large scale of the text box, drop-down lists, and checkbox. The information needed to fill the web form originates from an "off-chain" database.

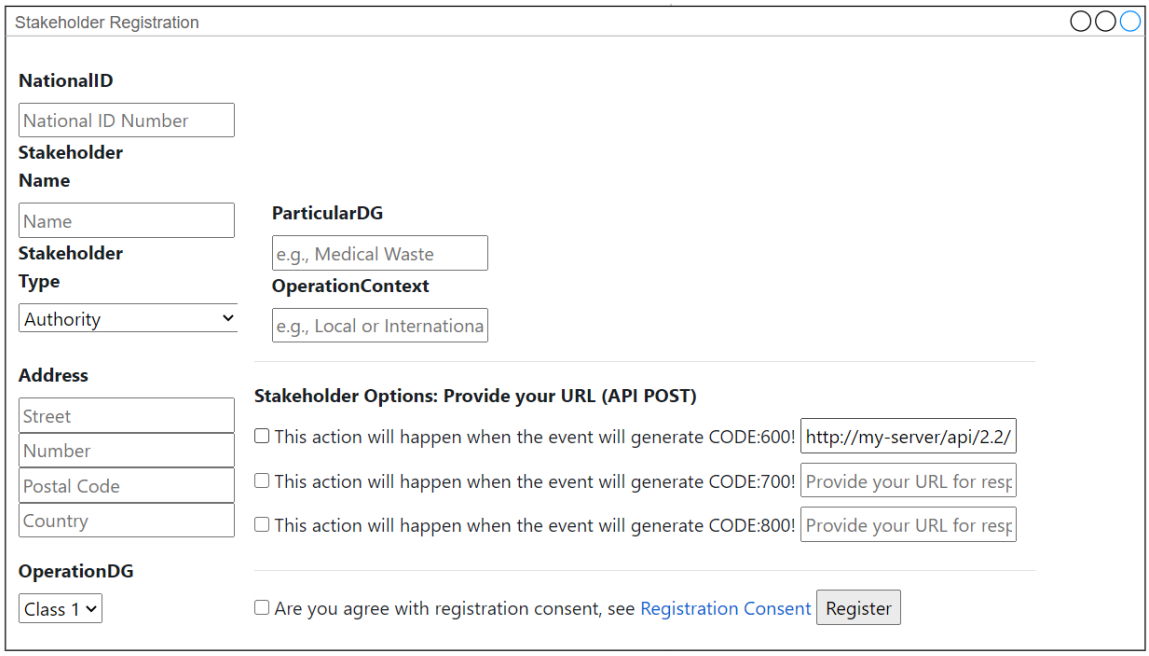

FIGURE 9.8: The user interface for stakeholder registration.

DG, the transport is possible only in the specific time frame, e.g., departure, not before 10:00 AM, and arrival not after 18:00 PM. Also, for specific DG, there should be at least three crew members, or specific DG is strictly not allowed to enter a specific country or location. Also, there might be a situation where specific DG treatment needs to be done within one month or meta-rules to highlight the specificity of DG, e.g., Nuclear Waste, and strict conditions on validation stakeholder transportation capabilities. Figure 9.9 shows and simple example from defining meta-rules. This web form might be extended and filled by using "off-chain" databases and further storing this information "on-chain" with the help of "SC_Stakeholde_Meta-Rule".

b) If the user (stakeholder) is login as "DG Provider", it has the possibility to express the meta-business rules applied toward prospective business partners. For example, the

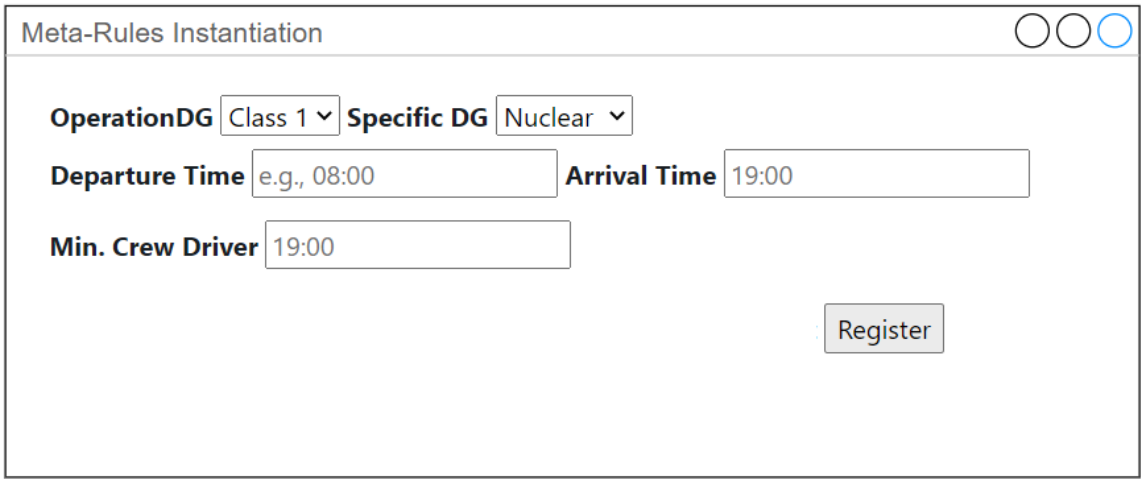

FIGURE 9.9: The example of expressing meta-rules for TDG.

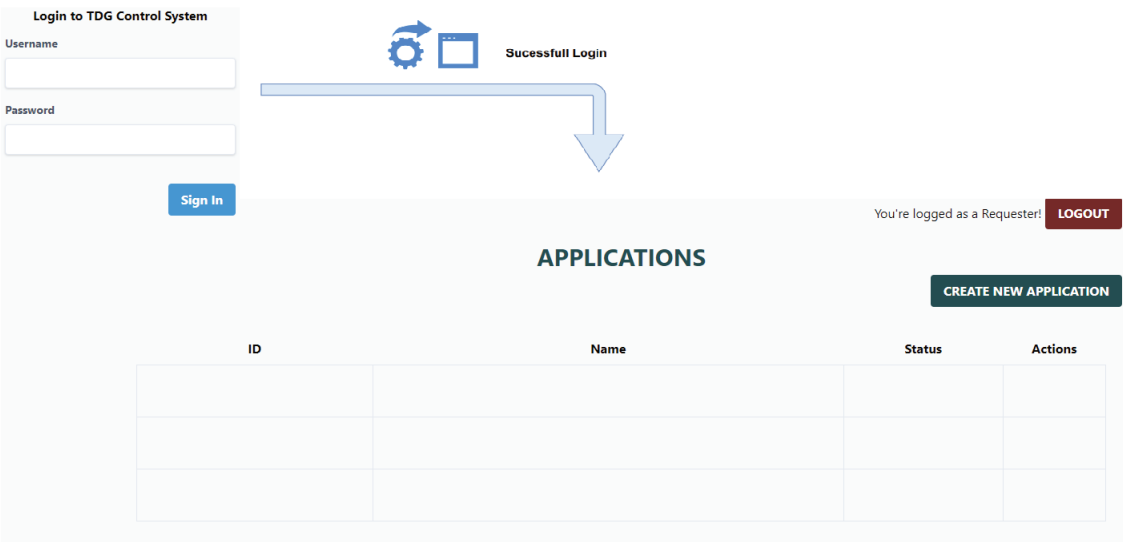

FIGURE 9.10: The user interface for accessing the TDG control system.

"DG Provide" among the specific meta-business rules are i) The "DG Provider" can provide DG only in the specific time frame, e.g., once in two months. Another example might be by showing business preferences. ii) Only "Transporter" located in Luxembourg or France must be selected to transport DG. Similarly, the "DG Receiver", expresses the general meta-business rules as opportunities in terms of capacity to treat DG: i) Determine what kind of DG treatment they perform, in what capacity, and for what time; ii) The preferred time for receiving specific DG and many other related rules. Also, the "Transporter" can express the general meta-business rules: i) Determining the type of DG it can transport; ii) Specific condition that it may require for transporting specific DG; iii) availability (free schedules) to transport DG in a specific time; iv) Determine the specific destination that it does not transport; v) Present a list of associated business partners that operate similar TDG.

The instantiated meta-rules from a business perspective might change continuously, and the stakeholders can update them by using web forms that further transmit these changes on BC (writes new transactions). Remarkably, applying meta-rules, especially from the business perspective, enables us to define an eco-system that would serve the concept of "Multi-Party Smart Contract" shown in 8.10.

Scenario UI 3: Request for authorization to transport DG.

The access to TDG-control system is possible for the stakeholder according to their profile (DT). To present an example, we distinguish two different user- roles, the "requester"[22] and "authority". The role of the "requester" is the stakeholder who requests authorization for TDG. The "authority" user-role authorizes or denies the request and monitors the TDG process. Once the login is completed, the stakeholder (requester) can create an application (**button**: create a new application) and fill in the information that is intended to describe the request for TDG.

For an illustration of the user interface, we present a scenario, which shows the required information for the request-authorization in TDG. In the application form, several pages

[22]The user-role of the requester can be "DG Provider" or "Transporter".

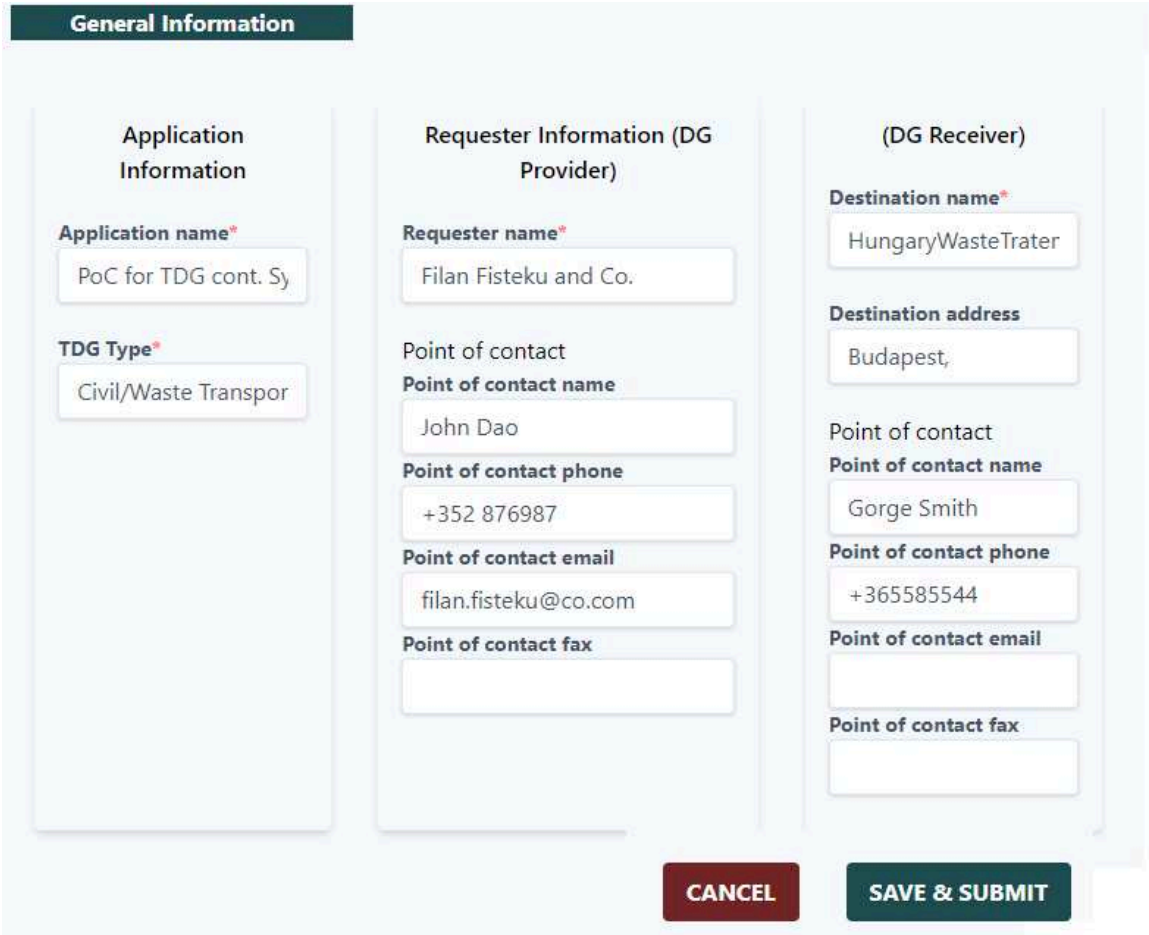

FIGURE 9.11: The web form for general information to request authorization to TDG.

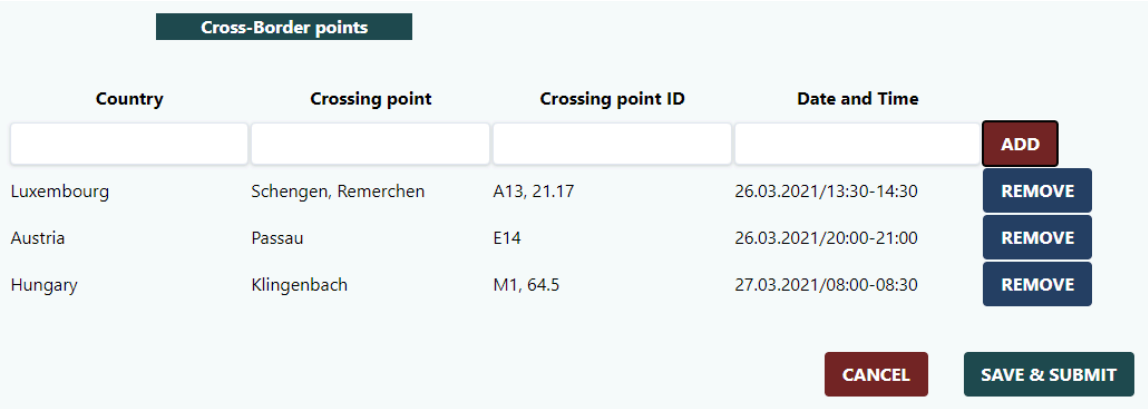

FIGURE 9.12: The identification of cross-border points for TDG.

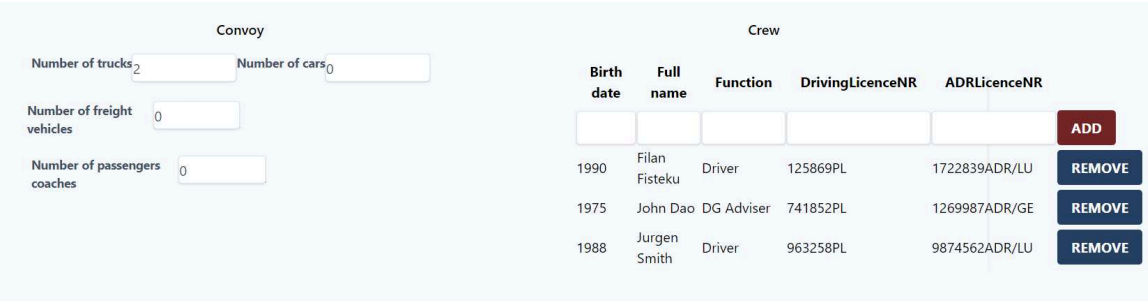

FIGURE 9.13: The convoy and crew details for the TDG.

allow adding information for the request to TDG. First, the page "general information" allows providing general information for the application such as the application, the requester (DG Provider), destination (DG Receiver) as shown in Figure 9.11. Secondly, another web form allows defining the cross-border points (name) that determine from which country the convoy with DG enters another country as illustrated in Figure 9.13. Thirdly, the requester determines the exact cross-border points and date, and approximate time. Recalling our design principles showed by our design method, the time plays significant role on managing TDG (as we have specified in Section 7.5.2, 7.5.4 and 8.3).

Once this information is filled in all web forms[23]) enables pushing the button "Save & Submit" in the final web form. Over the provided information, an automatic validation is performed by SC. This validation is applied based on the specification of the SC (shown in 7.5) for authorization of TDG and meta-rules that govern the TDG. The validation process performs a detailed verification of the application by considering time-related constraints, geographic constraints (as shown in Chapter 8), and other related constraints. This validation is performed based on the authorized operation of DT of "Authority". If the request does not pass the verification phase the TDG-controls system will immediately refuse the application (status: REFUSED) and submit it to the authority to notify them for the current activities. If all verification steps pass, the TDG-control system submits (status: SUBMITTED) application to the authorities for the final approval as shown in 9.14. This submission automatically

[23]The large-sized documents are submitted through the platform while hashes of these documents are stored on BC.

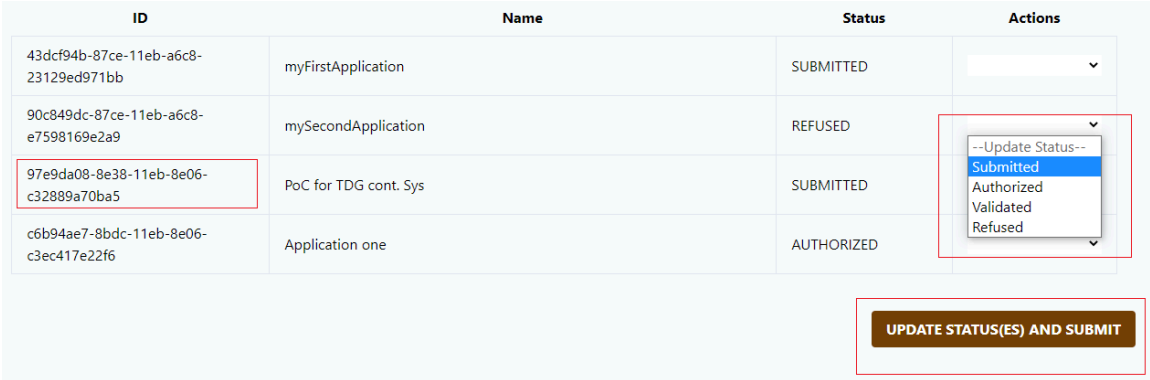

FIGURE 9.14: The user interface for "authority".

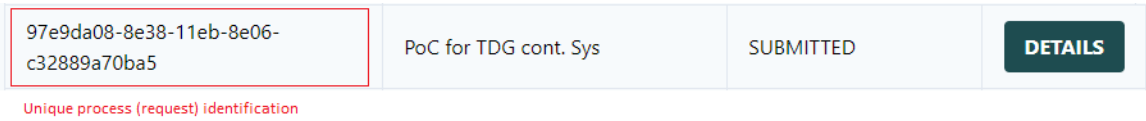

FIGURE 9.15: The process (request) identification for TDG.

involves the competent authorities of transit and destination countries involved in the request for TDG authorization by transferring relevant information to them.

In any request-authorization event there is generated a unique identifier (with the help of SC_DG_Process_Initialization) for the submitted application as shown in Figure 9.15. This identifier corresponds with "process ID" specified in Section 7.5.2, and it identifies any interaction for this process by involved stakeholders.

Scenario UI 4: Accessing TDG-control system with "authority" account.

Even those many pieces of information are validated automatically, in the TDG sill authorities must be aware of the request and give the final approval on it. When the stakeholder logs in as "authority", it can explore the request for TDG, ask for additional information (including transit and destination authorities) and finally approve (status: AUTHORIZED), the request, or refuses (for other certain reasons, i.e., political reasons) as specified in Section 7.5.1. Figure 9.14, illustrates the "authority" user interface and their actions toward the TDG requests. The **request** is considered to be authorized once all involved authorities "AUTHORIZE" the request.

The digital twin activities

As we can notice from the elaboration of scenarios above, the "requester", moreover its **DT** can perform only the activities specified for it at the design layer. Basically, it can provide information for TDG and interact with the TDG-control system accordingly to exchange information. Still, it cannot authorize TDG or see information for the other stakeholders. These are precisely the operation what the **DT** of "requester" is allowed to perform.

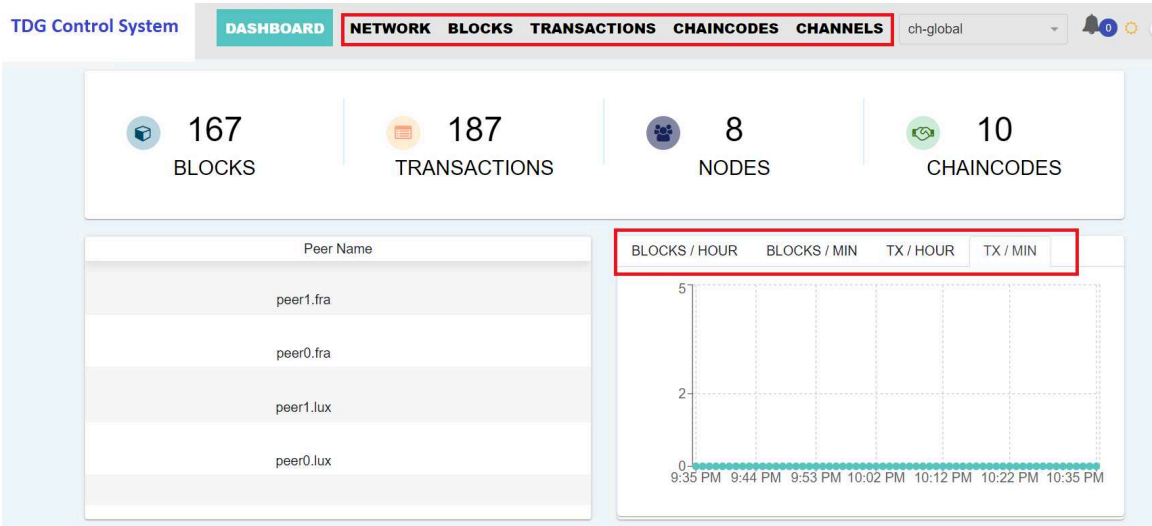

FIGURE 9.16: The exploration of BC components used in the PoC.

9.2.6 Efficiency

The digitalizing of processes and automatic validation of workflows enables efficiency in the TDG process. The TDG-control system facilitates real-time validation and sharing of digital information between involved stakeholders. This information is stored and exchanged based on the BC technology principles, thus avoiding concerns on data security[24]. The TDG workflow is mainly validated automatically with the help of SC. The proposed system performs an automatic check and validates information, processes, and interaction of digital components on the BC. The involvement of IoT devices reduces human involvement and aims at avoids possible mistakes.

9.2.7 Interoperability

From the perspective of the digital twin, once it is validated and deployed on the BC platform, it is compatible with all its authorized operations related to TDG at the national and international levels. That avoids any further technological development for the involved stakeholders. For example, "Transporter" does not need to develop any components or deploy any infrastructure that would allow being part of the TDG-control system. Furthermore, we provide a single web-interfaces, enabling access to the TDG-control system, which further allows stakeholders (digital twins) to perform operations according to their involvement in TDG. This access is enabled from the web interface or by using the API gateway. The proposed web interface might be limited to perform specific activity from the stakeholder; thus, we propose using an API gateway to access and interact with the digital twin in the BC. As shown in Figure 9.1, the API gateway enables communication with BC components, and it can be integrated with the stakeholder's application. The stakeholders will interact with their digital twin from their own application. Figure 9.17, shows a simplified schema of the

[24]This depends on the solution organization and managing the well-known BC security challenges as shown in 3.4.4

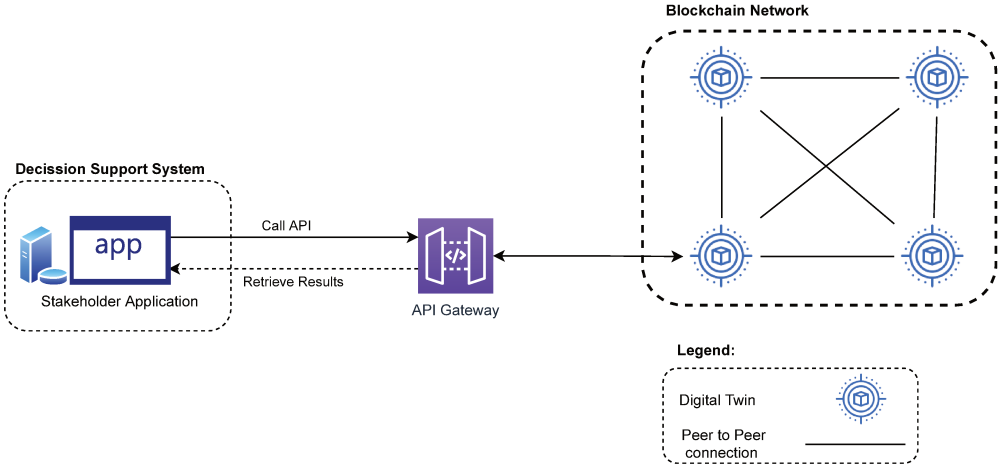

FIGURE 9.17: The interaction of TDG-control system with stakeholder application (DSS).

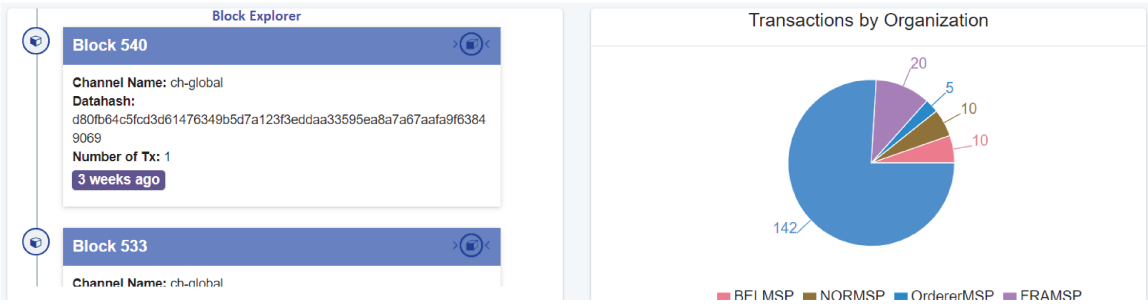

FIGURE 9.18: The exploration blocks and transaction performed by stakeholders (Organizations).

interaction of the stakeholder decision support system (shown in 2.3.1) (or other stakeholder-owned systems) with the digital twin deployed on the BC. Then, the digital twin will verify the authorized operation before any execution, thus performing authorized actions.

9.2.8 Exploring and Monitoring the Blockchain Components

For any successful action (event) performed in the user interface, a transaction is stored on BC. HF allows us to explore all events and also BC components stored and managed on the BC. Figure 9.16 shows the main components of the BC used for the implementation of the PoC. It allows us to explore the deployed network, the number of nodes, current blocks, transaction exploration, channels, and the current SC (chain code) deployed. Furthermore, it allows exploring the stakeholders' activities by showing the number of transactions for each of them, as illustrated in Figure 9.18.

9.2.9 Blockchain and IoT Integration

In this section, we show the main components used for BC and IoT integration as one of the essential technological components of TDG-control system architecture, as shown in 9.1.

In our approach, we use sensors to capture the environmental data such as humidity and temperature and measure the disturbance. The RFID is used for object identification purposes (e.g., identify trucks and other objects inside trucks). GPS tracker devices monitor the location of trucks. For performing small computing calculations and storage, we strive to use Raspberry Pi. Smartphones (or tablets) are used to monitor the process and interact with the TDG control system as required by the process. The conceptual approach for integration of BC and IoT is shown in Section 8.4, and an extensive study on implementing this PoC is shown in (Imeri et al., 2020b).

9.2.9.1 The used IoT Devices

This section presents some technical information about the implementation of the proposed solution. In this initial PoC implementation, we have integrated all layers of the proposed approach (9.2.9). In addition, we have used real IoT devices to simulate the end-to-end scenario for TDG in a lab environment.

The technical environment to support the IoT and BC integration is composed of the two following components (corresponding to L3 and L2 of the defined approach shown in 9.2.9, while the L1 is shown in 9.2.1):

1. *Lightweight BC Nodes*

   For this layer, we have chosen embedded devices with enough available computing resources to act as transaction validator, data aggregator, and data transmitter:

   - Raspberry Pi4[25]
     - Operating system: Raspbian
     - Communication protocol Bluetooth (or Zigbee)
     - Long Range Communication: 3G, SigFox
   - User interface: VuJS. HTML and JavaSCript.

2. *IoT devices Nodes*

   - Wireless Sensors
     - Environmental data:
       - Temperature and Humidity: HTU21D
     - Disturbance data: HC-SR04
     - Location: GPS coordinates
   - Mobile phone/tablet (Android or iOS)
   - RFID Readers
     - Barcode Reader
     - RFID tag Reader

[25]The minimal requirement is Raspberry Pi3 B

In (Imeri and Lamont, 2019), we showed a detailed technical report for the implementation of BC and IoT approach. The results from this implementation are extensively presented in (Imeri et al., 2020b).

The technical features of our BC and IoT approach enable storing immutable information. That information is treated *if and only if* it is retrieved from authenticated low complexity IoT devices that are already *"certified"* (8.4) and registered in the deployed system (BC full node). Furthermore, we use other types of more powerful embedded devices, (i.e., Raspberry Pi), which beyond signing transactions, they also act as *data aggregators*, in case the connection with the application layer, i.e., *BC Full Node* is temporarily lost. We argue that this capability to *aggregating data* significantly improves the *availability* of the system since generated data from IoT never vanishes. In the application layer (L1), we use an appropriate BC platform, i.e., HF, which supports and satisfies the performance requirements for such use cases. The configurable architecture of HF enables efficient data transmission, and scalability is assured.

The design principles used for the proposed use case determine that the data flow, monitoring aspects (traceability) of the movement of DG, are under the surveillance of the stakeholders. In such a scenario, information security, privacy, and confidentiality properties are highly required by stakeholders. This process follows at any time the requirements of the stakeholders and is always in compliance with the regulatory framework of the countries where the DG is transported at any time. This is achieved through the specification of customized SC to capture and execute the operations that are needed to make the process compliant with the stakeholder requirements and regulatory framework. For example, the case of notifying *stakeholders* when DG is crossing border incorporates privacy and confidentially issues (according to stakeholder requirements and regulatory framework). In this situation, this information is only sent to the relevant stakeholders, e.g., "Authorities" and authorized stakeholders.

9.2.10 The Temperature Checking Use Case Implementation

This section shows the implementation of dynamic parameterization of SC as a solution to overcome the problem of maintainability of SC, as shown in Section 8.11. This implementation is supported by BC and IoT integration showed in Section 9.2.9. The related research, results, and evaluation for this research are shown in (Imeri et al., 2019e). To perform this implementation the Hyperledger Composer[26] v0.20.4 over HF v1.4.1 is used. The technical report and the entire solution is accessible on GitHub (Imeri, 2019)

Scenario IoT 1: Dynamic parameterization of SC for managing temperature parameter during transportation of DG. There are three different cases to manage:

IoT 1.1: No problems occurred. This indicates, during the transportation, the parameter *TemperatureExceeded* is not exceeded.

IoT 1.2: Minor problems occurred: This indicates, during the transportation, the parameter *TemperatureExceeded* is shortly exceeded.

[26]Composer is now deprecated but still usable, next tools getting the succession and doing quite the same are named Convector & Hurley.

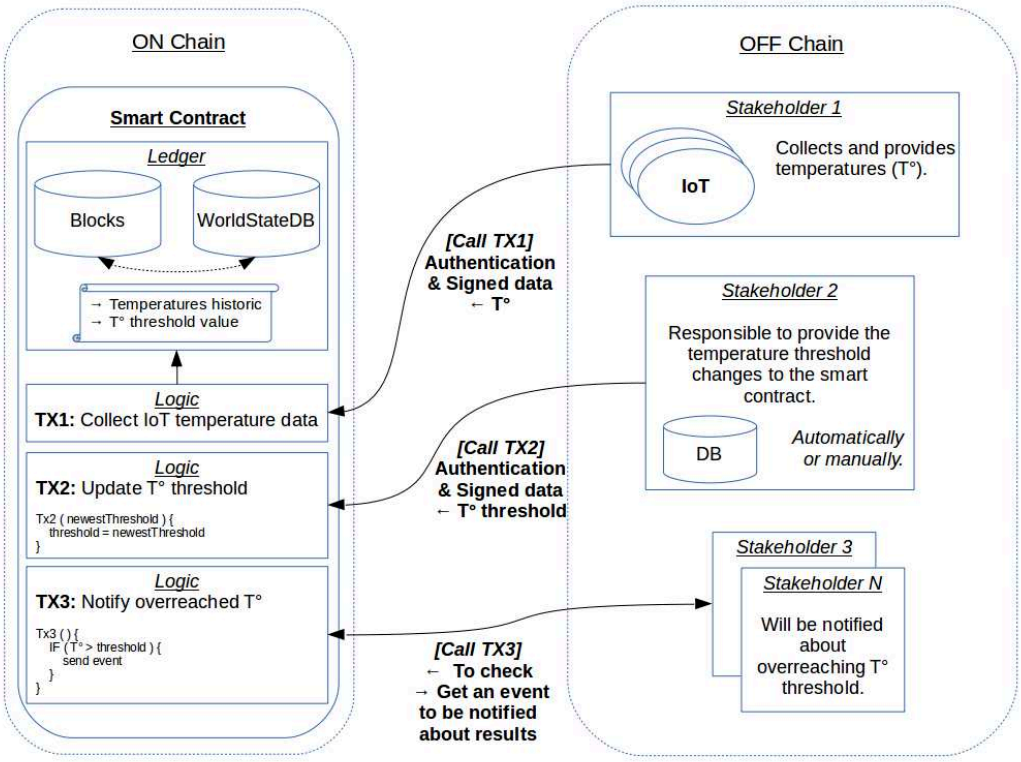

FIGURE 9.19: The conceptual model for illustrating the implementation of temperature checking use case.

IoT 1.3: Major problem: This indicates, during the transportation, the parameter *TemperatureExceeded* is exceeded entirely and for a specific time.

The implementation model is shown in Figure 9.19. On the left side of the model, there are presented "On Chain" components of the model, while on the right part of the model, there are presented components that support BC solution. For developing this model, three SC for performing the necessary functionalities expressed in transactions, e.g., *Collect* (TX1), *Set* (TX2), and *Notify* (TX3). The *DynParam* and *Temperature* are defined as assets (Assets, 2021) and also *TemperatureExceeded* as an event to notify stakeholders when the temperature threshold is exceeded. The *Collect* transaction serves to collect the last temperature value from the sensor. *Set* is used to define and change the needed dynamic constants. In our case, it is unique and named "Threshold_Tabc", which is hardcoded in the *Notify* transaction code to avoid users the need to know the name of this parameter. The *Notify* transaction serves to check if the temperature is exceeded the threshold (DynParam) defined by the *Set* transaction. If that happens, *Notify* will generate and send the *TemperatureExceeded* event to alert the competent stakeholders that are allowed to read that event. Thus, *Notify* is illustrating the usage of the dynamic constant through access to "Threshold_Tabc". So, an update of the threshold does not require an update of the *Notify* thanks to the *Set* transaction and the use of assets.

This implementation helps in managing the emergency situation as shown in 8.7. Once the *Notify* transaction is generated, it invokes particular SC called *SC_Alert* which collects specific information and sends them to a relevant stakeholder. This SC alerts stakeholders

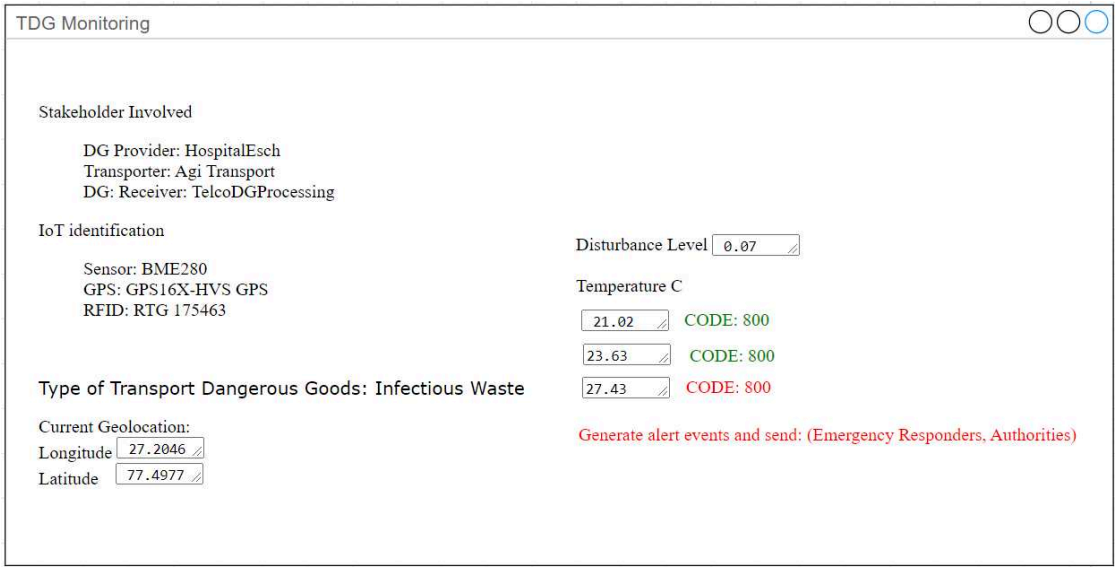

FIGURE 9.20: The web form for supporting the TDG monitoring (Process ID: 97e9da08).

when an emergency situation arises. SC_Alert is invoked:

- when the risk parameters are matched, e.g., high temperature and an accident probability is high (Scenario IoT 1.3).
- when the accident has happened (detected by a combination of information from sensors of temperature (humidity) and disturbance), the emergency alarm should be immediately sent to the responsible stakeholders, e.g., "Authorities" and should trigger "Emergency Responses" (Scenario IoT 1.3).

The stakeholder will receive notification in the provide link (as shown in 9.8) and/or directly on their devices (mobile phone, tablet or other monitoring devices). The *SC_Alert* is not invoke for the scenarios IoT 1.2 and IoT 1.1. A simplified specification of SC_Alert is shown in Listing 9.2. Figure 9.20 shows a set of information monitoring of TDG collected during TDG. The alerts showed in this web form are generated according to the scenario presented above.

```
1   //S0:@parameters:
2   var stakeholderList,
3   var IoTDeviceList,
4   var SubstanceList,
5   var typeOfDG,
6   var location,
7   var timestamp,
8   var riskLevel,
9   var driverInformation
10  var disturbanceMeasure,
11  var disturbanceLevelWave,
12  var TempLevelSubstance,
13  S1: function (if ReceivedTransaction in IoTDeviceList)
```

```
14      S2: function (if typeOfDG in
15    SubstanceList) and
16    ((TempLevelSubstance
17    >=TemperatureExceeded) or
18    disturbanceMeasure >=
19    disturbanceLevelWave)
20    S3: function (if stakeholder in stakeholderList)
21      S4: function (sendMessage: Alert (location,timestamp,typeOfDG,
          driverInformation )  -> stakeholderList[stakeholder])
```

LISTING 9.2: The short representation of code for SC_Alert.

9.3 Conclusion

In this chapter, we showed PoC implementation. For composing the PoC, we referred to the proposed design method, which further allows defining the components of the PoC. Initially, we showed the technical architecture of the TDG-controls system. Later based on that, we discover its associated technological components. The BC platform selection, its technical dependencies, deployment, and configuration are further shown. For developing a TDG-control system, we define the BC network, private channels, private data collection and implemented role-based access-control. The SC plays a significant role in the TDG-control system, and therefore we showed the development and deployment of SC. The user interface gave stakeholders the possibility to interact with BC easily. We implemented and explained a simple scenario to highlight the functionalities of the TDG-control system. Furthermore, the integration of BC and IoT enhances the TDG-controls system by presenting means for monitoring the TDG and reacting against adverse situations.

Chapter 10

Conclusions, Perspectives, and Future Research Directions

This thesis addressed the general problem of safe, secure, and efficient transport of dangerous goods (TDG), such as gases, explosives, nuclear materials, waste, medical waste, pharmaceutic products, fuel, acids, etc., that are treated and used by the public (or private) enterprise, and in some cases, by the military. The studied use case is from a real-world instance, which allows us to identify all the necessary requirements, gaps, and issues on the management of TDG (i.e., Medical Waste (MW)), and further to propose our perspective to improve the workflow of such process by proposing scientific approach and researching in cutting edge technologies such as BC and IoT. Among the primary motivation in this thesis is the improvement of trust and transparency in the process of TDG (i.e., MW). To achieve that, we proposed a scientific design method for designing software systems, including BC-based systems.

Designing and developing a BC-based solution is challenging since it is one of the most evolving technologies, and the best practices and standards are limited. The design principles proposed for specific BC platforms are not relevant for the other BC platforms. Using any top-down approach for developing BC-based solutions by gathering and considering stakeholder requirements and developing a straightforward solution leads to the risk of the non-standardized moreover not completed solution. Further adaptation of new requirements might impose enormous challenges.

Thus, in keeping with designing a BC-based solution, we propose a design method based on so-called model-driven architecture (MDA). It defines a new architecture-based method for designing software systems, including BC-based systems. The method enables the definition of different models as part of the entire architecture of the targeted system. For defining the system components, we use model transformation. The model transformation servers as a core component of our design method, and it presents one of the main contributions from this thesis. It allows dynamic transformation of models, consequently enabling discovering targeted system component interactions and examine the targeted system's inner behavior. The proposed design method is technology agnostic, thus enabling designing BC-based solutions and other technology-related solutions. Being a technology-agnostic design method allows the proposed architecture to be more robust, flexible, and agile, in the sense that it responds to changes in the business process, regulatory framework, and other provisions

that further impact the technological components, e.g., the BC platform. It considers different sources of knowledge, for example, regulatory framework, from that it enables performing of knowledge extraction and representation, developing standard common information model, which may be used for other purposes in TDG and related fields. It provides a mechanism for model transformation, which allows building new design method components, showing the targeted system component interactions, and to examine the targeted system's inner behavior.

Another considered scientific relevancy aspect of our design method, above all, our method is applicable on the *regulated* domains (or industries), where the processes are managed based on the regulatory framework (regulation policies and other contractual rules). The TDG and related industries are mainly regulated domains, and it is of paramount importance to consider the regulatory aspects when designing and systems or application means to supports their activities. Our approach intends to overcome these issues by considering the regulatory framework at the design level of any application, including a BC-based solution. Furthermore, we apply formal specification of the SC for the TDG, with the help of Linear Temporal Logic (LTL). The formal specification of SC allows the expression of business rules that determine the TDG process flow. The formal specification and further SC implementation based on this specification enable the management of business contracts in compliance with the regulatory framework.

The optimization of the TDG workflow process presents a significant contribution from this thesis. Referring to the actual way it is organized, mainly paper-based, to get authorization for TDG, there are required manual works and involvement of postal services to share the dossier among the involved stakeholder (as shown in the real case example in Section 5.3.6). Our approach provides a mechanism to improve the TDG process workflow by designing a BC-based solution. We deploy a unique solution that facilitates sharing and management of TDG related information. Instead of sharing a paper-based "dossier" for the TDG authorization purposes, the stakeholders fill in such information in a web form and share them directly with the competent authority. Furthermore, once the dossier is validated, it can be shared with other competent authorities, i.e., transit and destination. The information shared is based on BC security principles, thus avoiding information tampering, non-repudiation, and further increasing efficiency.

To highlight the dynamic aspects in the process of TDG, we presented time-related constraints (including strong and weak time constraints), geographic location constraints, the concept of the digital certificate, managing emergencies, the anomaly detection concept, shared responsibility, the multi-party SC, and SC maintainability approach.

Because of the sensitivity and risk exposure of the DG, geographic constraints are proposed as a preventive to avoid potentially disastrous situations in specific geographic areas. Its objective is to determine the specific set of geographic location constraints and to manage the process of TDG accordingly within the regulatory framework and specific conditions imposed by local (or international) authorities. Furthermore, to maintain the end-to-end lifecycle of the DG, we propose the digital certificate concept. It presents a digitally computed document for specific DG. It enables storing continuously immutable transaction-data, which

enables identification, traceability, and transparency over the DG lifecycle usability and treatment. Similarly, this concept can be used in other related domains where traceability and transparency of goods are required.

The management of emergency situations presents enormous challenges in the TDG. We propose a conceptual solution for managing emergencies based on the information received from IoT devices. This enables an accurate and adequate intervention process for the responsible stakeholders in the event of an accident with DG. To maintain and monitor the process of TDG under certain control, we propose the concept of anomaly detection. We specify SC for monitoring the state of DG and alerting when an "abnormal" situation is detected. For better management of the DG, we present two management-related concepts, i.e, "shared responsibility and "multi-party smart contract". Shared responsibility quantifies the stakeholders' responsibility involved in the process of TDG. The multi-party SC intends to improve the business ecosystem in the TDG and provide means to avoid risks in TDG, while for maintainability issues of SC, we proposed an approach based on the dynamic parameter (assets) change on the SC.

Beyond the MW use case, the proposed approach is also applicable in other use cases, for example, the transport and management of nuclear waste (or explosives). As already know, nuclear waste is a massive issue at European countries and beyond (NuclearWaste, 2019; RadioactiveWaste, 2021; Guardian, 2016; Science, 2018). Many concerns are raised about the actual way it is currently managed, especially on the transparency aspects. The responsible stakeholders provide an official statement, claiming that the nuclear waste is buried somewhere in the land, but, beyond that, many questions are raised about "Where these waste are buried and how this process is carried?"; "Which stakeholders carried this process?"; "Did authorities were able to monitor this process in an end-to-end way?"; "Does the nuclear disposal process has been subcontracted, and do these goods have been thrown illegally on open land or sea (Telegraph, 2016; Guardian, 2009)?"; " What is the impact for the future state of the area impacted and the earth itself"?

Today, the response to such a question is not well know nor supported by an available technological system. Furthermore, there is no formal proof behind the possible answer from the involved stakeholders, and the general public accessibility in such information is limited. In this thesis, we showed a scientific method that introduces a potential solution to these raised issues. Moreover, it provides a mechanism for sharing a certain level of information with the general public since nowadays, the general public is willing to be aware of the processes that might impact the earth and the entire life quality.

10.1 Perspectives: Deployment of TDG Blockchain Solution at International Level

The proposed research in this thesis, particularly the proposed design method, enables designing BC-based systems. We proposed a solution to deploying the TDG solution at local and international levels (cross-border context) using consortium BC and IoT. Such a method is also designated to be used for deploying BC-based solutions to an existing BC

network. In this perspective, we consider using the proposed design method to deploy a BC-based application of TDG at the European level BC network. Currently, there exists an initiative at the European level on developing "European Blockchain Services Infrastructure (EBSI)" (EBSI, 2021), which allows deploying a BC-based application in a certified BC network. This network is operated by official European country representatives (e.g., the Ministry of Digitalization at Luxembourg [1]) and currently, its objective is to cover public service use cases (EBSI, 2021). The EBSI networks correspond with the needs of BC-based application for TDG since competent authorities of each country govern it, and this might facilitate the exchange of information between competent authorities and stakeholders. Furthermore, another strong reason behind the intention to deploy TDG application at the EBSI network is to have more accessibility from any country and better control of DG at the International level. In such a way, it preserves to be part of an extensive network and to be used by any certified stakeholders.

10.2 Future Research Directions

This thesis is among the first studies that propose a design method for a BC-based system based on the standardized model-driven architecture (MDA) method. We consider this research a backbone for further direction on designing BC-based systems based on MDA and further advancing the engineering process of deploying BC-based systems. We consider the following future research directions.

a) Orchestration mechanism for several BC networks.

The deployment of the BC networks is currently occurring based on the domain (or business) application needs. At a certain level, these networks remain isolated in terms of further extension and, therefore, decrease the decentralization level. To this end, we consider that the current BC networks need a technological layer that enables orchestrations for several BC networks (i.e., platform) to improve the decentralization of the BC network. This intended research needs to be performed in two fields: at the infrastructure layer and at the application (consensus layer) to enable several BC networks to operate simultaneously. This layer will persist as a gateway between different BC networks and translate transaction structure (imposed by consensus mechanics) that will be stored in another BC network. This orchestration will also automatically adapt the BC node to a specific BC platform, e.g., Ethereum or Hyperledger, and allow the BC administrators some configuration level.

b) Anomaly detection in the end-to-end process workflow.

Another further research direction we consider is the anomaly detection in the end-to-end process workflow. We proposed a formal specification of the anomaly detection for the TDG. We consider, this topic opens different research perspectives by applying it in other use cases, such as in the financial domain (including finance SCM use cases). In such a scenario, a financial (or SCM stakeholder) institution might automatically detect the transaction's

[1] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/List+of+EBP+representatives

misbehavior, i.e., fraudulent or any other misbehavior. We consider an advancement in the field of anomaly detection by using BC and SC.

c) Multi-party smart contract to deal with conflict resolution among stakeholders.

The stakeholder collaboration in a dynamic environment engages conflicts in sharing or using specific resources. These are the cases where specific stakeholders' resources are significantly used compared to others in the same ecosystem (e.g., Roaming Services in 5G). In future works, we consider the SC to have a significant role to automatically detect, allocate and resolve stakeholder conflicts in a dynamic environment.

Appendix A

Appendix: Additional Chapter Components

This chapter comprises several sections whose purpose is to provide additional information for the concepts, examples, and general information to support thesis chapters.

A.1 Systematic research methods for the composition of the state-of-the-art

This section presents the systematic research method used for composing the state-of-the-art Chapter (3 and 4). In the context of our study, we systematically consider the most relevant research articles that are related to our use case. We remove articles that treat a similar topic several times, e.g., traceability in SpC, and consider only the most relevant articles. The relevance is quantified based on the research topic, article citations, publication media, and article clearance.

For each section in Chapter (3 and 4), we considered certain number N of articles, where $N \in \{5 - 25\}$ dedicated to each section. Therefore we defined four different sets of articles. *Set 1: Blockchain Technology and Smart Contracts* includes any article related to the definition, characteristics, and main functionality of blockchain technology, and it is mainly used in Chapter 3. *Set 2: Blockchain in SCM (trust, traceability, transparency, information sharing, business process management)* is used to compose several sections related to blockchain and Logistics, shown in Chapter 4. *Set 3: Integration of Blockchain and IoT*, includes articles related to blockchain and IoT. *Set 4: Formal Specification and Verification of Smart Contract* gathers articles that present formal methods for specification and verification of smart contracts.

The research libraries used are *Web of Science*, *IEEE Xplore digital library*, *ACM*, *Scopus*, *Google Scholar*, *Web of Science (CORE)*, *DBLP*, *White papers*, relevant websites and many other.

We formalized a set of queries used on main research libraries to compose the research article sets. The research method is presented below, and it is used in a different context based on the required set of articles. Following, we present the research method steps by targeting the set of articles that research on *formal specification and verification of the smart contract*:

S1: Query definition by using keywords: Smart contract and (or) formal modeling, model checking, security, verification, contract validate tools, compliant smart contract, consistency, correctness.

S2: Search (using S1) for research articles in main research libraries;

S3: Formalizing the set of articles;

S4: Analysis of the abstract and details of articles (selected on S3), and classifying the articles based on the research topic;

S5: Eliminating the non-relevant articles and reformulation queries (S1), if necessary after discovering other possible research articles (challenges);

S6: Repeating steps S2 - S5, until the same results are shown again;

A.2 Difference between CIM-MDA and CIM-DMTF

The CIM-MDA mainly reflects the domain issues by modeling the problem (for example, UML Class Diagram (or Use Case diagram or Activity Diagram, or textually expression), without referring to a particular system implementation or technology. CIM-MDA remains independent of the system implementation, whether the system is implemented with the help of technology or entirely mechanically. CIM-MDA does not show any information about how the system will be developed and is considered a primary source of information shared between domain experts and software engineers. CIM-MDA is mainly concentrated on one specific use case.

Contrary to CIM-MDA, the CIM-DMTF is broader and intends to define standardized system components. It presents a common definition of management information for application services and networks. CIM-DMTF is designed to serve as a referential data model and is shared among different applications. CIM-DMTF includes the necessary formal expression that serves as a source of knowledge for its intended systems, services, and network components. CIM-DMTF is a general model that allows different parties to share the same information based on the CIM-DMTF model. CIM-DMTF can be reused by different stakeholders in different contexts for developing future systems

A.3 Algorithm: Legal Text to BPMN

Algorithm 8 shows steps for translation of the regulatory framework into the BPMN. This expression helps to understand the process of extraction of concepts and mapping them into BPMN components. We highlight the fact that this process is composed manually by a human expert.

Algorithm 8: The algorithm for mapping BPMN model from legal documents (text).

```
1  Input: Legal Documents;
2  Output: List of Concepts; Relationships;
3  while Exists legal text for analysis do
4      S1: Concept identification: Stakeholder; Related Legal
        Documents; Administrative Procedures;
5      S2: Identify interaction between stakeholders;
6      tasks (activities), specific condition, dependencies, time
        constraint;
7      S3: Compose logical flow (Article X follows Article Y);
8      if Article X == condition_fulfilled then
9          Go To Article Y
10     else
11         End of Process
12     end
13 end
14 if List of Concepts is completed then
15     Perform mapping of concepts into BPMN;
16     task ← activity (task);
17     stakeholder ← lane;
18     condition ← gateway ;
19     information_exchange ← message_event;
20     send/receive_documents ← message_event;
21     automatic_responses ← time_based_event;
22     documents ← data_object;
23     data ← data_store;
24 else
25     End of Process
26 end
```

A.4 Presentation of DG Official Signs Based on ADR

Different singes indicate DG. These singes are defined by the official organization UNECE UNECE, 2017, which operates under United Nation's authority. The visual singes (labels) of the DG are presented as in Figure A.1.

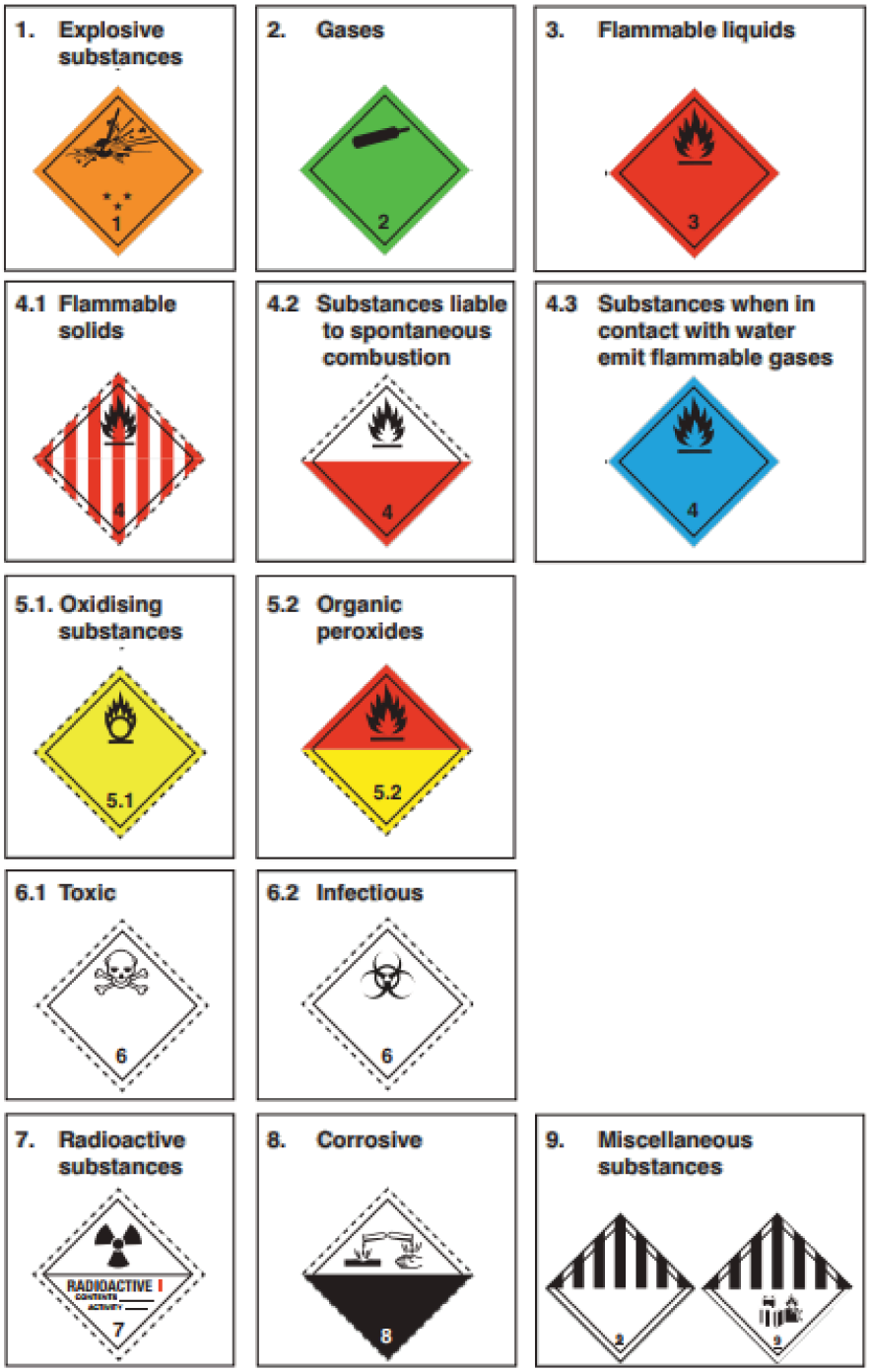

FIGURE A.1: The DG official labels according to DG classes.

A.5 Modeling Aspects: Defining the Basic Terminology

The presented models and related works are formulated based on the modeling standards provided by Object Management Group (OMG) (OMGGroup, 2020).

OMG is a consortium of international organizations that provides modeling standards. OMG is a non-profit organization that aims to develop technology standards commonly accepted by businesses and vendors, end-users, government agencies, and academic institutions (OMGGroup, 2020). A long range of modeling standards and architectural approaches are listed by OMG, including Unified Modeling Language (UML), Business Process Management Notion (BPMN), Model-Driven Architectures (MDA), and many others.

Concept present an abstract representation of the "thoughts", "object", "things" etc.

System presents a collection of parts and relationships between any organization (or elements) that are intending to realize something or fulfilling specific functions by working a whole (OMG-MDA, 2014). A system might be composed of several sub-systems that formulate the entire system, or a system might have relations (connections) to another system (Silva, 2015; Seidewitz, 2003; Kühne, 2006).

Meta-Model. In the modeling approach, the meta-model presents one of the highs abstract levels (Tjoa et al., 2006; Atkinson and Kuhne, 2003). There are several definitions regarding meta-model, and in general, there is not a common agreement over the definition of the meta-model. In (Kleppe et al., 2003) "meta-model is defined as a precise definition of components and rules that are used for defining a model". Meta-Object-Facility (MOF) (OMG, 2017) defines meta-model as language for expressing a model. The research from (Tjoa et al., 2006; Atkinson and Kuhne, 2003) defines "Metamodeling, or meta-modeling, is the analysis, construction, and development of the frames, rules, constraints, models, and theories applicable and useful for modeling a predefined class of problems."

Our *proposal for defining the meta-model* is as follows: "Meta-Model: is a referential model that provides a formal definition of the components, properties, and the relationships between these components".

Model is an abstraction and simplification of reality, resulting from a process of abstraction. A model intends to present formal specifications that highlight the functionality, structure, and behavior of the system (Tjoa et al., 2006; Selic, 2003; Seidewitz, 2003; Kühne, 2006). Generally, a model represents a simplified view of the studied system, and usually, there are many models developed for better representation of the in a more understandable way the addressed system (Silva, 2015; Seidewitz, 2003). A model can present a domain, software, business, environment, hardware, and other domain-related aspects (OMG-MDA, 2014). The modeling language and processes defined by the meta-model should be considered for the development of a model.

Modeling language. For models to be understandable by the involved stakeholders, a modeling language plays a significant role. A modeling language is any structure, terms, notations, syntax, semantics, and integrity rules that are used to express a model". For defining a modeling language, initially we should compose the *abstract syntax* and *concrete syntax*.

An *abstract syntax* presents domain analysis phase which allows identification of concepts, abstraction and relationship of domain components (OMG-MOF, 2019; Silva, 2015). A *concrete syntax* presents the notion of the modeling language. The notion of modeling language might be graphical visual textual, tabular, form based (Silva, 2015). Among the well known modeling language are: *"UML"* (A.5.1), *"BPMN"* (7.3), SQL Schema, OWL, and XML Schema (OMG-MDA, 2014; Truyer, 2006; Kühne, 2006).

Architecture or system architecture allows understanding the scope of interest by defining systems or improving the currently existing systems (OMG-MDA, 2014).

Viewpoint specifies an abstract technique that allows focusing on particular concerns within a system. It presents criteria for selecting and presenting part of the information (or model) in a present system (Truyer, 2006; OMG-MDA, 2014).

Platform is a set of resources or several subsystems and technologies that implement a specific set of functionalities to realize a system (Truyer, 2006; OMG-MDA, 2014). Examples of platforms are programming languages, databases, operating systems, etc.

A.5.1 Unified Modeling Language (UML)

UML is a standard modeling language (A.5) composed of different diagrams (models) used to visualize, specify, and document artifacts of the system under development. It covers an extensive range of modeling aspects that could be used in different domains, e.g., business, finance. The UML was adopted to them OMG standards, and it is listed among the main modeling standards. The range of UML modeling capabilities is extensive, and it might not be necessary for all domains. It covers different diagrams, e.g., "Class Diagram", "Activity Diagram", "Sequential Diagram", "State Machine Diagram", and many others. UML models present a different phase of the system under development. The UML model are classified as "structural diagram", "behavior diagram" or "interaction diagram". For each specific domain, only a part of the modeling components might be useful (OMG-UML, 2010) (Fowler, 2004).

A.5.1.0.1 UML Class Diagram and Relationships. UML class diagram presents the static structure of the system under development. It describes the structure system presented by classes, class attributes, operation, and the relationship. In UML relation presents the logical connection between two UML components. Figure A.2 shows an example of UML class and the relationship.

A.5.1.0.2 UML Sequence Diagram. The UML Sequential diagram presents a modeling language that shows the interaction of the component of the system (OMG-Superstructure, 2014). It is mainly used to illustrate the interaction (information exchange) between system components. The main components of the UML Sequential diagram are:

- *Actor* represents an external entity that interacts with the system,
- *Lifeline* presents an internal instance to the system,
- *Messaging* presents a particular communication between system components (lifeline).

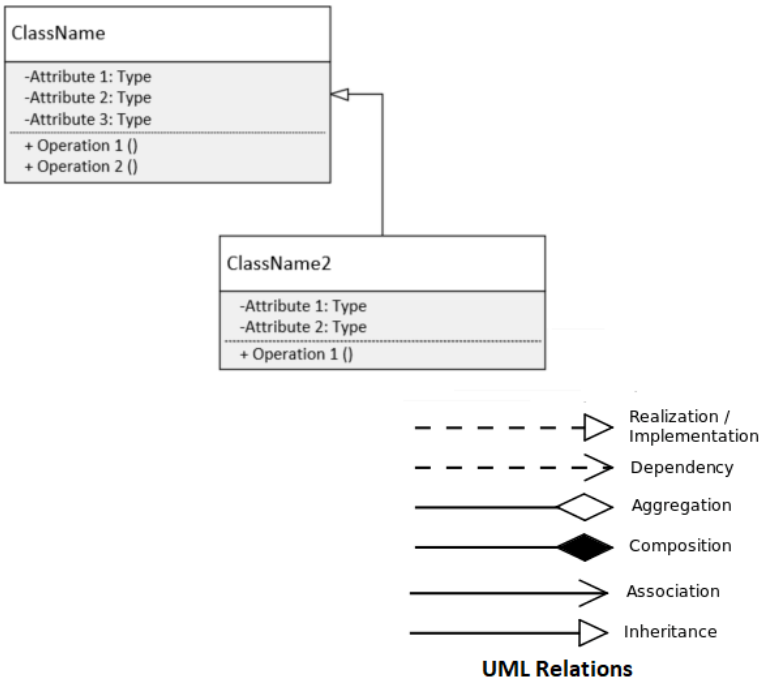

FIGURE A.2: The example of UML class and relationships.

A.5.2 Business Process Model and Notation (BPMN)

The modeling aspect presents one of the principal components of our approach. It allows us to formally express the main components necessary to consider in the process of TDG. This section presents the process model, its main components, relationships, and the interaction required for the TDG, expressed in BPM (A.5.2.1). The BPM follows a process approach, and this is suitable for the TDG process since we intend to cover an end-to-end process.

A.5.2.1 Business Process Management (BPM)

Business Process Management (BPM) presents a discipline dedicated to providing an overview of BP by analyzing, designing, implementing, and continuously improving the BP. BPM presents a life cycle that enables managing and improving BP. It is composed of six phases cycled together, that are presented as follows (Dumas et al., 2013; Brocke and Rosemann, 2015):

- **Process Identification**, which allows identification of the process;
- **Process Discovery**, this phase allows to discover the process *"as is"*;
- **Process Analysis**, allows us to analyze the "discovered" (as is) process and to find ongoing issues by performing qualitative and quantitative analysis;
- **Process Redesign**, this phase enables process redesigning and to improve it at the level of "to be" as a process;
- **Process Implementation** intends to push the process into execution, including some automation aspects of the process;

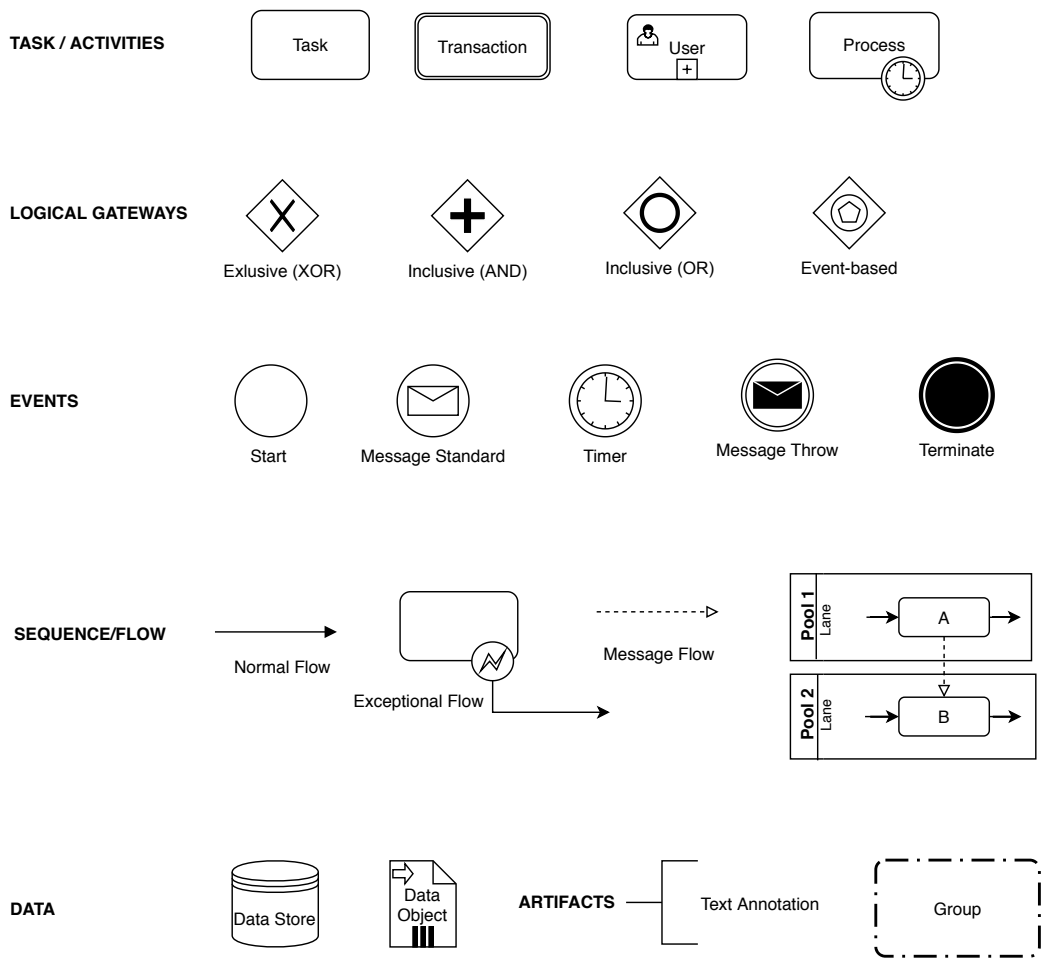

FIGURE A.3: The basic BPMN Symbols.

- **Process Monitoring** consists of monitoring the process (logging), analyzing the current state, and showing the performance of the process.

The process analysis and development provide in-depth knowledge regarding a process. We consider this step crucial for further developing our design method since it will allow us to discover requirements and perform analysis over the process of TDG. We follow the BPM approach's main steps to analyze and explore the organization of the process of TDG. Further, we evaluate the current performance in terms of efficiency (time required for completing a lifecycle in the process) and finally to find ways to optimize and improve the process.

A.5.2.2 Modeling Language: Business Process Model and Notation (BPMN)

Business Process Model Notation (BPMN) is a standard modeling language that is provided and maintained by OMG (OMG-BPMN, 2013). BPMN uses a graphical notation to describe the business process as a flow chart. As a standard modeling language, BPMN is understandable from the business community and from the technological users aiming to implement

it, for example, at the information technology level. Figure A.3[1], shows the main notation of BPMN. The first (highest) level from Figure A.3 shows "Activity" or "Task". Any work performed within the process is called activity (task). Secondly, during the process, there are divergence and convergence controlled by the "Gateways". Gateway presents a logical decision, and they control the flow of the process. Thirdly, anything that "happens" during the process is represented by "Events". The fourth levels show the "Sequence flow", which indicates the order that an "Activity" will be executed in the process. The "Message Flow" is used to perform the interaction between the process participants in an orchestrated manner. The fifth level shows other notation objects such as "Data Object", which shows any demanding data on the process. The "Data Store" is used as a source of data in a particular process.

A.6 Ontology Web Language (OWL)

In computer science, ontology [2] presents knowledge representation science to capture naming and representing concepts, relationships, and properties. In general, ontology is used to capture knowledge in a specific domain known as the universe of discourse (Horridge, 2009). Different ontologies allow knowledge representation. The Ontology Web Language (OWL) proposed and maintained by the W3C[3] community presents a standard ontology language[4]. The OWL[5] is considered a language that describes classes, attributes, and object relationships. This language is understandable from computers (machines) (Vo and Hoang, 2020). OWL uses the strength of description logic (DL) to manage the meaning of semantic annotation in order for them to be understandable and accessible from computer systems (and humans) (Horrocks et al., 2007; Hofweber, 2020).

The main OWL components are individuals, properties, and classes (Horridge, 2009). Individuals (also known as instances) present the object from the domain in which we are interested (the domain of discourse)—for example, Luxembourg, Good, John, all present individuals from different discourse domains. The properties present binary relations on individuals. For example, the *isTransporting*, links two individual, i.e., John *isTransporting* Goods. Classes are presented as a set that contains individuals. For example, DangerousGood contains all individuals of dangerous goods(Horridge, 2009). In general, classes present concepts of the domain of discourse. Axioms are used for determining and clarifying what is true in the domain. OWL offers a different range of axioms such as class axioms, object properties axioms, and many more (W3C, 2020).

[1]The BPMN elements showed in this image are the basics ones since there is an extensive set of this notation presented in (OMG-BPMN, 2013) and in `https://camunda.com/bpmn/reference/`

[2]Ontology is a philosophic branch know as the science that studies concepts of "existence, being, becoming and reality" `https://en.wikipedia.org/wiki/Ontology`.

[3]`https://www.w3.org/2007/OWL/wiki/OWL_Working_Group`

[4]OWL Syntax: `https://www.w3.org/2007/OWL/wiki/Syntax`

[5]There exists other languages for defining an ontology, such as RDF and RDFS, and many others

A.6.1 The OWL classes of CIM Profile for CIM-TDG Schema

Figure A.4 shows an extended version of the OWL classes for CIM-TDG Profile. At the time of writing, the current ontology matrix is composed of 163 Classes, a total of 770 axioms, 403 logical axioms, 346 declared axioms, and 129 individuals [6].

[6]The complete ontology documentation is presented in the following link: https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/CIM-TDG_Ontology_Documentation.zip

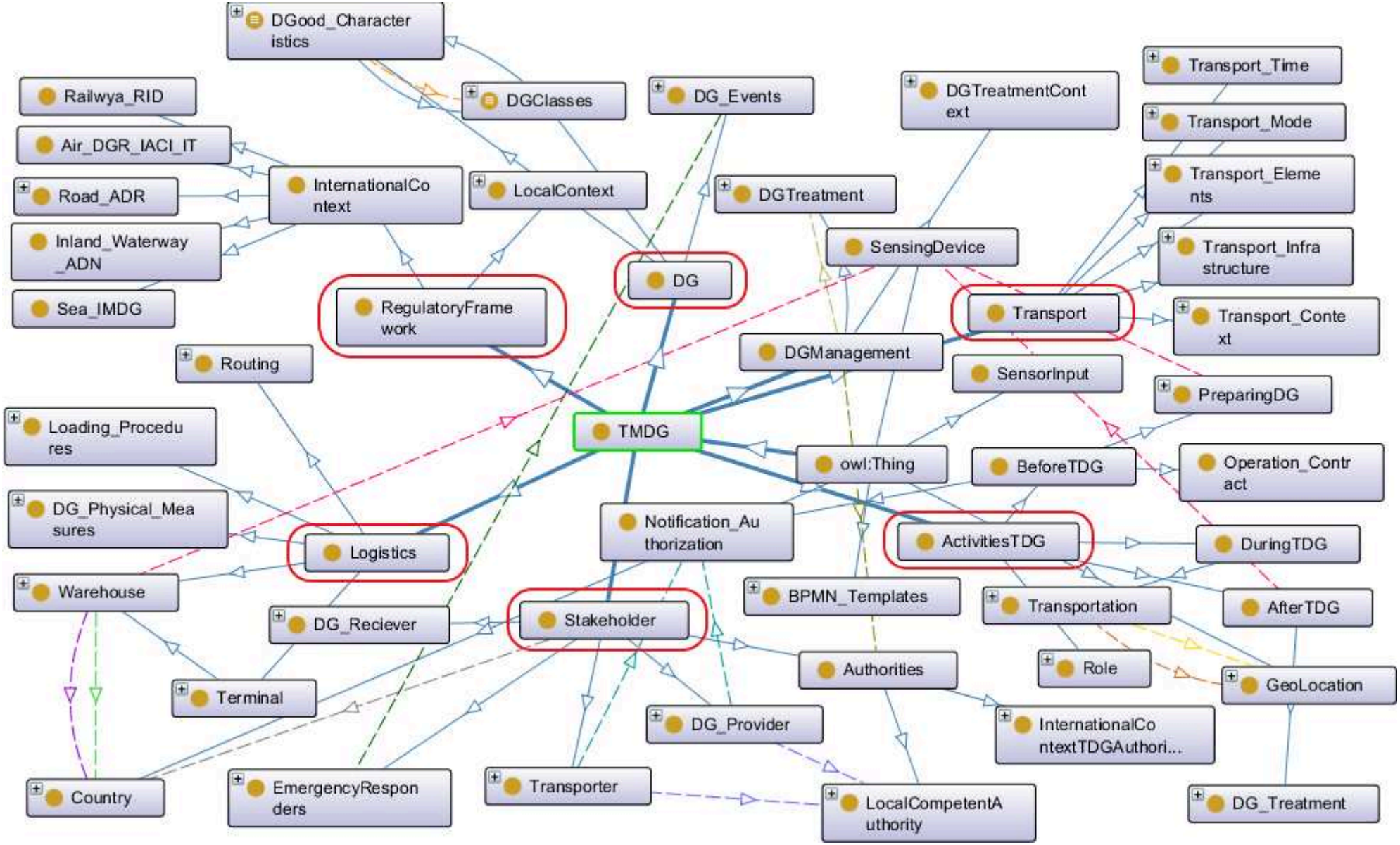

FIGURE A.4: The main classes in the OWL model representing the ontology of TDG.

A.7 Technical Components of Blockchain

This section shows some of the main blockchain technology components.

A.7.1 Hashes

A hash function is a mathematical method (one way function) that allows producing a fixed-size data as output (called usually digest, hashes or hash code) from any length input data (e.g., book or even a single character "c") (Dang, 2015). The hash gives a unique digest for any value give, and for any change, on the source (even a single character or space) a new hash will generate. The example[7] we provide below we have the hash for "0" which is*5feceb66ffc86f38d9527 86c6d696c79c2dbc239 dd4e91b46729d73a27fb57e9*, and we add new each time and element *i* (by increasing *i + 1*) and we receive each time different hashes (digest) (Dang, 2015; Antonopoulos, 2017; Yaga et al., 2018; Lewis, 2016).

a) The properties of the hash function indicate that it is hard (almost impossible) to calculate the original data from the hash, and the slight changes on the data (input) impose unpredictable changes on the output hash (digest) (Antonopoulos, 2017). b) for the given hash value input, it is almost impossible to find the second input value which produces the same hash value (thus given x, find y such that hash (x) = hash (y)). c) Hash is considered collision-resistant[8], meaning that cannot find two inputs different inputs that generate (hashes) the same output (Yaga et al., 2018; Antonopoulos, 2017).

Blockchain strongly utilizes the hash function[9] for address derivation, unique identifiers, digesting transaction data, and securing block header (Yaga et al., 2018). Anyone can prove that if the data has been changed or not. That is possible by comparing the hash of the data, for example, word "source_data"; hence hash (source_data) = 81B19DD8E3D8CA75B0 ECC9EBB5F071D251F4AFF504C2C1DEE0461AC0B77CEC34, then to prove if any change is performed in the "source_data", we perform hashing on the "source_data" again and if the hash match (same hash) then the data is not changed. Otherwise, if the hash is not matching, the "source_data" is changed.

```
import hashlib
class HashGeneration:
N = int(input("give the number N "));
for i in range(0, N):
genHeshFor = 0 + i
hash = hashlib.sha256(str(genHeshFor)).hexdigest()
i = i + 1
print hash
```

LISTING A.1: Example of hash digest generation for different inputs.

[7]This example is inspired from research in (Antonopoulos, 2017)

[8]It is almost impossible to have a collision where two different input would generate the same output (hash), hash (x) = hash (y).

[9]Secure Hash Algorithm (SHA-256) with an output size of 256 bits (or 32 bytes, (32 x 8 = 256 bits), displaying a 64-character hexadecimal string). There are 2^{265} possible digest values (Dang, 2015). National Institute of Standards and Technology (NIST) provides an official standard on the SHA algorithms, available on https://csrc.nist.gov/projects/hash-functions. An available link to try and generate the hash, in different output sizes, is provided as follows: https://passwordsgenerator.net/sha256-hash-generator/

For blockchain technology, the most used hashing algorithm is the Secured Hash Algorithm (SHA), with an output of 256-bit. There exist other hashing algorithms such as SHA-512, Keccak-p[10] (Dworkin, 2015; Yaga et al., 2018; Antonopoulos, 2017).

A.7.2 Public Key Infrastructure (PKI)

The public key cryptography or asymmetric-key cryptography is an encryption schema that provides two mathematically related keys but is not identical. The public and private keys are combined for the encryption and decryption process, respectively. The public key is used to encrypt the "message", while the private key is used to decrypt the "message" (Anshel et al., 1999; Al-Riyami and Paterson, 2003). The public key is publicly shared (they are long and hardly possible to remember them, mainly they are distributed on "certificates") since it is computationally infeasible to generate the private key by knowing the public key. The private key is kept secret by their owners (they are stored on specific software, or out of computers, or in hardware equipment, e.g., USB). The "messages" sent, for example, by Alice, that is encrypted by using the public key of, e.g., Bob, and it is possible to be decrypted only by the private key of "Bob" (Al-Riyami and Paterson, 2003).

In the blockchain, e.g., Bitcoin, Ethereum, and other blockchain frameworks, users use public and private keys to sign a transaction on blockchain-based systems digitally. The public key is used to receive funds (assets), and the private key is used to sign a transfer of funds (assets).

The most common PKI used from the blockchain technologies is the Elliptic Curve Digital Signature Algorithm (ECDSA) (Johnson et al., 2001). The private key (pk) is a number that is selected randomly, and there is used ECDSA (one-way function) for the generation of the public key (kP) (Antonopoulos, 2017). The private keys are used to sign the transaction digitally. Public keys are used to generate user addresses (one or several) and verify the signature generated with the private keys (Yaga et al., 2018). Private keys must be kept secure and safe. For managing public and private keys, many software solutions, i.e., wallets (A.7.5), are proposed. Losing a private key signifies losing all assets linked to that private key (Antonopoulos, 2017; Yaga et al., 2018).

A.7.3 Addresses

The user addresses (e.g., Bitcoin address: *18jJh1kSPJqbXtMB51SyczgcHL1drk DgxV*) are derived from their public key (and some additional information) by using the hash function (one way function) over it hash (kP). This makes the addresses shorter than the public key and more manageable for the users. This address is further distributed publically, and it is used to transfer or received digital assets. Beyond representing the public key, the address on the blockchain might represent other objects, e.g., scripts or smart contracts (Antonopoulos, 2017; Yaga et al., 2018).

[10]Ethereum uses Keccak-256 as a hashing algorithm.

A.7.4 Transaction

The transaction performs any interaction on the blockchain. A transaction presents financial interaction, i.e., transferring "cryptocurrency" from one blockchain address to another, recording the business organization's activities, performing activities on the blockchain, e.g., executing the smart contract, and many other activities based on the purpose of blockchain usability (Xu et al., 2017; Nakamoto, 2009).

A.7.5 Wallets

Since the public and private keys are long and hard to remember, many blockchain systems propose using software packages that securely store private, public, and other related addresses. This software is called wallets, e.g., Bitcoin Wallet, and is used to store and manage public and private keys as well digital assets associated with these keys (Yaga et al., 2018; Antonopoulos, 2017).

A.7.6 Mining Process Based on *Proof of Work* Consensus Algorithm

Practically, mining the blockchain block simply means hashing the block header, changing specific parameter (nonce) in the block header continuously until the generated hash matches the specific target (Antonopoulos, 2017; Wang et al., 2018a). The hash (SHA256) function features deny any possibility, create a pattern that generates a specific hash digest, or determine in advance a specific hash digest. Thus, the only way of finding a specific hash digest us by trying again and again until they find that targeted hash digest (Antonopoulos, 2017; Dang, 2015).

Following we present a python code[11] that generates hash digest values manipulated by "nonce":

```
1
2   import hashlib
3   import time
4   class HashNonceGeneration:
5   N = int(input("give the number N "));
6   firts_zeros = int(input("define leading number of zeros for hash digest ")); #
        for example 2
7   some_Text = "some text"
8   print "there will be " + str(N) + " iterations: "
9   start_time = time.time();
10  for nonce in range(0, N):
11  genHeshFor = some_Text + str(nonce)
12  hash = hashlib.sha256(str(genHeshFor)).hexdigest()
13  print "For the nonce: " + str(nonce) + ";" "the complete phrase
14  becomes " + str(genHeshFor) + "  and this is the hash digest: "
15  + str(hash)
16  nonce = nonce + 1
17  firstChars = hash[0:zeros]
```

[11] The full code is available on the following link: https://git.list.lu/adnan_imeri/thesisaimeri/-/blob/master/HashNonceGeneration.py

```
if (firstChars == '00'):
print "hash found " + str(hash)
end_time = time.time()
total_time = end_time - start_time
print "The time needed to generate hash digest: "
+ str(total_time)
sys.exit(-1)
```

LISTING A.2: Example of hash digest generation by manipulating the "nonce".

The code above generates a different hash digest based on the nonce. We challenge the above code by adding a *target*: "find a hash digest that starts with 0". The goal is to find a hash value that is less than the *target*. For example, from the execution of the code above, with N = 100, we found a hash digest that starts with 0 at the nonce = 40. So, in the 40th attempt we have hash digest "081f22220e42db569a44c59dc01e91d295ded2dec74791e5645647f2968469cc". Further, if we adjust the targe by adding the second leading zero (0), then the algorithm will need more attempts to find the target. So, we have, for the phrase "some_text" + nonce (217) = "006ce05c39cfd9c15e39fd8e4f91695d0b3aeea8790a160028b3e29802aacad4", and it took 217 attempt to find the digest that starts with "00". In this way, since the SHA-256 is deterministic on generating digests, the miners attempt to find the target by taking some input (proof) and essentially generate the targeted digest (work), thus forming Proof-of-Work (PoW) (Antonopoulos, 2017).

A.8 First-Order and Temporal Logic

A.8.1 First-Order Logic (FOL)

In mathematical logic, the first-order logic (or predicate logic) presents the collection of mathematical expressions that allow us to build formal semantic. It presents a necessary and sufficient condition in a way that the first-order sentence to be true (Chiswell and Hodges, 2007). Compare to propositional logic it use quantifier, such as "exists" $\exists$ e.g., $\exists\, x \wedge x > 10$ and "for all" $\forall$, e.g., $\forall\, x$ *is prime* (Chiswell and Hodges, 2007). FOL offers logical connectors such as conjunction ($\wedge$), distinction ($\vee$), negation ($\neg$), and implication ($\rightarrow$). FOL allows to express knowledge representation of systems, and it considers ideal for the axiomatization of ordinary mathematics. FOL allows us to express the statement and to combine them with operator logic, thus offering reasoning over the mathematical statements as representative of the knowledge system.

A.8.2 High Level Logic: Temporal Logic

Temporal logic enables the specification of system properties based on combination proposal logic operators and reasoning over the atomic properties (Rozier, 2011). The temporal aspects refer to the time logic, meaning that propositions are qualified in terms of time, so only the temporal order of events matters. The ordering of the events is performed on logic time (without introducing the time explicitly), and it considered two different time concepts:"Linear Temporal Logic" and "Branching Time Logic". We presents some general

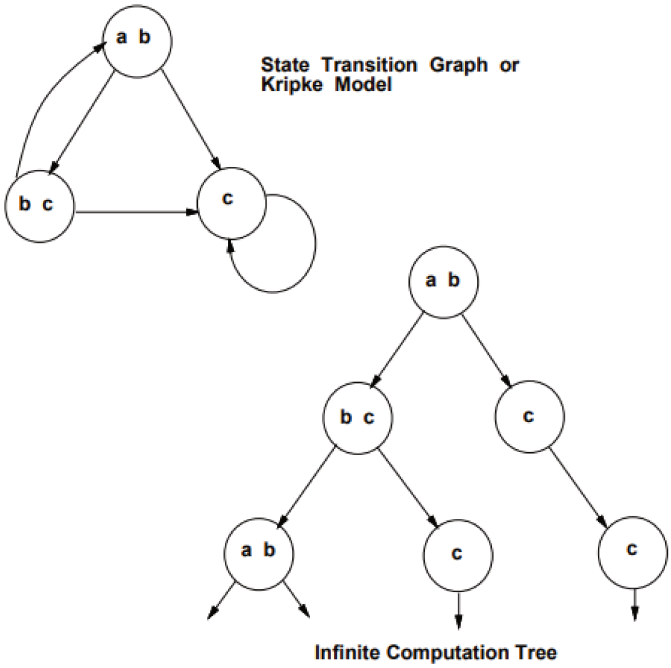

FIGURE A.5: Unwind State Graph to obtain Infinite Tree (Clarke, 1999).

notion regarding Temporal logic and its branching based in the following sources (Baier and Katoen, 2008; Rozier, 2011; Clarke, 1999; Rabbi, 2015; Gueffaz et al., 2012; Chiswell and Hodges, 2007).

A.8.2.1 Computational Temporal logic

In a state system, if some state is designated as the *initial state*, the structure can be unwound into an infinite tree with that state as the root. We will refer to this infinite tree as the *computation tree* of the system. Paths in the tree represent possible computations or behaviors of the program.

Formally, a *Kripke structure* is a triple $M = [S, R, L]$, where

- S is the set of states,
- $R \subseteq S \times S$ is the transition relation, and
- $L : S \to \rho(AP)$ gives the set of atomic propositions true in each state.

We assume that every state has at least one possible successor r (i.e., for all states $s \in S$ there exists a state $s' \in S$ such that $(s, s') \in S$).

A path in M is an infinite sequence of states, $\pi = s_0, s_1, ...$ such that for $i \geq 0, (s_i, s_i + 1) \in R$.

For a path $\pi = (s_0, s_1, ...)$, state s_0 is considered to be at the present time.

We write π^i to denote the *suffix* of π starting at s_i.

Unless otherwise stated, we assume *finite* Kripke structures.

Temporal logic may differ according to how they handle branching in the underlying computation tree. In linear temporal logic, operators are provided for describing events along a single computation path. In a branching-time logic, the temporal operators quantify over the paths that are possible from a given state.

Linear Temporal Logic (LTL). Formulas in LTL are applied over the infinite path in the Kripke structure. For defining such formulas, a combination of temporal operators, atomic propositions, and Boolean connectives from FOL-logic are used.

- $\mathbf{X}p$ — p holds true *next* time.
- $\mathbf{F}p$ — p holds true sometime in the *future*
- $\mathbf{G}p$ — p holds true *globally* in the future
- $p\mathbf{U}q$ — q holds true *until* p first becomes true
- $p\mathbf{R}q$ — is dual (p release q), and it steads that q must be true now and remains true until the time when p becomes true, thus releasing

For a path $\pi = (s_0, s_1, ...)$, state s_0 is considered to be at the present time.

- If p is an atomic proposition, then p is a state formula.
- If f and g are state formulas, then $\neg f$ and $f \vee g$ are state formulas.

Two additional rules are needed to specify the syntax of path formulas:

- If f is a state formula, then f is also a path formula. (A state formula f is true for a path π if the f is true in the initial state of the path π.)
- If f and g are path formulas, then $\neg f$, $f \vee g$, $\mathbf{X}\ f$, $\mathbf{F}\ f$, $\mathbf{G}\ f$, and $f\ \mathbf{U}\ g$ are path formulas.

Computational Temporal Logic (CTL) In this logic a *path quantifier* can prefix an assertion composed of arbitrary combinations of the usual *linear-time* operators. In temporal operators in CTL are combined with path quantifiers:

1. Path quantifiers:
   - **A** — "for every path"
   - **E** — "there exists a path"

There are eight basic CTL operators:

- **AX** and **EX**,
- **AG** and **EG**,
- **AF** and **EF**,
- **AU** and **EU**

Each of these can be expressed in terms of **EX**, **EG**, and **EU**:

- $\mathbf{AX}\, f = \neg\, \mathbf{EX}(\neg f)$
- $\mathbf{AG}\, f = \neg\, \mathbf{EF}(\neg f)$
- $\mathbf{AF}\, f = \neg\, \mathbf{EG}(\neg f)$
- $\mathbf{EX}\, f = \mathbf{E}[\mathit{true}\ \mathbf{U}\ f]$
- $\mathbf{A}[f\ \mathbf{U}\ g] \equiv \neg\, \mathbf{E}[\neg g\ \mathbf{U}\ \neg f \wedge \neg g] \wedge \neg\, \mathbf{EG}\ \neg g$

For a path $\pi = (s_0, s_1, ...)$, state s_0 is considered to be at the present time.

The syntax of state formulas is given by the following rules:

- If p is an atomic proposition, then p is a state formula.
- If f and g are state formulas, then $\neg f$ and $f \vee g$ are state formulas.
- If f is a path formula, then $\mathbf{E}(f)$ and $\mathbf{A}(f)$ are state formulas.

Two additional rules are needed to specify the syntax of path formulas:

- If f is a state formula, then f is also a path formula. (A state formula f is true for a path π if the f is true in the initial state of the path π.)
- If f and g are path formulas, then $\neg f$, $f \vee g$, $\mathbf{X}\ f$, $\mathbf{F}\ f$, $\mathbf{G}\ f$, and $f\ \mathbf{U}\ g$ are path formulas.

If f is a state formula, the notation $\boxed{M, s \models f}$ means that f holds at state s in the Kripke structure M.

Assume f_1 and f_2 are state formulas and g is a path formula. The relation $M, s \models f$ is defined inductively as follows:

1. $s \models p \Leftrightarrow$ atomic proposition p is true in s .
2. $s \models \neg f_1 \Leftrightarrow s \not\models f_1$.
3. $s \models f_1 \vee f_2 \Leftrightarrow s \models f_1$ or $s \models f_2$.
4. $s \models \mathbf{E}(g) \Leftrightarrow g$ holds true for some path π starting with s
5. $s \models \mathbf{A}(g) \Leftrightarrow g$ holds true for some path π starting with s

If f is a state formula, the notation $\boxed{M, \pi \models f}$ means that f holds at state s in the Kripke structure M.

Assume g_1 and g_2 are state formulas and f is a path formula. The relation $M, \pi \models f$ is defined inductively as follows:

1. $\pi \models f \Leftrightarrow$ s is the first state of π and $s \models f$.
2. $\pi \models \neg f_1 \Leftrightarrow s \not\models g_1$.

3. $\pi \models g_1 \vee g_2 \Leftrightarrow \pi \models g_1$ or $\pi \models g_2$.
4. $\pi \models \mathbf{X} g_1 \Leftrightarrow \pi^1 \models g_1$
5. $\pi \models \mathbf{F}\, g_1 \Leftrightarrow \pi^k \models g_1$ for some $k \geq 0$
6. $\pi \models \mathbf{G}\, g_1 \Leftrightarrow \pi^k \models g_1$ for some $k \geq 0$
7. $\pi \models g_1\, \mathbf{U}\, g_2 \Leftrightarrow$ there exists a $k \geq 0$ such that $\pi^k \models g_2$ and $\pi^j \models g_1$ for $0 \leq j < k$.

Recall: For $\pi = (s_0, s_1, ...)$, we write π^i to denote the suffix starting with s_i.

Notice that $\mathbf{F}p$, $\mathbf{FF}p$, $\mathbf{FFF}p$ etc., hold true for a path π even if p holds true at only the initial state in the path π.

A.9 The DT in TDG

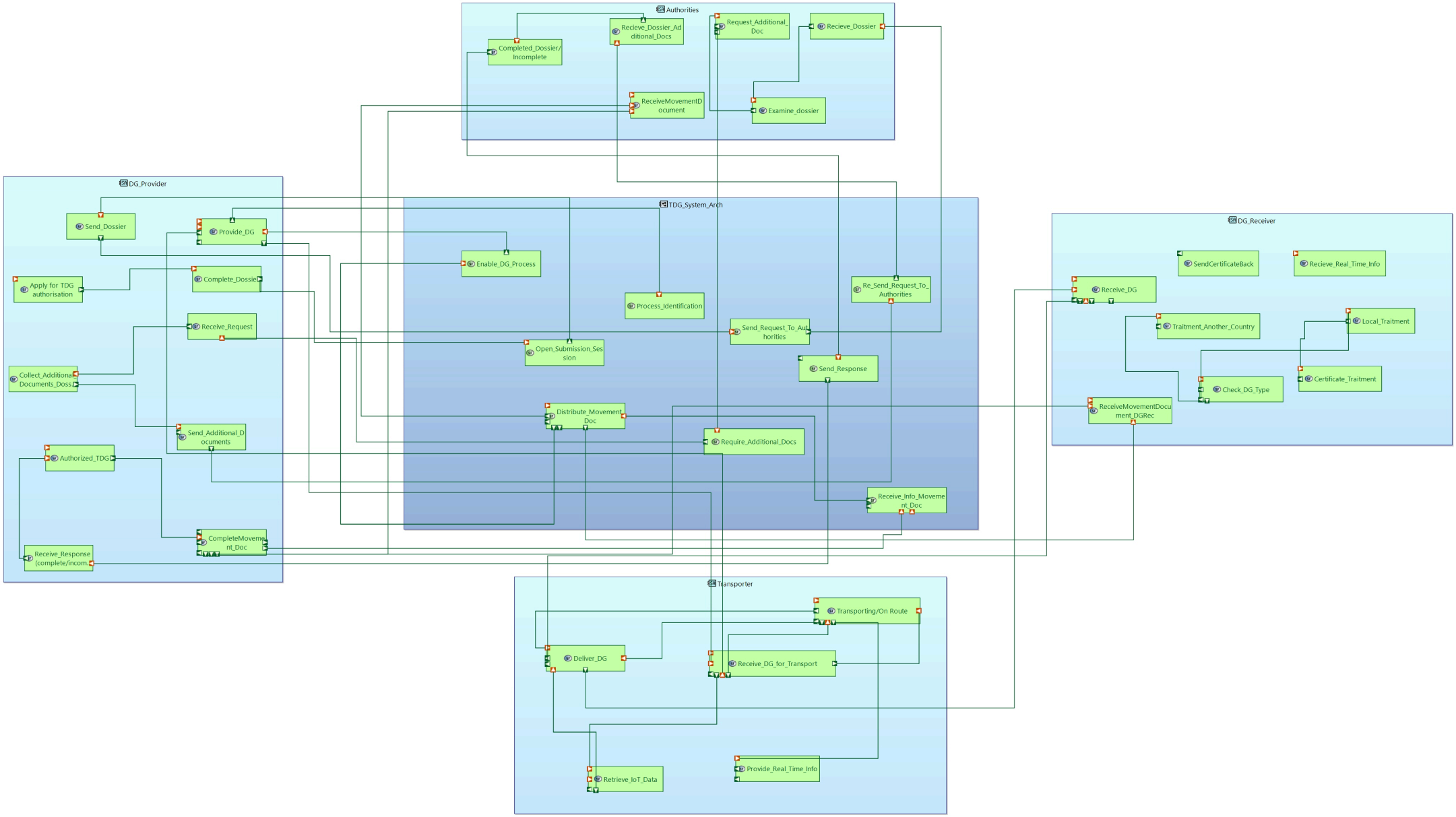

FIGURE A.6: The example of DT in TDG.

A.10 Smart Contract Code Examples

For example, we define SC for reading (querying) and write transactions to support the business logic. Once we develop the SC, we install (HF code for SC install: `./network.sh deployCC -ccn registedDG -ccp ../registerDG/chaincode-go -ccl go`) it on the channel e.g., "Global Channel". HF offers different SDK for developing SC (or chaincode), such as Go Lang, Java, and JavaScript (Typescript) (APIs, 2021). Figure A.7 shows an example of code of the SC.

```
import * as yup from 'yup';
import { ChaincodeTx } from '@worldsibu/convector-platform-fabric';
import {
  Controller,
  ConvectorController,
  Invokable,
  Param
} from '@worldsibu/convector-core';

import { DG, DGType } from './dg.model';
import { enum_to_array } from './utils'



@Controller('dg')
export class DGController extends ConvectorController<ChaincodeTx> {

  @Invokable()
  public async create(
  @Param(DG)
  dg: DG
  ) {
    if(this.tx.identity.getMSPID() != "dgproviderMSP")
    throw new Error("Only providers can create dangerous goods.");
    let is_existing = (await DG.getOne(dg.id)).id;
    if(!!is_existing)
    throw new Error("Dangerous Good with id "+dg.id+" is already existing.");
    await yup.string().oneOf(enum_to_array(DGType)).validate(dg.labelling);
    await dg.save();
  }

  // @Invokable()
  // public async treat(
  //   @Param(yup.string())
  //   dg_id: string
  // ) {
    // }

  @Invokable()
  public async get(
  @Param(yup.string())
  dg_id: string
  ) {
    let dg = await DG.getOne(dg_id);
    let is_existing = dg.id;
    if(!is_existing)
    throw new Error("Dangerous Good with id "+dg_id+" doesn't exist.");
    console.log(Object.keys(dg))
    return dg;
  }

```

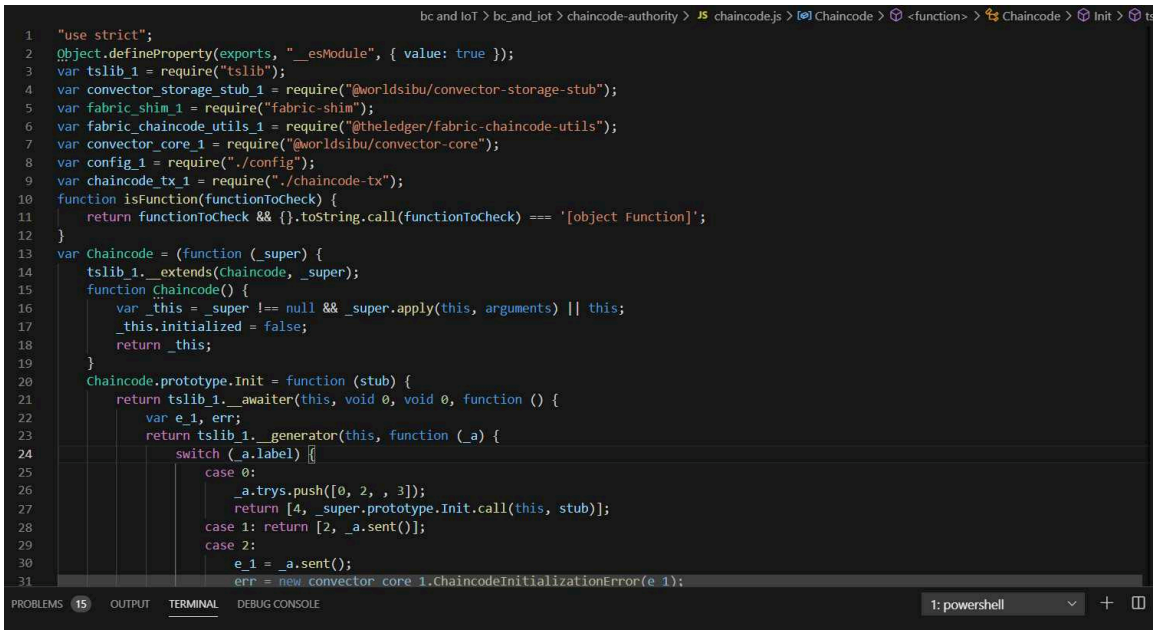

FIGURE A.7: The example of SC code expressed in JavaScript.

```
  @Invokable()
  public async get_all() {
    return await DG.getAll();
  }

  @Invokable()
  public async get_tx_history(
  @Param(yup.string())
  dg_id: string
  ) {
    let dg = await DG.getOne(dg_id);
    let is_existing = dg.id;
    if(!is_existing)
    throw new Error("Dangerous Good with id "+dg_id+" doesn't exist.");
    return await this.tx.stub.getHistoryForKeyAsList(dg_id);
  }

}
```

The code example of *"SC_Guard_Access"*.

```


package main

import (
"fmt"
"encoding/json"


"github.com/hyperledger/fabric-contract-api-go/contractapi"
)

type GuardContract struct {
  contractapi.Contract
}

type FunctionRule struct {
  CategoryEnabled []string `json:"categoryEnabled"`
```

```
var dossierInitialEvaluation = function (dossier_ID){
    var not_Comleted = function(dossier_ID)
    {
        list_of_document == [dossier_ID];
    }
    if(dossier_ID == not_Comleted)
    {
        var requireAdditionalDoccument  = function (dossier_ID, stakeholder_ID) {

            //Smart contract API ...
        }
    }

    else(registration == completed)
    {
        console.log("Registration comleted")

    }
```

FIGURE A.8: The simplified SC for stakeholder registration.

```
  MSPOrganizationEnabled []string `json:"mspOrganizationEnabled"`
}

type Function  map[string]*FunctionRule

/* RuleTable is an object that contains a list of function, in each function we
     set the list of the categories and the list
of the organization MSPs   */

type RuleTable struct {
  TableID string `json:"tableID"`
  Function Function `json:"function"`
}

/* CreateRuleTable is a function that create a table of the function rules */

/* this function takes in arguments:
-the key(ID) of the table
-the table name.

It returns an error in case of it can't create the table or if the table
    already exists    */

func (s *GuardContract) CreateRuleTable(ctx contractapi.
    TransactionContextInterface, key string, tableRule string ) error{

  // Check if the table already exists
  tableAsBytes, err := ctx.GetStub().GetState(key)
  if err != nil {
    return fmt.Errorf("failed to get table: %v", err)
  } else if tableAsBytes != nil {
    fmt.Println("table already exists: " + key )
    return fmt.Errorf("this table already exists: " + key)
  }

  // Save table in the world state
  err = ctx.GetStub().PutState(key, []byte(tableRule))
  if err != nil {
    return fmt.Errorf("Unable to interact with world state")
  }

  return nil
```

```
}

/*  AddRule is a function that add a function rule to a table of rules. A function rule in this case, it is resumed in a couple
(categories ,MSP organizations) wich defines the category requirement, that the a user should be at least a memebr of at least
one category, and the organizations that a user should be a member of at least once */

/*  This function takes in arguments:
-the table of rules name
-the function name that will add rules to it
-the category name that a user should satisfy to execute this function
-the organiazation MSP that a user should be a member of it in order to execute the function in question

This function returns an error if cannot get the table or it can't add the rule */

func (s *GuardContract) AddRule(ctx contractapi.TransactionContextInterface, tableName string, functionName string, category string, organizationMSP string  ) error{

  // get the table
  tableAsBytes, err := ctx.GetStub().GetState(tableName)
  if err != nil {
    return fmt.Errorf("failed to get table: %v", err)
  }

  //add rule
  var ruleTable RuleTable
  ruleTable.Function = make(Function)

  err = json.Unmarshal(tableAsBytes, &ruleTable)
  if err != nil {
    return fmt.Errorf("failed to unmarshal JSON: %v", err)
  }

  var newCategoryList []string
  var newMSPList []string
  if ruleTable.Function[functionName] == nil {
    newCategoryList = []string{category}
    newMSPList = []string{organizationMSP}
  } else {
    newCategoryList = append(ruleTable.Function[functionName].CategoryEnabled, category)
    newMSPList = append(ruleTable.Function[functionName].MSPOrganizationEnabled, organizationMSP)
  }


  ruleTable.Function[functionName] = &FunctionRule{newCategoryList,newMSPList}

  // Save the new table in the world state

  tableAsBytes, err = json.Marshal(ruleTable)
  err = ctx.GetStub().PutState(tableName, tableAsBytes)
  if err != nil {
    return fmt.Errorf("Unable to save the new tableRule")
  }
```

```
112     return nil
113   }
114
115   /*  CheckAccessToFunction is a function that check if the user who is calling a
          funtion in the table rule of funtions, satisfies
116   the requirements in order to execute this function  */
117
118   /*  this function takes in arguments:
119   - the function name
120   - the table name
121
122   this function returns:
123   - a boolean value (true or false) : it's the response of if this user who's
          calling this function can execute this function
124   or not
125   - an error : if it cannot get the MSP of the organization of the caller user or
          its category   */
126
127   func (s *GuardContract) CheckAccessToFunction(ctx contractapi.
          TransactionContextInterface, functionName string, tableName string ) (bool,
          error){
128     //Getting the client identity
129     MSPID, err := ctx.GetClientIdentity().GetMSPID()
130     if err != nil {
131       return false, fmt.Errorf("error getting MSP ID: %v", err)
132     }
133
134     category, foundCategory , err := ctx.GetClientIdentity().GetAttributeValue("
          category")
135     if err != nil {
136       return false,  fmt.Errorf("We cannot get the client category:  %v", err)
137     }
138
139     if foundCategory == false {
140       return false, fmt.Errorf("this client is not found in any category", err)
141     }
142     // Getting the table of rules
143     tableAsBytes, err := ctx.GetStub().GetState(tableName)
144     if err != nil {
145       return false, fmt.Errorf("failed to get table: %v", err)
146     }
147
148     var ruleTable RuleTable
149
150     err = json.Unmarshal(tableAsBytes, &ruleTable)
151     if err != nil {
152       return false, fmt.Errorf("failed to unmarshal JSON: %v", err)
153     }
154
155     _, categoryFound := Find(ruleTable.Function[functionName].CategoryEnabled,
          category)
156     _, mspfound := Find(ruleTable.Function[functionName].MSPOrganizationEnabled,
          MSPID)
157
158     if categoryFound && mspfound {
159       return true, nil
160     }
161
162     return false, nil
163
164   }
165
```

```
/* this is a find function that is used by the CheckAccessToFunction function
    to search a value in a slice */

func Find(slice []string, val string)  (int,bool) {
  for i, item := range slice {
    if item == val {
      return i, true
    }
  }
  return -1, false
}
```

A.11 Notification and Movement Documents

ANNEX IA

Notification document for transboundary movements/shipments of waste **EU**

1. Exporter - notifier Registration No:

Name:

Address:

Contact person:

Tel.: Fax:

E-mail:

2. Importer - consignee

Registration No:

Name:

Address:

Contact person:

Tel.: Fax:

E-mail:

8. Intended carrier(s)

Registration No:

Name (7):

Address:

Contact person:

Tel.: Fax:

E-mail:

Means of transport (5):

9. Waste generator(s)/producer(s) (1, 7, 8)

Registration No:

Name:

Address:

Contact person:

Tel.: Fax:

E-mail:

Site and process of generation (6):

10. Disposal facility (2): ☐ **or recovery facility** (2): ☐

Registration No:

Name:

Address:

Contact person:

Tel.: Fax:

E-mail:

Actual site of disposal/recovery:

3. Notification:

Notification concerning

A. (i) Individual shipment: ☐

(ii) Multiple shipments: ☐

B. (i) Disposal (1): ☐

(ii) Recovery: ☐

C. Pre-consented recovery facility (2, 3) Yes ☐ No ☐

4. Total intended number of shipments:

5. Total intended quantity (kg/litre) (4):

6. Intended period of time for shipment(s) (4):

First departure: Last departure:

7. Packaging type(s) (5):

Special handling requirements (6): Yes ☐ No ☐

11. Disposal/recovery operation(s) (2)

D code/R code (5):

Technology employed (6):

Reason for export (1, 6):

12. Designation and composition of the waste (6):

13. Physical characteristics (5):

14. Waste identification *(fill in relevant codes)*

(i) Basel Annex VIII (or IX if applicable):

(ii) OECD code (if different from (i)):

(iii) EC list of wastes:

(iv) National code in country of export:

(v) National code in country of import:

(vi) Other (specify):

(vii) Y-code:

(viii) H-code (5):

(ix) UN class (5):

(x) UN number:

(xi) UN shipping name:

(xii) Customs code(s) (HS):

15. Countries/States concerned (a), code No of competent authorities where applicable (b), specific points of exit or entry (c)

State of export/dispatch	State(s) of transit (entry and exit)						State of import/destination
(a)							
(b)							
(c)							

16. Customs offices of entry and/or exit and/or export (European Community):

Entry: Exit: Export:

17. Exporter's/notifier's - generator's/producer's (1) **declaration:**

I certify that the information is complete and correct to my best knowledge.

I also certify that legally enforceable written contractual obligations have been entered into and that any applicable insurance or other financial guarantee is or shall be in force covering the transboundary movement:

Exporter's/notifier's name: Signature Date

Generator's/producer's name: Signature Date

18. Number of annexes attached:

FOR USE BY COMPETENT AUTHORITIES

19. Acknowledgement from the relevant competent authority of countries of import - destination/transit (1) **/ export - dispatch** (9):

Country:

Notification received on:

Acknowledgement sent on:

Name of competent authority:

Stamp and/or signature:

20. Written consent (1, 8) **to the movement provided by the competent authority of** *(country)*:

Consent given on:

Consent valid from: until:

Specific conditions: No ☐ If Yes, see block 21 (6) ☐

Name of competent authority:

Stamp and/or signature:

21. Specific conditions on consenting to the movement or reasons for objecting:

(1) Required by the Basel Convention.
(2) In the case of an R12/R13 or D13-D15 operation, also attach corresponding information on the subsequent R1-R11 or D1-D12 facilit(y)ies when required.
(3) To be completed for movements within the OECD area and only if B(ii) applies.
(4) Attach detailed list if multiple shipments.
(5) See list of abbreviations and codes on the next page.
(6) Attach details if necessary.
(7) Attach list if more than one.
(8) If required by national legislation.
(9) If applicable under the OECD Decision.

List of abbreviations and codes used in the notification document

DISPOSAL OPERATIONS (block 11)

D 1 Deposit into or onto land (e.g. landfill, etc.)

D2 Land treatment (e.g. biodegradation of liquid or sludgy discards in soils, etc.)

D3 Deep injection (e.g. injection of pumpable discards into wells, salt domes or naturally occurring repositories, etc.)

D4 Surface impoundment (e.g. placement of liquid or sludge discards into pits, ponds or lagoons, etc.)

D5 Specially engineered landfill, (e.g. placement into lined discrete cells which are capped and isolated from one another and the environment, etc.)

D6 Release into a water body except seas/oceans

D7 Release into seas/oceans including sea-bed insertion

D8 Biological treatment not specified elsewhere in this list which results in final compounds or mixtures which are discarded by means of any of the operations in this list

D9 Physico-chemical treatment not specified elsewhere in this list which results in final compounds or mixtures which are discarded by means of any of the operations in this list (e.g. evaporation, drying, calcination, etc.)

D10 Incineration on land

D11 Incineration at sea

D12 Permanent storage (e.g. emplacement of containers in a mine, etc.)

D13 Blending or mixing prior to submission to any of the operations in this list

D14 Repackaging prior to submission to any of the operations in this list

D15 Storage pending any of the operations numbered in this list

RECOVERY OPERATIONS (block 11)

R1 Use as a fuel (other than in direct incineration) or other means to generate energy/use principally as a fuel or other means to generate energy

R2 Solvent reclamation/regeneration

R3 Recycling/reclamation of organic substances which are not used as solvents

R4 Recycling/reclamation of metals and metal compounds

R5 Recycling/reclamation of other inorganic materials

R6 Regeneration of acids or bases

R7 Recovery of components used for pollution abatement

R8 Recovery of components from catalysts

R9 Used oil re-refining or other reuses of previously used oil

R10 Land treatment resulting in benefit to agriculture or ecological improvement

R11 Uses of residual materials obtained from any of the operations numbered R1 to R10

R12 Exchange of wastes for submission to any of the operations numbered R1 to R11

R13 Accumulation of material intended for any operation in this list.

PACKAGING TYPES (block 7)

1. Drum
2. Wooden barrel
3. Jerrican
4. Box
5. Bag
6. Composite packaging
7. Pressure receptacle
8. Bulk
9. Other (specify)

MEANS OF TRANSPORT (block 8)

R = Road

T = Train/rail

S = Sea

A = Air

W = Inland waterways

PHYSICAL CHARACTERISTICS (block 13)

1. Powdery/powder
2. Solid
3. Viscous/paste
4. Sludgy
5. Liquid
6. Gaseous
7. Other (specify)

H CODE AND UN CLASS (block 14)

UN Class	H code	Characteristics
1	H1	Explosive
3	H3	Flammable liquids
4.1	H4.1	Flammable solids
4.2	H4.2	Substances or wastes liable to spontaneous combustion
4.3	H4.3	Substances or wastes which, in contact with water, emit flammable gases
5.1	H5.1	Oxidising
5.2	H5.2	Organic peroxides
6.1	H6.1	Poisonous (acute)
6.2	H6.2	Infectious substances
8	H8	Corrosives
9	H10	Liberation of toxic gases in contact with air or water
9	H11	Toxic (delayed or chronic)
9	H12	Ecotoxic
9	H13	Capable, by any means, after disposal of yielding another material, e.g. leachate, which possesses any of the characteristics listed above

Further information, in particular related to waste identification (block 14), i.e. on Basel Annexes VIII and IX codes, OECD codes and Y codes, can be found in a Guidance/Instruction Manual available from the OECD and the Secretariat of the Basel Convention.

ANNEX IB

Movement document for transboundary movements/shipments of EU waste **EU**

1. Corresponding to notification No:	**2. Serial/total number of shipments:**
3. Exporter - notifier Registration No: Name: Address: Contact person: Tel.: Fax: E-mail:	**4. Importer - consignee** Registration No: Name: Address: Contact person: Tel.: Fax: E-mail:
5. Actual quantity: kg: litre:	**6. Actual date of shipment:**
7. Packaging type(s) *(1)*: Number of packages: **Special handling requirements**: *(2)* Yes ☐ No ☐	

8 (a)1tst carrier *(3)*: Registration No: Name: Address: Tel.: Fax: E-mail:	**8 b) 2nd carrier:** Registration No: Name: Address: Tel.: Fax: E-mail:	**8 c) Last carrier:** Registration No: Name: Address: Tel.: Fax: E-mail:
- - - - - - - ***To be completed by carrier's representative*** - - - - - - -		*More than three carriers (2)* ☐
Means of transport *(1)*: Date of transfer: Signature:	Means of transport *(1)*: Date of transfer: Signature:	Means of transport *(1)*: Date of transfer: Signature:

9. Waste generator(s)/producer(s) *(4;5;6)*: Registration No: Name: Address: Contact person: Tel.: Fax: E-mail: Site of generation *(2)*:	**12. Designation and composition of the waste** *(2)*:
	13. Physical characteristics *(1)*:
10. Disposal facility ☐ **or recovery facility** ☐ Registration No: Name: Address: Contact person: Tel.: Fax: E-mail: Actual site of disposal/recovery *(2)*	**14. Waste identification** *(fill in relevant codes)* (i) Basel Annex VIII (or IX if applicable): (ii) OECD code (if different from (i)): (iii) EC list of wastes: (iv) National code in country of export: (v) National code in country of import: (vi) Other (specify): (vii) Y code: (viii) H code *(1)*: (ix) UN class *(1)*: (x) UN number: (xi) UN shipping name: (xii) Customs code(s) (HS):
11. Disposal/recovery operation(s) D code/R code *(1)*:	

15. Exporter's - notifier's/generator's/producer's *(4)* **declaration:**

I certify that the above information is complete and correct to my best knowledge. I also certify that legally enforceable written contractual obligations have been entered into, that any applicable insurance or other financial guarantee is in force covering the transboundary movement and that all necessary consents have been received from the competent authorities of the countries concerned.

Name: Signature:

Date:

16. For use by any person involved in the transboundary movement in case additional information is required:

TO BE COMPLETED BY DISPOSAL /RECOVERY FACILITY

17. Shipment received at disposal facility ☐ **or recovery facility** ☐

Date of reception: Accepted: ☐ Rejected*: ☐

Quantity received: kg: litre:

* *immediately contact competent authorities*

Approximate date of disposal/recovery:

Disposal/recovery operation (1):

Date:

Name:

Signature:

18. I certify that the disposal/recovery of the waste described above has been completed.

Date:

Name:

Signature and stamp:

(1) See list of abbreviations and codes on the next page.
(2) Attach details if necessary.
(3) If more than three carriers, attach information as required in blocks 8 (a,b,c).
(4) Required by the Basel Convention.
(5) Attach list if more than one.
(6) If required by national legislation.

FOR USE BY CUSTOMS OFFICES (if required by national legislation)	
19. COUNTRY OF EXPORT - DISPATCH OR CUSTOMS OFFICE OF EXIT The waste described in this movement document left the country on: Signature: Stamp:	**20. COUNTRY OF IMPORT - DESTINATION OR CUSTOMS OFFICE OF ENTRY** The waste described in this movement document entered the country on: Signature: Stamp:

21. STAMPS OF CUSTOMS OFFICES OF TRANSIT COUNTRIES			
Name of country:		Name of country:	
Entry:	Exit:	Entry:	Exit:
Name of country:		Name of country:	
Entry:	Exit:	Entry:	Exit:

List of abbreviations and codes used in the movement document

DISPOSAL OPERATIONS (block 11)

D 1 Deposit into or onto land (e.g. landfill, etc.)
D 2 Land treatment (e.g. biodegradation of liquid or sludgy discards in soils, etc.)
D 3 Deep injection (e.g. injection of pumpable discards into wells, salt domes or naturally occurring repositories, etc.)
D 4 Surface impoundment (e.g. placement of liquid or sludge discards into pits, ponds or lagoons, etc.)
D 5 Specially engineered landfill (e.g. placement into lined discrete cells which are capped and isolated from one another and the environment)
D 6 Release into a water body except seas/oceans
D 7 Release into seas/oceans including sea-bed insertion
D 8 Biological treatment not specified elsewhere in this list which results in final compounds or mixtures which are discarded by means of any of the operations in this list
D 9 Physico-chemical treatment not specified elsewhere in this list which results in final compounds or mixtures which are discarded by means of any of the operations in this list (e.g. evaporation, drying, calcination)
D 10 Incineration on land
D 11 Incineration at sea
D 12 Permanent storage (e.g. emplacement of containers in a mine, etc.)
D 13 Blending or mixing prior to submission to any of the operations in this list
D 14 Repackaging prior to submission to any of the operations in this list
D 15 Storage pending any of the operations in this list

Recovery operations (block 11)

R 1 Use as a fuel (other than in direct incineration) or other means to generate energy/Use principally as a fuel or other means to generate energy
R 2 Solvent reclamation/regeneration
R 3 Recycling/reclamation of organic substances which are not used as solvents
R 4 Recycling/reclamation of metals and metal compounds
R 5 Recycling/reclamation of other inorganic materials
R 6 Regeneration of acids or bases
R 7 Recovery of components used for pollution abatement
R 8 Recovery of components from catalysts
R 9 Used oil re-refining or other reuses of previously used oil
R 10 Land treatment resulting in benefit to agriculture or ecological improvement
R 11 Uses of residual materials obtained from any of the operations numbered R 1 to R 10
R 12 Exchange of wastes for submission to any of the operations numbered R 1 to R 11
R 13 Accumulation of material intended for any operation in this list

PACKAGING TYPES (block 7)

1. Drum
2. Wooden barrel
3. Jerrican
4. Box
5. Bag
6. Composite packaging
7. Pressure receptacle
8. Bulk
9. Other (specify)

MEANS OF TRANSPORT (block 8)

R = road T = train/rail
S = sea A = air
W = inland waterways

PHYSICAL CHARACTERISTICS (block 13)

1. Powdery/powder
2. Solid
3. Viscous/paste
4. Sludgy
5. Liquid
6. Gaseous
7. Other (specify)

H CODE AND UN CLASS (block 14)

UN Class	H code	Characteristics
1	H1	Explosive
3	H3	Flammable liquids
4.1	H4.1	Flammable solids
4.2	H4.2	Substances or wastes liable to spontaneous combustion
4.3	H4.3	Substances or wastes which, in contact with water, emit flammable gases
5.1	H5.1	Oxidising
5.2	H5.2	Organic peroxides
6.1	H6.1	Poisonous (acute)
6.2	H6.2	Infectious substances
8	H8	Corrosives
9	H10	Liberation of toxic gases in contact with air or water
9	H11	Toxic (delayed or chronic)
9	H12	Ecotoxic
9	H13	Capable, by any means, after disposal of yielding another material, e.g. leachate, which possesses any of the characteristics listed above

Further information, in particular related to waste identification (block 14), i.e. on Basel Annexes VIII and IX codes, OECD codes and Y codes, can be found in a Guidance/Instruction Manual available from the OECD and the Secretariat of the Basel Convention.

```
adnan@adnan-VirtualBox:~/Desktop/Fabric/fabric-samples/test-network$ ./network.sh up
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb'
LOCAL_VERSION=2.2.2
DOCKER_IMAGE_VERSION=2.2.2
```

```
adnan@adnan-VirtualBox:~/Desktop/Fabric/fabric-samples/test-network$ ./network.sh createChannel -c ChannelAB
Creating channel 'ChannelAB'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb
Generating channel create transaction 'ChannelAB.tx'
+ configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/ChannelAB.tx -channelID ChannelAB
2021-03-20 20:21:21.509 CET [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-03-20 20:21:21.527 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/adnan/Desktop/Fab
2021-03-20 20:21:21.527 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
2021-03-20 20:21:21.530 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
+ res=0
Creating channel ChannelAB
Using organization 1
```

```
adnan-VirtualBox:~/Desktop/Fabric/fabric-samples/test-network$ ./network.sh createChannel -c ChannelBC
ng channel 'ChannelBC'.
work is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb
ting channel create transaction 'ChannelBC.tx'
igtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/ChannelBC.tx -channelID ChannelBC
3-20 20:22:24.604 CET [common.tools.configtxgen] main -> INFO 001 Loading configuration
3-20 20:22:24.624 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/adnan/Desktop/Fabric/fa
3-20 20:22:24.624 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
3-20 20:22:24.626 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
0
ng channel ChannelBC
```

```
adnan@adnan-VirtualBox:~/Desktop/Fabric/fabric-samples/test-network$ ./network.sh createChannel -c GlobalChannel
Creating channel 'GlobalChannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb
Generating channel create transaction 'GlobalChannel.tx'
+ configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/GlobalChannel.tx -channelID GlobalChannel
2021-03-20 20:20:33.415 CET [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-03-20 20:20:33.437 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/adnan/Desktop/Fabric/
2021-03-20 20:20:33.437 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
2021-03-20 20:20:33.439 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
+ res=0
Creating channel GlobalChannel
```

FIGURE A.9: The HF commands for creation of channels.

Bibliography

Abdellatif, Tesnim and Kei-Leo Brousmiche (2018). "Formal Verification of Smart Contracts Based on Users and Blockchain Behaviors Models". In: *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. Paris: IEEE, pp. 1–5. ISBN: 978-1-5386-3662-6. DOI: 10.1109/NTMS.2018.8328737. URL: http://ieeexplore.ieee.org/document/8328737/.

Abeyratne, Saveen A and Radmehr P. Monfared (2016). "Blockchain Ready Manufacturing Supply Chain Using Distributed Ledger". In: *International Journal of Research in Engineering and Technology* 05.09, pp. 1–10. ISSN: 23217308. DOI: 10.15623/ijret.2016.0509001. URL: https://ijret.org/volumes/2016v05/i09/IJRET20160509001.pdf.

ADN, UNECE (2016). "European Agreement concerning the international carriage of dangerous goods by inland waterways". In: *UN*. New York-Geneva. ISBN: 978-92-1-139157-2 978-92-1-058306-0. DOI: https://doi.org/10.18356/868dfa06-en. URL: https://www.un-ilibrary.org/international-law-and-justice/european-agreement-concerning-the-international-carriage-of-dangerous-goods-by-inland-waterways-adn-2019_868dfa06-en.

Ahmed, Mohamed et al. (2020). "Enhancing B2B supply chain traceability using smart contracts and IoT". In: *Hamburg International Conference of Logistics (HICL)*. Vol. 29. Hamburg University of Technology (TUHH), Institute of Business Logistics and General Management, pp. 559–589. URL: https://ideas.repec.org/h/zbw/hiclch/228933.html.

Ahmed-Rengers, Mansoor and Kari Kostiainen (2020). "Don't Mine, Wait in Line: Fair and Efficient Blockchain Consensus with Robust Round Robin". In: *arXiv:1804.07391*. URL: https://ui.adsabs.harvard.edu/abs/2018arXiv180407391A/abstract.

Al-Breiki, Hamda et al. (2020). "Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges". In: *IEEE Access* 8, pp. 85675–85685. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2992698. URL: https://ieeexplore.ieee.org/document/9086815/.

Al-Riyami, Sattam S. and Kenneth G. Paterson (2003). "Certificateless Public Key Cryptography". In: *Advances in Cryptology - ASIACRYPT*. Vol. 2894. Springer Berlin Heidelberg,

pp. 452–473. ISBN: 978-3-540-20592-0 978-3-540-40061-5. DOI: 10.1007/978-3-540-40061-5_29. URL: http://link.springer.com/10.1007/978-3-540-40061-5_29.

Al-Sarawi, S. et al. (2017). "Internet of Things (IoT) communication protocols: Review". In: *8th International Conference on Information Technology (ICIT)*, pp. 685–690. DOI: 10.1109/ICITECH.2017.8079928.

Albert, Elvira et al. (2020). "GASOL: Gas Analysis and Optimization for Ethereum Smart Contracts". In: *Tools and Algorithms for the Construction and Analysis of Systems*. Springer, pp. 118–125. ISBN: 978-3-030-45236-0 978-3-030-45237-7. DOI: 10.1007/978-3-030-45237-7_7. URL: http://link.springer.com/10.1007/978-3-030-45237-7_7.

Alharby, Maher and Aad van Moorsel (2017). "Blockchain Based Smart Contracts : A Systematic Mapping Study". In: *Computer Science & Information Technology (CS & IT)*. Academy & Industry Research Collaboration Center (AIRCC), pp. 125–140. ISBN: 978-1-921987-70-0. DOI: 10.5121/csit.2017.71011. URL: http://airccj.org/CSCP/vol7/csit77211.pdf.

Ali, Toqeer et al. (2019). "A Comparative Analysis of Blockchain Architecture and Its Applications: Problems and Recommendations". In: *IEEE Access*, pp. 1–32. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2957660. URL: https://ieeexplore.ieee.org/document/8922632/.

Amani, Sidney et al. (2018). "Towards verifying ethereum smart contract bytecode in Isabelle/HOL". In: *7th ACM SIGPLAN International Conference on Certified Programs and Proofs - CPP*. Los Angeles, USA, pp. 66–77. ISBN: 978-1-4503-5586-5. DOI: 10.1145/3167084. URL: http://dl.acm.org/citation.cfm?doid=3176245.3167084.

Ammar, Mahmoud, Giovanni Russello, and Bruno Crispo (2018). "Internet of Things: A survey on the security of IoT frameworks". In: *Journal of Information Security and Applications*, pp. 8–27. ISSN: 22142126. DOI: 10.1016/j.jisa.2017.11.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S2214212617302934.

Amoussou-Guenou, Yackolley et al. (2019). "Dissecting Tendermint". In: *Networked Systems*. Springer, pp. 166–182. ISBN: 978-3-030-31277-0. DOI: 10.1007/978-3-030-31277-0_11.

Anceaume, Emmanuelle et al. (2019). "Blockchain Abstract Data Type". In: *31st ACM Symposium on Parallelism in Algorithms and Architectures*. New York, USA, pp. 349–358. ISBN:

978-1-4503-6184-2. DOI: 10.1145/3323165.3323183. URL: https://doi.org/10.1145/3323165.3323183.

Androulaki, Elli et al. (2018). "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains". In: *Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15. DOI: https://doi.org/10.1145/3190508.3190538. URL: https://dl.acm.org/doi/10.1145/3190508.3190538.

Anjum, A., M. Sporny, and A. Sill (2017a). "Blockchain Standards for Compliance and Trust". In: *IEEE Cloud Computing* 4.4, pp. 84–90. DOI: 10.1109/MCC.2017.3791019.

Anjum, Ashiq, Manu Sporny, and Alan Sill (2017b). "Blockchain Standards for Compliance and Trust". In: *IEEE Cloud Computing*, pp. 84–90. ISSN: 2325-6095. DOI: 10.1109/MCC.2017.3791019. URL: http://ieeexplore.ieee.org/document/8066010/.

Anshel, Iris, Michael Anshel, and Dorian Goldfeld (1999). "An algebraic method for public-key cryptography". In: *Mathematical Research Letters*, pp. 287–291. ISSN: 10732780, 1945001X. DOI: 10.4310/MRL.1999.v6.n3.a3. URL: http://www.intlpress.com/site/pub/pages/journals/items/mrl/content/vols/0006/0003/a003/.

Antonopoulos, Andreas M (2017). *Mastering Bitcoin: Programming the open blockchain*. " O'Reilly Media, Inc."

APIs, Contract (2021). *Fabric Contract APIs and Application SDKs — hyperledger-fabricdocs main documentation*. URL: https://hyperledger-fabric.readthedocs.io/en/latest/sdk_chaincode.html.

Argañaraz, Mauro, Ana Funes, and Aristides Dasso (2010). "An MDA Approach to Business Process Model Transformations". In: *Electronic Journal of SADIO (EJS)*, pp. 24–48. ISSN: 1514-6774. URL: https://publicaciones.sadio.org.ar/index.php/EJS/article/view/77.

Arluck, Jacob (2018). *Liquid Proof-of-Stake*. URL: https://medium.com/tezos/liquid-proof-of-stake-aec2f7ef1da7.

Assets (2021). *Hyperledger Fabric Model. Secured asset transfer in Fabric*. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.2/secured_asset_transfer/secured_private_asset_transfer_tutorial.html.

Atkinson, C. and T. Kuhne (2003). "Model-driven development: a metamodeling foundation". In: *IEEE Software*, pp. 36–41. ISSN: 0740-7459. DOI: 10.1109/MS.2003.1231149. URL: http://ieeexplore.ieee.org/document/1231149/.

Atzei, Nicola, Massimo Bartoletti, and Tiziana Cimoli (2017). "A Survey of Attacks on Ethereum Smart Contracts SoK". In: *6th International Conference on Principles of Security and Trust*, pp. 164–186. ISBN: 978-3-662-54454-9. DOI: 10.1007/978-3-662-54455-6_8. URL: http://doi.org/10.1007/978-3-662-54455-6_8.

Atzori, Marcella (2017). "Blockchain Governance and the Role of Trust Service Providers: The TrustedChainn Network". In: *SSRN Electronic Journal*. ISSN: 1556-5068. DOI: 10.2139/ssrn.2972837. URL: http://www.ssrn.com/abstract=2972837.

AWS IoT (n.d.). URL: https://aws.amazon.com/iot/.

Bach, L. M., B. Mihaljevic, and M. Zagar (2018). "Comparative analysis of blockchain consensus algorithms". In: *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1545–1550. DOI: 10.23919/MIPRO.2018.8400278.

Baclawski, Kenneth et al. (2001). "Extending UML to Support Ontology Engineering for the Semantic Web". In: *The Unified Modeling Language. Modeling Languages, Concepts, and Tools*. Lecture Notes in Computer Science. Springer, pp. 342–360. ISBN: 978-3-540-45441-0. DOI: 10.1007/3-540-45441-1_26.

Badzar, Amina (2016). "Blockchain for securing sustainable transport contracts and supply chain transparency - An explorative study of blockchain technology in logistics". In: URL: http://lup.lub.lu.se/student-papers/record/8880383.

Bai, Xiaomin et al. (2018). "Formal Modeling and Verification of Smart Contracts". In: *7th International Conference on Software and Computer Applications - ICSCA*. Kuantan, Malaysia: ACM Press, pp. 322–326. ISBN: 978-1-4503-5414-1. DOI: 10.1145/3185089.3185138. URL: http://dl.acm.org/citation.cfm?doid=3185089.3185138.

Baier, Christel and Joost-Pieter Katoen (2008). *Principles of model checking*. Cambridge, Mass: The MIT Press, p. 994. ISBN: 978-0-262-02649-9.

Bajec, Marko and Marjan Krisper (2005). "A methodology and tool support for managing business rules in organisations". In: *Information Systems* 30.6, pp. 423–443. ISSN: 03064379. DOI: 10.1016/j.is.2004.05.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306437904000481.

Baliga, Arati et al. (2018). "Performance Evaluation of the Quorum Blockchain Platform". In: *arXiv:1809.03421 [cs]*. URL: http://arxiv.org/abs/1809.03421.

Bamakan, Seyed Mojtaba Hosseini, Amirhossein Motavali, and Alireza Babaei Bondarti (2020). "A survey of blockchain consensus algorithms performance evaluation criteria". In: *Expert Systems with Applications* 154. ISSN: 09574174. DOI: https://doi.org/10.1016/j.eswa.2020.113385. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417420302098.

Bartoletti, Massimo, Stefano Lande, and Alessandro Sebastian Podda (2017). "A Proof-of-Stake protocol for consensus on Bitcoin subchains". In: *International Conference on Financial Cryptography and Data Security*. Springer, pp. 568–584. URL: https://link.springer.com/chapter/10.1007/978-3-319-70278-0_36.

Bashir, Imran (2017). *Mastering blockchain*. Packt Publishing Ltd.

Beck, Roman et al. (2016). "Blockchain-the Gateway to Trust-Free Cryptographic Transactions." In: *European Conference on Information Systems (ECIS)*. URL: https://aisel.aisnet.org/ecis2016_rp/153.

Benekos, I. and D. Diamantidis (2017). "On risk assessment and risk acceptance of dangerous goods transportation through road tunnels in Greece". In: *Safety Science* 91, pp. 1–10. ISSN: 09257535. DOI: 10.1016/j.ssci.2016.07.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925753516301473.

Bentov, Iddo et al. (2014). "Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake". In: *ACM SIGMETRICS Performance Evaluation Review*, pp. 34–37. ISSN: 0163-5999. DOI: 10.1145/2695533.2695545. URL: http://doi.org/10.1145/2695533.2695545.

Berson, Alex (1996). *Client/server architecture*. McGraw-Hill, Inc. URL: https://dl.acm.org/doi/abs/10.5555/227054.

Bhargavan, Karthikeyan et al. (2016). "Formal Verification of Smart Contracts: Short Paper". In: *Workshop on Programming Languages and Analysis for Security - PLAS*. Vienna, Austria: ACM Press, pp. 91–96. ISBN: 978-1-4503-4574-3. DOI: 10.1145/2993600.2993611. URL: http://dl.acm.org/citation.cfm?doid=2993600.2993611.

Biggs, Jonathan et al. (2017). "Blockchain: Revolutionizing the Global Supply Chain by Building Trust and Transparency". In: Rutgers University: Camden, NJ, USA, p. 26.

Biryukov, Alex, Dmitry Khovratovich, and Ivan Pustogarov (2014). "Deanonymisation of Clients in Bitcoin P2P Network". In: *Conference on Computer and Communications Security*. Scottsdale Arizona USA: ACM, pp. 15–29. ISBN: 978-1-4503-2957-6. DOI: 10.1145/

2660267.2660379. URL: https://dl.acm.org/doi/10.1145/2660267.2660379.

BitcoinWiki (2018). *Majority attack*. URL: https://en.bitcoin.it/wiki/Majority_attack.

Blog, Hyperledger (2020). *A Blockchain Platform for the Enterprise — hyperledger-fabricdocs master documentation*. URL: https://hyperledger-fabric.readthedocs.io/en/release-1.4/.

Borrie, Helen (2004). "Introduction to Client/Server Architecture". In: *The Firebird Book: A Reference for Database Developers*. Berkeley, CA: Apress, pp. 75–84. ISBN: 978-1-4302-0743-6. DOI: 10.1007/978-1-4302-0743-6_5. URL: https://doi.org/10.1007/978-1-4302-0743-6_5.

Bouzidi, Aljia et al. (2020). "From BPMN to Sequence Diagrams: Transformation and Traceability". In: *15th International Conference on Evaluation of Novel Approaches to Software Engineering*. ISBN: 978-989-758-421-3. DOI: 10.5220/0009418104380445. URL: https://www.scitepress.org/Link.aspx?doi=10.5220/0009418104380445.

Brahim, Bousetta, El Beggar Omar, and Gadi Taoufiq (2013). "A methodology for CIM modelling and its transformation to PIM". In: *Journal of Information Engineering and Applications*, p. 1. ISSN: 2225-0506. URL: https://www.iiste.org/Journals/index.php/JIEA/article/view/4350.

Brocke, Jan vom and Michael Rosemann (2015). "Business Process Management". In: *Wiley Encyclopedia of Management*. Chichester, UK: John Wiley & Sons, Ltd, pp. 1–9. ISBN: 978-1-118-78531-7 978-1-119-97251-8. DOI: 10.1002/9781118785317.weom070213. URL: http://doi.wiley.com/10.1002/9781118785317.weom070213.

Brown, Alan W. (2004). "Model driven architecture: Principles and practice". In: *Software and Systems Modeling*. ISSN: 1619-1366, 1619-1374. DOI: 10.1007/s10270-004-0061-2. URL: http://link.springer.com/10.1007/s10270-004-0061-2.

Brown, Richard Gendal et al. (2016). "Corda: An Introduction". In: White Paper. DOI: 10.13140/RG.2.2.30487.37284. URL: http://rgdoi.net/10.13140/RG.2.2.30487.37284.

Buterin (2017). "A Next Generation Smart Contract and Decentralized Application Platform". In: White Paper, pp. 1–36. URL: https://translatewhitepaper.com/wp-content/uploads/2021/04/EthereumOrijinal-ETH-English.pdf.

Cachin, C. (2016). "Architecture of the Hyperledger Blockchain Fabric". In: *IBM Research*. URL: https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf.

Cae-Imt-Inria (2021). *Les Verrous Technologiques Des Blockchains. ISBN : 978-2-11-162212-8*. URL: https://www.entreprises.gouv.fr/files/files/etudes-et-statistiques/rapport-final-blockchain.pdf.

Carrefour (2019). *Carrefour-blockchain*. URL: https://www.carrefour.com/en/newsroom/carrefour-launches-europes-first-food-blockchain.

Cassez, Franck et al., eds. (2001). *Modeling and Verification of Parallel Processes*. Lecture Notes in Computer Science. Springer. ISBN: 978-3-540-42787-2. DOI: 10.1007/3-540-45510-8. URL: http://www.springer.com/gp/book/9783540427872.

Castillo, Michael del (2016). *The DAO Attacked: Code Issue Leads to $60 Million Ether Theft*. en-US. Publication Title: CoinDesk. URL: https://www.coindesk.com/dao-attacked-code-issue-leads-60-million-ether-theft.

Castro, Miguel and Barbara Liskov (1999). "Practical Byzantine Fault Tolerance". In: *Third Symposium on Operating Systems Design and Implementation*, pp. 173–186. URL: http://pmg.csail.mit.edu/papers/osdi99.pdf.

Chandra, Charu and Jānis Grabis (2007). *Supply Chain Configuration: Concepts, Solutions, and Applications*. Springer US. ISBN: 978-1-4419-3778-0. DOI: 10.1007/978-0-387-68155-9. URL: https://www.springer.com/gp/book/9781441937780.

Chang, Yanling, Eleftherios Iakovou, and Weidong Shi (2020). "Blockchain in global supply chains and cross border trade: a critical synthesis of the state-of-the-art, challenges and opportunities". In: *International Journal of Production Research* 58.7, pp. 2082–2099. ISSN: 0020-7543, 1366-588X. DOI: 10.1080/00207543.2019.1651946. URL: https://www.tandfonline.com/doi/full/10.1080/00207543.2019.1651946.

Chase, Brad and Ethan MacBrough (2018). "Analysis of the XRP Ledger Consensus Protocol". In: arXiv: 1802.07242. URL: http://arxiv.org/abs/1802.07242.

Chiswell, Ian and Wilfrid Hodges (2007). *Mathematical logic*. Oxford texts in logic. London: Oxford University Press. ISBN: 978-0-19-921562-1 978-0-19-857100-1.

Cho, Hyun, Jeff Gray, and Eugene Syriani (2012). "Creating visual Domain-Specific Modeling Languages from end-user demonstration". In: *4th International Workshop on Modeling in Software Engineering (MISE)*. Zurich, Switzerland: IEEE, pp. 22–28. ISBN: 978-1-4673-1757-3

978-1-4673-1756-6. DOI: 10.1109/MISE.2012.6226010. URL: http://ieeexplore.ieee.org/document/6226010/.

Christidis, Konstantinos and Michael Devetsikiotis (2016). "Blockchains and Smart Contracts for the Internet of Things". In: *IEEE Access* 4, pp. 2292–2303. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2566339.

Christopher, Martin (2011). *Logistics & supply chain management*. 4th Edition. Harlow: Financial Times, Prentice Hall. ISBN: 978-0-273-73112-2.

Cimatti, Alessandro et al. (2000). "NUSMV: a new symbolic model checker". In: *International Journal on Software Tools for Technology Transfer (STTT)* 2.4, pp. 410–425. ISSN: 1433-2779, 1433-2787. DOI: 10.1007/s100090050046. URL: http://link.springer.com/10.1007/s100090050046.

Clarke, Edmund M (1999). "Model Checking II Temporal Logic Model Checking". In: School of Computer Science, Carnegie Mellon University, p. 32. URL: https://www.cs.cmu.edu/~emc/15817-f09/lecture2.pdf.

Clarke, Edmund M. (2001). *Model Checking Overview*. URL: http://www.cs.cmu.edu/~emc/15-398/lectures/overview.pdf.

Clincy, V. and H. Shahriar (2019). "Blockchain Development Platform Comparison". In: *43rd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 1. ISSN: 0730-3157, pp. 922–923. DOI: 10.1109/COMPSAC.2019.00142.

Cole, Rosanna, Mark Stevenson, and James Aitken (2019). "Blockchain technology: implications for operations and supply chain management". In: *Supply Chain Management: An International Journal* 24.4. Publisher: Emerald Publishing Limited, pp. 469–483. ISSN: 1359-8546. DOI: 10.1108/SCM-09-2018-0309. URL: https://doi.org/10.1108/SCM-09-2018-0309.

Collins, Michael (1998). *Formal Methods*. URL: https://users.ece.cmu.edu/~koopman/des_s99/formal_methods/#targetText=Formal%20methods%20are%20system%20design,order%20to%20ensure%20correct%20behavior..

Commission, European (2008). *Directive 2008/98/EC on Waste (Waste Framework Directive) - Environment - European Commission*. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX\%3A02008L0098-20180705.

Commission, European and Parliament (2006). *Règlement (Ce) No 1013/2006 Du Parlement Européen Et Du Conseil*. URL: https://eur-lex.europa.eu/legal-content/FR/TXT/PDF/?uri=CELEX:32006R1013&from=EN.

Conca, Andrea, Chiara Ridella, and Enrico Sapori (2016). "A Risk Assessment for Road Transportation of Dangerous Goods: A Routing Solution". In: *Transportation Research Procedia* 14, pp. 2890–2899. ISSN: 23521465. DOI: 10.1016/j.trpro.2016.05.407. URL: https://linkinghub.elsevier.com/retrieve/pii/S2352146516304136.

Conoscenti, Marco, Antonio Vetrò, and Juan Carlos De Martin (2016). "Blockchain for the Internet of Things: A systematic literature review". In: *13th International Conference of Computer Systems and Applications (AICCSA)*. ISSN: 2161-5330, pp. 1–6. DOI: 10.1109/AICCSA.2016.7945805.

Coq (n.d.). URL: https://coq.inria.fr/.

Coy, Steven P (2008). "Security Implications of the Choice of Distributed Database Management System Model: Relational vs. Object-Oriented". In: *University of Maryland*, pp. 428–437. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.7784&rep=rep1&type=pdf.

Cranefield, Stephen (2006). "Networked Knowledge Representation and Exchange using UML and RDF". In: *Journal of Digital Information* 1.8. Number: 8. ISSN: 1368-7506. URL: https://journals.tdl.org/jodi/index.php/jodi/article/view/30.

Croneri (2020). *Transport of Dangerous Goods*. URL: https://app.croneri.co.uk/topics/transport-dangerous-goods/indepth.

Dabbagh, Mohammad, Mohsen Kakavand, and Mohammad Tahir (2020). "Towards Integration of Blockchain and IoT: A Bibliometric Analysis of State-of-the-Art". In: *Blockchain and Applications*. Vol. 1010. Springer, pp. 27–35. ISBN: 978-3-030-23812-4 978-3-030-23813-1. DOI: 10.1007/978-3-030-23813-1_4. URL: http://link.springer.com/10.1007/978-3-030-23813-1_4.

Dang, Quynh H. (2015). *Secure Hash Standard*. Tech. rep. NIST FIPS 180-4. National Institute of Standards and Technology. DOI: 10.6028/NIST.FIPS.180-4. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

De Angelis, Stefano et al. (2018). "PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain". In: *Italian Conference on Cyber Security, Milan, Italy*, pp. 1–11. URL: https://eprints.soton.ac.uk/415083/.

Deloitte (2019). *Deloitte 2019 Global Blockchian Survey. Deloitte Insights*. URL: https://www2.deloitte.com/content/dam/Deloitte/se/Documents/risk/DI_2019-global-blockchain-survey.pdf.

Demertzis, Konstantinos et al. (2020). "Anomaly detection via blockchained deep learning smart contracts in industry 4.0". In: *Neural Computing and Applications*. ISSN: 0941-0643, 1433-3058. DOI: 10.1007/s00521-020-05189-8. URL: http://link.springer.com/10.1007/s00521-020-05189-8.

DGAssistant (n.d.). *DGAssistant*. URL: https://www.dgassistant.com/en/index.aspx.

Dib, Omar et al. (2018). "Consortium Blockchains: Overview, Applications and Challenges". In: *International Journal On Advances in Telecommunications* 11.1, p. 15. URL: https://hal.archives-ouvertes.fr/hal-02271063.

Dillon, T., C. Wu, and E. Chang (2010). "Cloud Computing: Issues and Challenges". In: *24th IEEE International Conference on Advanced Information Networking and Applications*. ISSN: 2332-5658, pp. 27–33. DOI: 10.1109/AINA.2010.187.

Ding, Lianhong, Yifan Chen, and Juntao Li (2016). "Monitoring Dangerous Goods in Container Yard Using the Internet of Things". In: *Scientific Programming* 2016, pp. 1–12. ISSN: 1058-9244, 1875-919X. DOI: 10.1155/2016/5083074. URL: https://www.hindawi.com/journals/sp/2016/5083074/.

DMTF (2012). "Common Information Model (CIM) Infrastructure". In: Document Number: DSP0004 2.7.0, p. 186. URL: https://www.dmtf.org/sites/default/files/standards/documents/DSP0004_2.7.0.pdf.

DMTF, CIM (2020). *Common Information Model*. 2020. URL: https://www.dmtf.org/standards/cim.

DMTF-MM (2014). "Common Information Model (CIM) Metamodel". In: Document Number: DSP0004 2.7.0, p. 186. URL: https://www.dmtf.org/sites/default/files/standards/documents/DSP0004_3.0.1.pdf.

Docker (2019). *Enterprise Container Platform | Docker*. URL: https://www.docker.com/.

Dorri, Ali, Salil S Kanhere, and Raja Jurdak (2016). "Blockchain in internet of things: challenges and solutions". In: *arXiv preprint arXiv:1608.05187*.

Dorri, Ali, Salil S. Kanhere, and Raja Jurdak (2017a). "Towards an Optimized BlockChain for IoT". In: *Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 173–178. URL: https://ieeexplore.ieee.org/abstract/document/7946872.

Dorri, Ali et al. (2017b). "Blockchain for IoT security and privacy: The case study of a smart home". In: *International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 618–623. DOI: 10.1109/PERCOMW.2017.7917634. URL: https://arxiv.org/ftp/arxiv/papers/1608/1608.05187.pdf.

Douceur, John R (2002). "The sybil attack". In: *International workshop on peer-to-peer systems*. Springer, pp. 251–260. URL: https://link.springer.com/chapter/10.1007/3-540-45748-8_24.

Dumas, Marlon et al. (2013). *Fundamentals of Business Process Management*. Springer Berlin Heidelberg. ISBN: 978-3-642-33142-8 978-3-642-33143-5. DOI: 10.1007/978-3-642-33143-5. URL: http://link.springer.com/10.1007/978-3-642-33143-5.

Dworkin, Morris J. (2015). *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Tech. rep. 202. National Institute of Standards and Technology. DOI: 10.6028/NIST.FIPS.202. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

EBSI, EU (2021). *European Blockchain Services Infrastructure | Shaping Europe's digital future*. https://ec.europa.eu/digital-single-market/en/european-blockchain-services-infrastructure.

Egenhofer, M. J. (1994). "Spatial SQL: a query and presentation language". In: *IEEE Transactions on Knowledge and Data Engineering* 6.1, pp. 86–95. ISSN: 1558-2191. DOI: 10.1109/69.273029.

El Ioini, Nabil and Claus Pahl (2018). "A Review of Distributed Ledger Technologies". In: *Meaningful Internet Systems OTM Conference*. Vol. 11230. Springer, pp. 277–288. ISBN: 978-3-030-02670-7 978-3-030-02671-4. DOI: 10.1007/978-3-030-02671-4_16. URL: http://link.springer.com/10.1007/978-3-030-02671-4_16.

Ellram, Lisa M (1991). "Supply-Chain Management: The Industrial Organisation Perspective". In: *International Journal of Physical Distribution & Logistics Management*.

Ellul, Joshua and Gordon J. Pace (2018). "Runtime Verification of Ethereum Smart Contracts". In: *14th European Dependable Computing Conference (EDCC)*, pp. 158–163. ISBN: 978-1-5386-8060-5. DOI: 10.1109/EDCC.2018.00036. URL: https://ieeexplore.ieee.org/document/8530777/.

Engelenburg, Sélinde van, Marijn Janssen, and Bram Klievink (2019). "Design of a software architecture supporting business-to-government information sharing to improve public safety and security: Combining business rules, Events and blockchain technology". In: *Journal of Intelligent Information Systems* 52.3, pp. 595–618. ISSN: 0925-9902, 1573-7675. DOI: 10.1007/s10844-017-0478-z. URL: http://link.springer.com/10.1007/s10844-017-0478-z.

Environment, Ministry (2020). *Waste collection and transportation permit — Business — Guichet.lu - Administrative Guide // Luxembourg*. URL: https://guichet.public.lu/en/entreprises/urbanisme-environnement/dechets-subst-dangereuses/transport-dechets/transport-collecte-dechets.html.

Environment-Agency (2020). *Environment Agency — Single Window for Logistics // Luxembourg*. URL: https://logistics.public.lu/en/formalities-procedures/agencies/environment-agency.html.

EU Note (n.d.). URL: https://ec.europa.eu/competition/international/multilateral/2010_information_exchange.pdf.

Eurostat (2019). *Eurostat - Data Explorer*. URL: https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=road_go_ta_tott&lang=en.

Fabric-v.2.2, Hyperledger (2021). *Introduction — hyperledger-fabricdocs master documentation*. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatis.html.

Feng Tian (2016). "An agri-food supply chain traceability system for China based on RFID & blockchain technology". In: *13th International Conference on Service Systems and Service Management (ICSSSM)*. Kunming, China: IEEE, pp. 1–6. ISBN: 978-1-5090-2842-9. DOI: 10.1109/ICSSSM.2016.7538424. URL: http://ieeexplore.ieee.org/document/7538424/.

Fernández-Caramés, Tiago M. and Paula Fraga-Lamas (2018). "A Review on the Use of Blockchain for the Internet of Things". In: *IEEE Access* 6, pp. 32979–33001. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2842685.

Feuchtwanger, Hal (2017). "Logistics on the blockchain? It's happening". In: *Retrieved December* 12, p. 2017. URL: https://blog.sweetbridge.com/logistics-on-the-blockchain-consider-this-319859d87089.

Fournier, Gregory and Fabio Petrillo (2020). "Architecting Blockchain Systems: A Systematic Literature Review". In: ICSEW, pp. 664–670. DOI: 10.1145/3387940.3392196. URL: https://doi.org/10.1145/3387940.3392196.

Fowler, Martin (2004). *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.

Frank, William C., Jean-Claude Thill, and Rajan Batta (Feb. 1, 2000). "Spatial decision support system for hazardous material truck routing". In: *Transportation Research Part C: Emerging Technologies* 8.1, pp. 337–359. ISSN: 0968-090X. DOI: 10.1016/S0968-090X(00)00007-3. URL: http://www.sciencedirect.com/science/article/pii/S0968090X00000073.

Frantz, Christopher K. and Mariusz Nowostawski (2016). "From Institutions to Code: Towards Automated Generation of Smart Contracts". In: *1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. Augsburg, Germany: IEEE, pp. 210–215. ISBN: 978-1-5090-3651-6. DOI: 10.1109/FAS-W.2016.53. URL: http://ieeexplore.ieee.org/document/7789470/.

Frooman, Jeff (1999). "Stakeholder influence strategies". In: *Academy of management review* 24.2, pp. 191–205. URL: https://journals.aom.org/doi/abs/10.5465/AMR.1999.1893928.

Gamma, Erich et al. (1995). *Elements of reusable object-oriented software*. Vol. 99. Addison-Wesley Reading, Massachusetts.

Gervais, Arthur et al. (2016). "On the Security and Performance of Proof of Work Blockchains". In: *Conference on Computer and Communications Security*. Vienna Austria: ACM, pp. 3–16. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978341. URL: https://dl.acm.org/doi/10.1145/2976749.2978341.

Glaessgen, Edward and David Stargel (2012). "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles". In: *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Honolulu, Hawaii: American Institute of Aeronautics and Astronautics. ISBN: 978-1-60086-937-2. DOI: 10.2514/6.2012-1818. URL: http://arc.aiaa.org/doi/abs/10.2514/6.2012-1818.

Golatowski, Frank et al. (2019). "Challenges and Research Directions for Blockchains in the Internet of Things". In: *International Conference on Industrial Cyber Physical Systems (ICPS)*, pp. 712–717. DOI: 10.1109/ICPHYS.2019.8780270.

Gong, C. et al. (2010). "The Characteristics of Cloud Computing". In: *39th International Conference on Parallel Processing Workshops*. ISSN: 2332-5690, pp. 275–279. DOI: 10.1109/ICPPW.2010.45.

Goodman, L M (2014). "Tezos — a self-amending crypto-ledger". In: *White paper*, pp. 1–17. URL: https://academy.bit2me.com/wp-content/uploads/2021/04/tezos-whitepaper.pdf.

Grieves, Michael (2014). "Digital twin: manufacturing excellence through virtual factory replication". In: *White paper*, pp. 1–7. URL: https://theengineer.markallengroup.com/production/content/uploads/2014/12/Digital_Twin_White_Paper_Dr_Grieves.pdf.

Grossman, Shelly et al. (2017). "Online detection of effectively callback free objects with applications to smart contracts". In: *Proceedings of the ACM on Programming Languages* 2, pp. 1–28. ISSN: 24751421. DOI: 10.1145/3158136. URL: http://dl.acm.org/citation.cfm?doid=3177123.3158136.

Guardian (2016). *Europe faces €253bn nuclear waste bill | Nuclear waste*. URL: https://www.theguardian.com/environment/2016/apr/04/europe-faces-253bn-nuclear-waste-bill.

Guardian, The (2009). *Shipwreck may hold radioactive waste sunk by mafia off Italian coast | Italy | The Guardian*. https://www.theguardian.com/world/2009/sep/16/shipwreck-waste-mafia-italy.

Gueffaz, Mahdi, Sylvain Rampacek, and Christophe Nicolle (2012). "Temporal Logic To Query Semantic Graphs Using The Model Checking Method". In: *Journal of Software* 7.7, pp. 1462–1472. ISSN: 1796-217X. DOI: 10.4304/jsw.7.7.1462-1472. URL: http://ojs.academypublisher.com/index.php/jsw/article/view/7851.

Guichet.lu (2020). *Waste trading or brokering permit*. URL: https://guichet.public.lu/en/entreprises/urbanisme-environnement/dechets-subst-dangereuses/transport-dechets/negoce-courtage-dechets.html.

GuideADR (2018). *Carriage of Dangerous Goods by Road A Guide For Business*. URL: https://www.hsa.ie/eng/Your_Industry/ADR_-_Carriage_of_Dangerous_Goods_by_Road/New_ADR_Guide_for_Business.pdf.

Hackius, Niels and Moritz Petersen (2017). "Blockchain in logistics and supply chain : trick or treat?" In: Digitalization in Supply Chain Management and Logistics: Smart and Digital Solutions for an Industry 4.0 Environment. Proceedings of the Hamburg International Conference of Logistics (HICL), pp. 3–18. DOI: 10.15480/882.1444. URL: https://tore.tuhh.de/handle/11420/1447.

Halle, Barbara von (2001). *Business Rules Applied: Building Better Systems Using the Business Rules Approach*. 1st. Wiley Publishing. ISBN: 978-0-471-41293-9. URL: https://www.wiley.com/en-al/Business+Rules+Applied%3A+Building+Better+Systems+Using+the+Business+Rules+Approach-p-9780471412939.

Hammi, Mohamed Tahar et al. (2018). "Bubbles of Trust: A decentralized blockchain-based authentication system for IoT". In: *Computers & Security* 78, pp. 126–142. ISSN: 01674048. DOI: 10.1016/j.cose.2018.06.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167404818300890.

Hang, Lei and Do-Hyeun Kim (2019). "Design and Implementation of an Integrated IoT Blockchain Platform for Sensing Data Integrity". In: *Sensors* 19.10, p. 2228. ISSN: 1424-8220. DOI: 10.3390/s19102228. URL: https://www.mdpi.com/1424-8220/19/10/2228.

Hanson, M. David (2000). "The client/server architecture". In: *Server Management*. Auerbach Publications, pp. 17–28.

Harbor, Cara (2018). *IOTA: The Distributed Ledger Technology Designed for the Internet of Things*. URL: https://www.hackster.io/news/iota-the-distributed-ledger-technology-designed-for-the-internet-of-things-f6abaf0e45a4.

Hardin, Garrett (2009). "The tragedy of the commons". In: *Journal of Natural Resources Policy Research* 1.3, pp. 243–253. URL: http://ecoevo.wdfiles.com/local--files/start/Hardin1968.pdf.

Hardjono, Thomas, Alexander Lipton, and Alex Pentland (2018). "Towards a Design Philosophy for Interoperable Blockchain Systems". In: *arXiv:1805.05934*. URL: http://arxiv.org/abs/1805.05934.

Health, Ministry of (2020). *Radiation Protection Department — Single Window for Logistics // Luxembourg*. URL: https://logistics.public.lu/en/formalities-procedures/agencies/radiation-protection-division.html.

Hearn, Mike and Richard Gendal Brown (2019). "Corda: A distributed ledger". In: White Paper, pp. 1–73. URL: https://www.corda.net/content/corda-technical-whitepaper.pdf.

Heutger, Matthias and Markus Kückelhaus (2018). *Perspectives on the upcoming impact of blockchain technology and use cases for the logistics industry*. https://www.dhl.com/content/dam/dhl/global/core/documents/pdf/glo-core-blockchain-trend-report.pdf.

Hildenbrandt, Everett et al. (2018). "KEVM: A Complete Formal Semantics of the Ethereum Virtual Machine". In: *31st Computer Security Foundations Symposium (CSF)*. Oxford: IEEE, pp. 204–217. ISBN: 978-1-5386-6680-7. DOI: 10.1109/CSF.2018.00022. URL: https://ieeexplore.ieee.org/document/8429306/.

Hölbl, Marko et al. (2018). "A Systematic Review of the Use of Blockchain in Healthcare". In: *Symmetry* 10.10, p. 470. ISSN: 2073-8994. DOI: 10.3390/sym10100470. URL: http://www.mdpi.com/2073-8994/10/10/470.

Hochhalter, Jacob et al. (2014). "Coupling Damage-Sensing Particles to the Digitial Twin Concept". In: *NASA/TM–2014-218257*. URL: https://nari.arc.nasa.gov/sites/default/files/Hochhalter_NASA-TM-2014-218257_0.pdf.

Hofweber, Thomas (2020). "Logic and Ontology". In: *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University. URL: https://plato.stanford.edu/archives/sum2020/entries/logic-ontology/.

Holzmann, G. J. (1997). "The model checker SPIN". In: *IEEE Transactions on Software Engineering* 23.5, pp. 279–295. ISSN: 1939-3520. DOI: 10.1109/32.588521.

Horridge, Matthew (2009). "A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.2". In: *The University Of Manchester*, p. 109.

Horrocks, I. et al. (2007). "OWL: a Description-Logic-Based Ontology Language for the Semantic Web". In: *The Description Logic Handbook*. 2nd ed. Cambridge: Cambridge University Press, pp. 458–486. ISBN: 978-0-511-71178-7. DOI: 10.1017/CBO9780511711787.016.

Hoy, Matthew B. (2012). "Cloud Computing Basics for Librarians". In: *Medical Reference Services Quarterly* 31.1, pp. 84–91. ISSN: 0276-3869. DOI: 10.1080/02763869.2012.641853. URL: https://doi.org/10.1080/02763869.2012.641853.

Huang, Hui, Xiaofeng Chen, and Jianfeng Wang (2020). "Blockchain-based multiple groups data sharing with anonymity and traceability". In: *Science China Information Sciences* 63.3,

p. 130101. ISSN: 1674-733X, 1869-1919. DOI: 10.1007/s11432-018-9781-0. URL: http://link.springer.com/10.1007/s11432-018-9781-0.

Huh, Seyoung, Sangrae Cho, and Soohyung Kim (2017). "Managing IoT devices using blockchain platform". In: *19th International Conference on Advanced Communication Technology (ICACT)*, pp. 464–467. ISBN: 978-89-968650-9-4. DOI: 10.23919/ICACT.2017.7890132. URL: http://ieeexplore.ieee.org/document/7890132/.

Hutchinson, John, Mark Rouncefield, and Jon Whittle (2011). "Model-driven engineering practices in industry". In: *33rd international conference on Software engineering - ICSE*, p. 633. ISBN: 978-1-4503-0445-0. DOI: 10.1145/1985793.1985882. URL: http://portal.acm.org/citation.cfm?doid=1985793.1985882.

Hyperledger.org (2016). *Hyperledger – Open Source Blockchain Technologies*. URL: https://www.hyperledger.org/.

IATA (2017). *Dangerous Goods Regulations for AIR Transport*. URL: https://www.iata.org/en/publications/dgr/.

IBM and Maersk (2017). URL: www-03.ibm.com/press/us/en/pressrelease/51712.wss.

Imeri (2019). *Gr4pha/hyperledger-dynamic-smart-contract*. URL: https://github.com/Gr4pha/hyperledger-dynamic-smart-contract.

Imeri, A, N Agoulmine, and D Khadraoui (2020a). "Smart Contract modeling and verification techniques: A survey". In: *8th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2020)*. Cancun, Mexico, pp. 1–8. URL: https://hal.archives-ouvertes.fr/hal-02495158.

Imeri, Adnan, Nazim Agoulmine, and Djamel Khadraoui (2019a). "A secure and smart environment for the transportation of dangerous goods by using Blockchain and IoT devices". In: *7th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2019)*. Praia, Cape Verde, pp. 1–8. URL: https://hal.archives-ouvertes.fr/hal-02439984.

Imeri, Adnan, Abdelaziz Khadraoui, and Djamel Khadraoui (2017). "A Conceptual and Technical Approach for Transportation of Dangerous Goods in Compliance with Regulatory Framework". In: *Journal of Software* 12.9, p. 14. ISSN: 1796-217X. DOI: 10.1016/j.ssci.2016.09.008.

Imeri, Adnan and Djamel Khadraoui (2018). "The Security and Traceability of Shared Information in the Process of Transportation of Dangerous Goods". In: 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS). Paris: IEEE, pp. 1–5. ISBN: 978-1-5386-3662-6. DOI: 10.1109/NTMS.2018.8328751. URL: http://ieeexplore.ieee.org/document/8328751/.

Imeri, Adnan, Djamel Khadraoui, and Nazim Agoulmine (2019b). "Blockchain Technology for the Improvement of SCM and Logistics Services: A Survey". In: *Industrial Engineering in the Big Data Era*. Springer, pp. 349–361. ISBN: 978-3-030-03317-0. URL: https://link.springer.com/chapter/10.1007%2F978-3-030-03317-0_29.

Imeri, Adnan and Jonathan Lamont (2019). *Proof of Concept (PoC): Blockchain and IoT*. URL: https://github.com/Gr4pha/hyperledger-blockchain-and-iot.

Imeri, Adnan et al. (2018). "Solving the trust issues in the process of transportation of dangerous goods by using blockchain technology". In: 11th International Conference on Security of Information and Networks - SIN '18. Cardiff, United Kingdom: ACM Press, pp. 1–2. ISBN: 978-1-4503-6608-3. DOI: 10.1145/3264437.3264470. URL: http://dl.acm.org/citation.cfm?doid=3264437.3264470.

Imeri, Adnan et al. (2019c). "Blockchain: Analysis of the New Technological Components as Opportunity to Solve the Trust Issues in Supply Chain Management". In: *Intelligent Computing*. Vol. 998. Series Title: Advances in Intelligent Systems and Computing. Springer, pp. 474–493. ISBN: 978-3-030-22867-5 978-3-030-22868-2. DOI: 10.1007/978-3-030-22868-2_36. URL: http://link.springer.com/10.1007/978-3-030-22868-2_36.

Imeri, Adnan et al. (2019d). *Method For Improving Blockchain Applications*. URL: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2020165382&tab=PCTBIBLIO.

Imeri, Adnan et al. (2019e). "Model of dynamic smart contract for permissioned blockchains". In: Practice of Enterprise Modelling Conference Forum (PoEM 2019 Forum). Luxembroug, pp. 1–16. URL: http://ceur-ws.org/Vol-2586/paper1.pdf.

Imeri, Adnan A., Nazim Agoulmine, and Djamel Khadraoui (2020b). "Blockchain and IoT integrated approach for a trusted and secured process to manage the transportation of dangerous goods". In: *Computing and System Journal* 10.1, pp. 1–17. ISSN: 2237-2903. URL: https://revistas.unifacs.br/index.php/rsc/article/view/6444.

Insights, Chain Business (2017). *Blockchain in Supply Chain Edging Toward Higher Visibility Benchmark Survey*. URL: https://www.chainbusinessinsights.com/blockchain-in-supply-chain-edging-toward-higher-visibility-survey.html.

IoT-Salesforce (n.d.). URL: https://cdn.ihs.com/www/pdf/enabling-IOT.pdf.

Iota (2017). *IOTA-Next Generation Block chain | International Journal of Engineering and Computer Science*. http://ijecs.in/index.php/ijecs/article/view/4007.

IOTABlog (2018). *The Tangle*. URL: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf.

IoTBlog (2020). *IoT & Embedded Technology Blog*. URL: https://www.vdcresearch.com/News-events/iot-blog/.

Jennings, Nicholas R. and Michael J. Wooldridge (1998). *Agent Technology*. Springer Berlin Heidelberg. ISBN: 978-3-642-08344-0 978-3-662-03678-5. DOI: 10.1007/978-3-662-03678-5. URL: http://link.springer.com/10.1007/978-3-662-03678-5.

Jeppsson, André and Oskar Olsson (2017). "Blockchains as a solution for traceability and transparency". In: *Lund University*. URL: http://lup.lub.lu.se/student-papers/record/8919957.

Jiang, Yiming et al. (2019). "A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management". In: *Sensors* 19.9. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 2042. DOI: 10.3390/s19092042. URL: https://www.mdpi.com/1424-8220/19/9/2042.

Johnson, Don, Alfred Menezes, and Scott Vanstone (2001). "The elliptic curve digital signature algorithm (ECDSA)". In: *International journal of information security* 1.1, pp. 36–63.

Juma, Hussam (2020). "Cross-Border Trade Through Blockchain". In: *International Triple Helix Summit*. Lecture Notes in Civil Engineering. Springer, pp. 165–179. ISBN: 978-3-030-23898-8. DOI: 10.1007/978-3-030-23898-8_13.

Justice, Ministry of (2020). *Ministry of Justice — Single Window for Logistics // Luxembourg*. URL: https://logistics.public.lu/en/formalities-procedures/agencies/ministry-justice.html.

Kahani, Nafiseh et al. (2019). "Survey and classification of model transformation tools". In: *Software & Systems Modeling* 18.4, pp. 2361–2397. ISSN: 1619-1366, 1619-1374. DOI:

10.1007/s10270-018-0665-6. URL: http://link.springer.com/10.1007/s10270-018-0665-6.

Kalra, Sukrit et al. (2018). "ZEUS: Analyzing Safety of Smart Contracts". In: *Network and Distributed System Security Symposium*. San Diego, CA: Internet Society. ISBN: 978-1-891562-49-5. DOI: 10.14722/ndss.2018.23082. URL: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_09-1_Kalra_paper.pdf.

Kanj, Hassan (2016). "Contribution to risk analysis related to the transport of hazardous materials by agent-based simulation". Theses. Université Grenoble Alpes. URL: https://tel.archives-ouvertes.fr/tel-01447765.

Kanter, Joel P. (1998). *Understanding thin-client/server computing*. Strategic technology series. Redmond, Wash: Microsoft Press. ISBN: 978-1-57231-744-4.

Kardoš, Martin and Matilda Drozdová (2010). "Analytical Method of CIM to PIM Transformation in Model Driven Architecture (MDA)". In: *Journal of information and organizational sciences* 34.1. URL: https://core.ac.uk/reader/14425480.

Keller, Er et al. (2001). *RunTime Application Management*. IFIP/EEE International Workshop on Distributed Systems: Operations & Management. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.353.4355&rep=rep1&type=pdf.

Khalil, Amal and Juergen Dingel (2018). "Optimizing the Symbolic Execution of Evolving Rhapsody Statecharts". In: *Advances in Computers*. Vol. 108. Elsevier, pp. 145–281. ISBN: 978-0-12-815119-8. DOI: 10.1016/bs.adcom.2017.09.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0065245817300487.

Khan, Minhaj Ahmad and Khaled Salah (2018). "IoT security: Review, blockchain solutions, and open challenges". In: *Future Generation Computer Systems* 82, pp. 395–411. ISSN: 0167-739X. DOI: 10.1016/j.future.2017.11.022. URL: http://www.sciencedirect.com/science/article/pii/S0167739X17315765.

Khan, Rafiullah et al. (2012). "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges". In: *10th International Conference on Frontiers of Information Technology*. Islamabad, Pakistan: IEEE, pp. 257–260. ISBN: 978-0-7695-4927-9 978-1-4673-4946-8. DOI: 10.1109/FIT.2012.53. URL: http://ieeexplore.ieee.org/document/6424332/.

Kühne, Thomas (2006). "Matters of (Meta-) Modeling". In: *Software & Systems Modeling*, pp. 369–385. ISSN: 1619-1366, 1619-1374. DOI: 10.1007/s10270-006-0017-9. URL: http://link.springer.com/10.1007/s10270-006-0017-9.

Kim, Henry M. and Marek Laskowski (2018). "Toward an ontology-driven blockchain design for supply-chain provenance". In: *Intelligent Systems in Accounting, Finance and Management* 25.1, pp. 18–27. ISSN: 1099-1174. DOI: https://doi.org/10.1002/isaf.1424. URL: http://onlinelibrary.wiley.com/doi/abs/10.1002/isaf.1424.

Kleppe, Anneke G et al. (2003). *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley Professional.

Košťál, Kristián et al. (2019). "Management and Monitoring of IoT Devices Using Blockchain". In: *Sensors* 19.4, p. 856. DOI: 10.3390/s19040856. URL: https://www.mdpi.com/1424-8220/19/4/856.

Kolluri, Aashish et al. (2018). "Exploiting The Laws of Order in Smart Contracts". In: *28th ACM SIGSOFT international symposium on software testing and analysis*, pp. 363–373. URL: https://dl.acm.org/doi/10.1145/3293882.3330560.

Kononenko, Oleksii et al. (2014). "Mining modern repositories with elasticsearch". In: *11th Working Conference on Mining Software Repositories*, pp. 328–331. ISBN: 978-1-4503-2863-0. DOI: 10.1145/2597073.2597091. URL: http://dl.acm.org/citation.cfm?doid=2597073.2597091.

Korpela, Kari, Jukka Hallikas, and Tomi Dahlberg (2017). "Digital Supply Chain Transformation toward Blockchain Integration". In: *50th Hawaii international conference on system sciences*. ISBN: 978-0-9981331-0-2. DOI: 10.24251/HICSS.2017.506. URL: http://scholarspace.manoa.hawaii.edu/handle/10125/41666.

Kraft, Daniel (2016). "Difficulty control for blockchain-based consensus systems". In: *Peer-to-Peer Networking and Applications* 9.2, pp. 397–413. ISSN: 1936-6442, 1936-6450. DOI: 10.1007/s12083-015-0347-x. URL: http://link.springer.com/10.1007/s12083-015-0347-x.

Kshetri, Nir (2018). "1 Blockchain's roles in meeting key supply chain management objectives". In: *International Journal of Information Management* 39, pp. 80–89. ISSN: 02684012. DOI: 10.1016/j.ijinfomgt.2017.12.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S0268401217305248.

Kuo, Tsung-Ting, Hugo Zavaleta Rojas, and Lucila Ohno-Machado (2019). "Comparison of blockchain platforms: a systematic review and healthcare examples". In: *Journal of the American Medical Informatics Association* 26.5, pp. 462–478. ISSN: 1527-974X. DOI: 10.1093/jamia/ocy185. URL: https://academic.oup.com/jamia/article/26/5/462/5419321.

Laarabi, Mohamed Haitam et al. (2014). "A scalable communication middleware for real-time data collection of dangerous goods vehicle activities". In: *Transportation Research Part C: Emerging Technologies* 48, pp. 404–417. ISSN: 0968090X. DOI: 10.1016/j.trc.2014.09.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X14002538.

Lambert, Douglas M. (2008). *Supply chain management: processes, partnerships, performance*. Supply Chain Management Inst.

Lamport, Leslie, Robert Shostak, and Marshall Pease (1982). "The Byzantine Generals Problem". In: *ACM Transactions on Programming Languages and Systems* 4.3, p. 20. URL: http://people.cs.uchicago.edu/~shanlu/teaching/33100_wi15/papers/byz.pdf.

Larsen, Kim G., Paul Pettersson, and Wang Yi (1997). "Uppaal in a nutshell". In: *International Journal on Software Tools for Technology Transfer* 1.1-2, pp. 134–152. ISSN: 1433-2779. DOI: 10.1007/s100090050010. URL: http://link.springer.com/10.1007/s100090050010.

Leonardos, Stefanos, Daniël Reijsbergen, and Georgios Piliouras (2020). "PREStO: A Systematic Framework for Blockchain Consensus Protocols". In: *IEEE Transactions on Engineering Management* 67.4, pp. 1028–1044. ISSN: 1558-0040. DOI: 10.1109/TEM.2020.2981286. URL: https://ieeexplore.ieee.org/abstract/document/9082016.

Leung, Yee, Rongrong Li, and Nannan Ji (2017). "Application of extended Dempster–Shafer theory of evidence in accident probability estimation for dangerous goods transportation". In: *Journal of Geographical Systems* 19.3, pp. 249–271. ISSN: 1435-5930, 1435-5949. DOI: 10.1007/s10109-017-0253-2. URL: http://link.springer.com/10.1007/s10109-017-0253-2.

Lewandowski, Scott M. (1998). "Frameworks for component-based client/server computing". In: *ACM Computing Surveys* 30.1, pp. 3–27. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/274440.274441. URL: https://dl.acm.org/doi/10.1145/274440.274441.

Lewis, Anthon (2016). *A gentle introduction to immutability of blockchains – Bits on Blocks*. https://bitsonblocks.net/2016/02/29/a-gentle-introduction-to-immutability-of-blockchains/. (Accessed on 01/09/2020).

Liao, Chun-Feng et al. (2017). "On design issues and architectural styles for blockchain-driven IoT services". In: *International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, pp. 351–352. DOI: 10.1109/ICCE-China.2017.7991140.

Lim, Written Johanne and Relly Noman (2018). *Introduction – OpenPort*. URL: https://openport.com/whitepaper/executive-summary/.

Lin, Fu-ren, Yu-wei Sung, and Yi-pong Lo (2005). "Effects of Trust Mechanisms on Supply-Chain Performance: A Multi-Agent Simulation Study". In: *International Journal of Electronic Commerce* 9.4, pp. 9–112. ISSN: 1086-4415. DOI: 10.1080/10864415.2003.11044342. URL: https://doi.org/10.1080/10864415.2003.11044342.

Lin, Iuon-Chang and Tzu-Chun Liao (2017). "A Survey of Blockchain Security Issues and Challenges". In: *International Journal of Network Security* 19.5, pp. 653–659. ISSN: 1816-353X. DOI: 10.6633/IJNS.201709.19(5).01.

Lin, Jun, Zhiqi Shen, and Chunyan Miao (2017). "Using Blockchain Technology to Build Trust in Sharing LoRaWAN IoT". In: *2nd International Conference on Crowd Science and Engineering*. ICCSE'17. New York, USA, pp. 38–43. ISBN: 978-1-4503-5375-5. DOI: 10.1145/3126973.3126980. URL: https://doi.org/10.1145/3126973.3126980.

Lin, Jun et al. (2018). "Blockchain and IoT based Food Traceability for Smart Agriculture". In: *3rd International Conference on Crowd Science and Engineering - ICCSE*. Singapore, Singapore: ACM Press, pp. 1–6. ISBN: 978-1-4503-6587-1. DOI: 10.1145/3265689.3265692. URL: http://dl.acm.org/citation.cfm?doid=3265689.3265692.

Liu, B. et al. (2017). "Blockchain Based Data Integrity Service Framework for IoT Data". In: *International Conference on Web Services (ICWS)*, pp. 468–475. DOI: 10.1109/ICWS.2017.54.

Liu, Bowen, Siwei Sun, and Pawel Szalachowski (2020). "SMACS: Smart Contract Access Control Service". In: *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. Valencia, Spain: IEEE, pp. 221–232. ISBN: 978-1-72815-809-9. DOI: 10.1109/DSN48063.2020.00039. URL: https://ieeexplore.ieee.org/document/9153398/.

Liu, Yue et al. (2018). "Applying Design Patterns in Smart Contracts". In: *International Conference on Blockchain – ICBC*. Lecture Notes in Computer Science. Springer, pp. 92–106. ISBN: 978-3-319-94477-7 978-3-319-94478-4. DOI: 10.1007/978-3-319-94478-4_7. URL: http://link.springer.com/10.1007/978-3-319-94478-4_7.

López-Pintado, Orlenys et al. (2018). "Dynamic Role Binding in Blockchain-Based Collaborative Business Processes". In: *International Conference on Advanced Information Systems Engineering*. DOI: 10.1007/978-3-030-21290-2_25. URL: https://link-springer-com.proxy.bnl.lu/chapter/10.1007/978-3-030-21290-2_25.

López-Pintado, Orlenys et al. (2019). "CATERPILLAR: A Business Process Execution Engine on the Ethereum Blockchain". In: *Software: Practice and Experience*. DOI: https://doi.org/10.1002/spe.2702. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2702.

Lu, Qinghua and Xiwei Xu (2017). "Adaptable Blockchain-Based Systems: A Case Study for Product Traceability". In: *IEEE Software* 34.6, pp. 21–27. ISSN: 0740-7459. DOI: 10.1109/MS.2017.4121227. URL: http://ieeexplore.ieee.org/document/8106871/.

Lu, Qinghua et al. (2020). "Integrated Model-Driven Engineering of Blockchain Applications for Business Processes and Asset Management". In: *Software: Practice and Experience*. DOI: https://doi.org/10.1002/spe.2931. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2931.

Luu, Loi et al. (2016). "Making Smart Contracts Smarter". In: *Conference on Computer and Communications Security - CCS*. Vienna, Austria: ACM Press, pp. 254–269. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978309. URL: http://dl.acm.org/citation.cfm?doid=2976749.2978309.

Macdonald, M, L Liu-Thorrold, and R Julien (2017). "The Blockchain: A Comparison of Platforms and Their Uses Beyond Bitcoin". In: *COMS4507-Adv. Computer and Network Security*. DOI: 10.13140/RG.2.2.23274.52164. URL: https://www.ijsdr.org/papers/IJSDR1811024.pdf.

Magnusson, Camilla Nyquist (2015). "Transportation of dangerous goods: A multiple stakeholder analysis for improved efficiency and safety through information sharing". In: *NOFOMA*, p. 17. URL: https://www.lunduniversity.lu.se/lup/publication/72a5bace-2fcf-4b42-b6ee-bbaea06b5700.

Mahindra and IBM (2017). URL: https://www.ibm.com/blogs/blockchain/2017/03/disrupting-supply-chain-financing-mahindra/.

Majewska, Marta, Bartosz Kryza, and Jacek Kitowski (2007). "Translation of Common Information Model to Web Ontology Language". In: *Computational Science – ICCS*. Vol. 4487. Lecture Notes in Computer Science. Springer, pp. 414–417. ISBN: 978-3-540-72583-1 978-3-540-72584-8. DOI: 10.1007/978-3-540-72584-8_53. URL: http://link.springer.com/10.1007/978-3-540-72584-8_53.

Malik, S. et al. (2019). "TrustChain: Trust Management in Blockchain and IoT Supported Supply Chains". In: *International Conference on Blockchain (Blockchain)*, pp. 184–193. DOI: 10.1109/Blockchain.2019.00032.

Mandolla, Claudio et al. (2019). "Building a digital twin for additive manufacturing through the exploitation of blockchain: A case analysis of the aircraft industry". In: *Computers in Industry* 109, pp. 134–152. ISSN: 0166-3615. DOI: 10.1016/j.compind.2019.04.011. URL: http://www.sciencedirect.com/science/article/pii/S0166361518308741.

Martin, James and James J. Odell (1994). *Object-Oriented Methods*. USA: Prentice Hall PTR. ISBN: 0136308562.

Massey, Rob et al. (2019). *Blockchain. Deloitte Insights*. URL: https://www2.deloitte.com/content/dam/Deloitte/se/Documents/risk/DI_2019-global-blockchain-survey.pdf.

Mavridou, Anastasia (2019). *Correct-by-Design Smart Contracts: FSolidM / VeriSolid Framework*. URL: https://github.com/anmavrid/smart-contracts.

Mavridou, Anastasia and Aron Laszka (2018). "Designing Secure Ethereum Smart Contracts: A Finite State Machine Based Approach". In: *International Conference on Financial Cryptography and Data Security*. DOI: 10.1007/978-3-662-58387-6_28. URL: https://link.springer.com/chapter/10.1007/978-3-662-58387-6_28.

McFarland, Daniel and Darren B. Nicholson (2007). "Client/Server Computing Basics". In: *Handbook of Computer Networks: Distributed Networks, Network Planning, Control, Management, and New Trends and Applications* 3. Publisher: Wiley Online Library, pp. 1–15.

Medaglia, Carlo Maria and Alexandru Serbanati (2010). "An Overview of Privacy and Security Issues in the Internet of Things". In: *The Internet of Things*. New York: Springer, pp. 389–395. ISBN: 978-1-4419-1674-7. DOI: 10.1007/978-1-4419-1674-7_38.

Media, Hyperledger (2019). *Hyperledger-fabricdocs Documentation*. URL: https://buildmedia.readthedocs.org/media/pdf/hyperledger-fabric/latest/hyperledger-fabric.pdf..

Mell, Peter and Timothy Grance (2011). "The NIST Definition of Cloud Computing". In: *National Institute of Standards and Technology (NIST). Computer Security Resource Center*. DOI: SP800-145.

Mendling, Jan et al. (2018). "Blockchains for Business Process Management - Challenges and Opportunities". In: *ACM Transactions on Management Information Systems (TMIS)* 9 (1), pp. 1–16. DOI: https://doi.org/10.1145/3183367. URL: https://dl.acm.org/doi/abs/10.1145/3183367.

Mentzer, John T et al. (2001). "Defining Supply Chain Management". In: *Journal Of Business Logistics* 22.2, pp. 1–25. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2158-1592.2001.tb00001.x.

Mik, Eliza (2017). "Smart contracts: terminology, technical limitations and real world complexity". In: *Law, Innovation and Technology* 9.2, pp. 269–300. ISSN: 1757-9961, 1757-997X. DOI: 10.1080/17579961.2017.1378468. URL: https://www.tandfonline.com/doi/full/10.1080/17579961.2017.1378468.

Mikk, E. et al. (1998). "Implementing statecharts in PROMELA/SPIN". In: *2nd IEEE Workshop on Industrial Strength Formal Specification Techniques*, pp. 90–101. DOI: 10.1109/WIFT.1998.766303.

Mingxiao, Du et al. (Oct. 2017). "A review on consensus algorithm of blockchain". In: *International Conference on Systems, Man, and Cybernetics (SMC)*. 2017 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2567–2572. ISBN: 978-1-5386-1645-1. DOI: 10.1109/SMC.2017.8123011. URL: http://ieeexplore.ieee.org/document/8123011/.

Müller, M. et al. (2019). "HIDALS: A Hybrid IoT-based Decentralized Application for Logistics and Supply Chain Management". In: *10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. ISSN: 2644-3163, pp. 0802–0808. DOI: 10.1109/IEMCON.2019.8936305.

Müller, Marcel, Nadine Ostern, and Michael Rosemann (2020). "Silver Bullet for All Trust Issues? Blockchain-Based Trust Patterns for Collaborative Business Processes". In: *Business Process Management: Blockchain and Robotic Process Automation Forum*. Vol. 393. Springer, pp. 3–18. ISBN: 978-3-030-58778-9 978-3-030-58779-6. DOI: 10.1007/978-3-030-58779-6_1. URL: http://link.springer.com/10.1007/978-3-030-58779-6_1.

Mofarrahi, Maryam, Masoud Rahiminezhad Galankashi, and Ali Shahidinejad (2014). "The State Of The Art Information Sharing Technologies For Supply Chain Management". In: *Journal of Theoretical and Applied Information Technology* 59, p. 7.

Mohan, Gowtham and Timotheus Kampik (n.d.). *BPMNSol*. URL: https://github.com/signavio/BPMN-Sol.

Molero, Gemma Dolores et al. (2017). "Total safety by design: Increased safety and operability of supply chain of inland terminals for containers with dangerous goods". In: *Safety Science* 100, pp. 168–182. ISSN: 09257535. DOI: 10.1016/j.ssci.2016.10.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925753516303289.

Morgan, Tony (2002). *Business rules and information systems: aligning IT with business goals*. Addison-Wesley Professional.

Márquez, Adolfo Crespo (2010). *Dynamic Modelling for Supply Chain Management: Dealing with Front-end, Back-end and Integration Issues*. London: Springer. ISBN: 978-1-84882-680-9. DOI: 10.1007/978-1-84882-681-6. URL: https://www.springer.com/gp/book/9781848826809.

Murray, Yvonne and David A. Anisi (2019). "Survey of Formal Verification Methods for Smart Contracts on Blockchain". In: *10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. Canary Islands, Spain: IEEE, pp. 1–6. ISBN: 978-1-72811-542-9. DOI: 10.1109/NTMS.2019.8763832. URL: https://ieeexplore.ieee.org/document/8763832/.

Mutunda, Fortunat Lufunda (2017). "Timer Service for an Ethereum BPMN Engine". In: *Master Thesis at University Of Tartu*, p. 51. URL: https://core.ac.uk/download/pdf/237084691.pdf.

Nakamoto, Satoshi (2009). "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *White Paper*, p. 9. URL: https://bitcoin.org/bitcoin.pdf.

Nakasumi, Mitsuaki (2017). "Information Sharing for Supply Chain Management Based on Block Chain Technology". In: *19th Conference on Business Informatics (CBI)*, pp. 140–149. ISBN: 978-1-5386-3035-8. DOI: 10.1109/CBI.2017.56. URL: http://ieeexplore.ieee.org/document/8010716/.

Nassar, Heba A., Aysh Alhroob, and Ayad T. Imam (2017). "An Algorithmic Approach for Sketching Sequence Diagram". In: *International Conference on Advances in Image Processing - ICAIP*, pp. 156–160. ISBN: 978-1-4503-5295-6. DOI: 10.1145/3133264.3133304. URL: http://dl.acm.org/citation.cfm?doid=3133264.3133304.

Natoli, Christopher et al. (2019). "Deconstructing Blockchains: A Comprehensive Survey on Consensus, Membership and Structure". In: *arXiv:1908.08316 [cs]*. arXiv: 1908.08316. URL: http://arxiv.org/abs/1908.08316.

Nehai, Zeinab, Pierre-Yves Piriou, and Frederic Daumas (2018). "Model-Checking of Smart Contracts". In: *International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. Halifax, Canada: IEEE, pp. 980–987. ISBN: 978-1-5386-7975-3. DOI: 10.1109/Cybermatics_2018.2018.00185. URL: https://ieeexplore.ieee.org/document/8726806/.

Nguyen, Giang Truong and Kyungbaek Kim (2018). "A survey about consensus algorithms used in Blockchain". In: *Journal of Information Processing Systems* 14.1, pp. 101–128. ISSN: 2092805X. DOI: 10.3745/JIPS.01.0024.

Nijsse, Jeff and Alan Litchfield (2020). "A Taxonomy of Blockchain Consensus Methods". en. In: *Cryptography* 4.4, p. 32. ISSN: 2410-387X. DOI: 10.3390/cryptography4040032. URL: https://www.mdpi.com/2410-387X/4/4/32.

Nizamuddin, N. et al. (2019). "Decentralized document version control using ethereum blockchain and IPFS". In: *Computers & Electrical Engineering* 76, pp. 183–197. ISSN: 00457906. DOI: 10.1016/j.compeleceng.2019.03.014. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045790618333093.

Notheisen, Benedikt, Jacob Benjamin Cholewa, and Arun Prasad Shanmugam (2017). "Trading Real-World Assets on Blockchain: An Application of Trust-Free Transaction Systems in the Market for Lemons". In: *Business & Information Systems Engineering* 59.6, pp. 425–440. ISSN: 2363-7005, 1867-0202. DOI: 10.1007/s12599-017-0499-8. URL: http://link.springer.com/10.1007/s12599-017-0499-8.

Notion, United (2015). *Transport of Dangerous Goods*. URL: http://www.unece.org/fileadmin/DAM/trans/danger/publi/unrec/rev19/Rev19e_Vol_I.pdf.

NuclearWaste (2019). *World_Nuclear_Waste_Report_2019_Focus_Europe_0.pdf*. https://www.boell.de/sites/default/files/2019-11/World_Nuclear_Waste_Report_2019_Focus_Europe_0.pdf.

Ocalir-Akunal, Ebru Vesile (2016). "Decision Support Systems in Transport Planning". In: *Procedia Engineering* 161, pp. 1119–1126. ISSN: 18777058. DOI: 10.1016/j.proeng.2016.08.518. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877705816327266.

Odun-Ayo, I. et al. (2018). "Cloud Computing Architecture: A Critical Analysis". In: *18th International Conference on Computational Science and Applications (ICCSA)*, pp. 1–7. DOI: 10.1109/ICCSA.2018.8439638.

OECD (2008). *Decision of the Council on the OECD Legal Instruments Control of Transboundary Movements of Wastes Destined for Recovery Operations*. URL: https://legalinstruments.oecd.org/public/doc/221/221.en.pdf.

OMG (2014). *MDA Specifications | Object Management Group*. URL: https://www.omg.org/mda/specs.htm.

OMG, MOF (2017). *Meta-Modeling and the MOF*. URL: https://www.omg.org/ocup-2/documents/Meta-ModelingAndtheMOF.pdf.

OMG-BPMN (2013). *Business Process Model and Notation (BPMN), Version 2.0*. URL: https://www.omg.org/spec/BPMN/2.0.2/PDF.

OMG-MARTE (2019). *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems, v1.2*. URL: https://www.omg.org/spec/MARTE/1.2/PDF.

OMG-MDA (2014). *Model Driven Architecture (MDA) MDA Guide rev. 2.0*. URL: https://www.omg.org/cgi-bin/doc?ormsc/14-06-01.

OMG-MOF (2019). *Meta Object Facility (MOF) Core Specification*. URL: https://www.omg.org/spec/MOF/2.5.1/PDF.

OMG-QVT (2016). *OMG-QVT*. URL: https://www.omg.org/spec/QVT/1.3/PDF.

OMG-Superstructure (2014). *UML 2.0 Infrastructure Final Adopted Specifcation, Object Management Group (OMG)*. URL: https://www.omg.org/spec/UML/2.3/Superstructure/PDF.

OMG-UML (2010). *About the Unified Modeling Language Specification Version 2.4.1*. URL: https://www.omg.org/spec/UML/2.4.1/About-UML/.

OMGGroup (2020). *About OMG | Object Management Group*. URL: https://www.omg.org/about/index.htm.

Ongaro, Diego and John Ousterhout (2014). "In search of an understandable consensus algorithm". In: USENIX Annual Technical Conference, pp. 305–320. URL: https://web.stanford.edu/~ouster/cgi-bin/papers/raft-atc14.

Çorak, B. H. et al. (2018). "Comparative Analysis of IoT Communication Protocols". In: *International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6. DOI: 10.1109/ISNCC.2018.8530963.

OTIF (2019). *International Martime Organization (IMDG)*. URL: https://www.labelmaster.com/shop/regulatory-publications/maritime-transport/.

Paliwal, Vineet, Shalini Chandra, and Suneel Sharma (2020). "Blockchain Technology for Sustainable Supply Chain Management: A Systematic Literature Review and a Classification Framework". In: *Sustainability* 12.18, p. 7638. ISSN: 2071-1050. DOI: 10.3390/su12187638. URL: https://www.mdpi.com/2071-1050/12/18/7638.

Panarello, Alfonso et al. (Aug. 6, 2018). "Blockchain and IoT Integration: A Systematic Survey". In: *Sensors* 18.8, p. 2575. ISSN: 1424-8220. DOI: 10.3390/s18082575. URL: http://www.mdpi.com/1424-8220/18/8/2575.

Patcha, Animesh and Jung-Min Park (2007). "An overview of anomaly detection techniques: Existing solutions and latest technological trends". In: *Computer Networks* 51.12, pp. 3448–3470. ISSN: 13891286. DOI: 10.1016/j.comnet.2007.02.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S138912860700062X.

Patidar, S., D. Rane, and P. Jain (2012). "A Survey Paper on Cloud Computing". In: *Second International Conference on Advanced Computing Communication Technologies*. ISSN: 2327-0659, pp. 394–398. DOI: 10.1109/ACCT.2012.15.

Paulavičius, Remigijus et al. (2019). "A Decade of Blockchain: Review of the Current Status, Challenges, and Future Directions". In: *Informatica, Vilnius University Institute of Mathematics and Informatics* 30.4, p. 21. DOI: https://doi.org/10.15388/Informatica.2019.227. URL: https://informatica.vu.lt/journal/INFORMATICA/article/1141/info.

Pawczuk, Linda, Rob Massey, and Jonathan Holdowsky (2018). *Blockchain | Deloitte Insights*. URL: https://www2.deloitte.com/bg/en/pages/technology/articles/global-blokchain-survey-2018.html.

Perboli, G., S. Musso, and M. Rosano (2018). "Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases". In: *IEEE Access* 6, pp. 62018–62028. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2875782.

Petersen, Oskar and Fredrik Jansson (2017). "Blockchain Technology in Supply Chain Traceability Systems". In: *Lund University*. URL: http://lup.lub.lu.se/student-papers/record/8918347.

Pillai, Babu, Kamanashis Biswas, and Vallipuram Muthukkumarasamy (2020). "Cross-chain interoperability among blockchain-based systems using transactions". In: *The Knowledge Engineering Review* 35. ISSN: 0269-8889, 1469-8005. DOI: 10.1017/S0269888920000314.

Planas, E. et al. (2008). "Results of the MITRA project: Monitoring and intervention for the transportation of dangerous goods". In: *Journal of Hazardous Materials* 152.2, pp. 516–526. ISSN: 03043894. DOI: 10.1016/j.jhazmat.2007.07.032. URL: https://linkinghub.elsevier.com/retrieve/pii/S0304389407010084.

Pnueli, A. (1977). "The temporal logic of programs". In: *18th Annual Symposium on Foundations of Computer Science*. ISSN: 0272-5428, pp. 46–57. DOI: 10.1109/SFCS.1977.32.

Polge, Julien, Jérémy Robert, and Yves Le Traon (2020). "Permissioned blockchain frameworks in the industry: A comparison". In: *ICT Express*, S2405959520301909. ISSN: 24059595. DOI: 10.1016/j.icte.2020.09.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S2405959520301909.

Polkadot (2017). *Polkadot: Decentralized Web 3.0 Blockchain Interoperability Platform*. URL: https://polkadot.network/.

Poon, Joseph and Thaddeus Dryja (2016). "The Bitcoin Lightning Network". In: White paper, p. 59. URL: https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf.

Popov, Serguei (2018). *On the Tangle, White Papers, Proofs, Airplanes, and Local Modifiers*. Medium. URL: https://blog.iota.org/on-the-tangle-white-papers-proofs-airplanes-and-local-modifiers-44683aff8fea.

Popović, K. and Ž Hocenski (2010). "Cloud computing security issues and challenges". In: *The 33rd International Convention MIPRO*, pp. 344–349. URL: https://ieeexplore.ieee.org/abstract/document/5533317/.

Private-Data, HF (2019). *Private data — hyperledger-fabricdocs master documentation*. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.2/private-data/private-data.html.

Pîrlea, George and Ilya Sergey (2018). "Mechanising blockchain consensus". In: *7th ACM SIGPLAN International Conference on Certified Programs and Proofs*. New York, USA, pp. 78–90. ISBN: 978-1-4503-5586-5. DOI: 10.1145/3167086. URL: https://doi.org/10.1145/3167086.

Project SmartLog (2019). URL: https://github.com/project-smartlog.

Provenance (2017). *A platform for product transparency | Provenance*. URL: https://www.provenance.org/business/platform.

PseudocodeJS (2017). *PseudocodeJS*. URL: https://fjp.at/public/pseudocode.js-1.1/.

PseudocodeToJavaScript (2020). *Dondi*. URL: https://dondi.lmu.build/share/intro/pseudocode2js-v02.pdf.

Putz, Benedikt et al. (2021). "EtherTwin: Blockchain-based Secure Digital Twin Information Management". In: *Information Processing & Management* 58.1. ISSN: 03064573. DOI:

10.1016/j.ipm.2020.102425. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306457320309195.

Qian, Yongfeng et al. (2018). "Towards decentralized IoT security enhancement: A blockchain approach". In: *Computers & Electrical Engineering* 72, pp. 266–273. ISSN: 00457906. DOI: 10.1016/j.compeleceng.2018.08.021. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045790618300508.

Quirolgico, Stephen et al. (2004). "Toward a Formal Common Information Model Ontology". In: *Web Information Systems – WISE Workshops*. Lecture Notes in Computer Science. Springer, pp. 11–21. ISBN: 978-3-540-23892-8 978-3-540-30481-4. DOI: 10.1007/978-3-540-30481-4_2. URL: http://link.springer.com/10.1007/978-3-540-30481-4_2.

Rabbi, Fazle (2015). "Computation Tree Logic (CTL)". In: *Bergen University College, Bergen, Norway*, p. 25. URL: https://www.uio.no/studier/emner/matnat/ifi/INF5140/v15/slides/ctl.pdf.

RadioactiveWaste (2021). *Radioactive Waste Challenge for European Integration and Enlargement: Soviet Nuclear Legacy in Central and Eastern Europe After 1989 – EuropeNow*. https://www.europenowjournal.org/2019/05/06/radioactive-waste-challenge-for-european-integration-and-enlargement-soviet-nuclear-legacy-in-central-and-eastern-europe-after-1989/.

Rafique, Wajid et al. (2020). "Complementing IoT Services through Software Defined Networking and Edge Computing: A Comprehensive Survey". In: *IEEE Communications Surveys & Tutorials*, pp. 1–1. ISSN: 1553-877X, 2373-745X. DOI: 10.1109/COMST.2020.2997475. URL: https://ieeexplore.ieee.org/document/9099866/.

Raft (2012). *Ongardie/raftscope: super hacky visualization of Raft*. https://github.com/ongardie/raftscope. (Accessed on 01/04/2021).

Rahmani, Tirdad, Daniel Oberle, and Marco Dahms (2010). "An Adjustable Transformation from OWL to Ecore". In: *Model Driven Engineering Languages and Systems*. Lecture Notes in Computer Science. Springer, pp. 243–257. ISBN: 978-3-642-16129-2. DOI: 10.1007/978-3-642-16129-2_18.

Raikwar, M. et al. (2018). "A Blockchain Framework for Insurance Processes". In: *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. ISSN: 2157-4960, pp. 1–4. DOI: 10.1109/NTMS.2018.8328731.

Rakic, Branimir et al. (2017). *OriginTrail Whitepaper*. URL: https://whitepaper.io/document/144/origintrail-whitepaper.

Raschendorfer, Alexander et al. (2019). "On IOTA as a potential enabler for an M2M economy in manufacturing". In: 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering 79, pp. 379–384. ISSN: 2212-8271. DOI: 10.1016/j.procir.2019.02.096. URL: http://www.sciencedirect.com/science/article/pii/S2212827119302136.

Reifsnider, Kenneth and Prasun Majumdar (2013). "Multiphysics Stimulated Simulation Digital Twin Methods for Fleet Management". In: *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2013-1578. URL: https://arc.aiaa.org/doi/10.2514/6.2013-1578.

Resigard (n.d.). *Resigard*. URL: https://www.igi-global.com/chapter/a-tool-for-gis-based-risk-analysis-for-transportation-of-dangerous-goods-on-road-the-ragisadr/142649.

Reyna, Ana et al. (2018). "On blockchain and its integration with IoT. Challenges and opportunities". In: *Future Generation Computer Systems* 88, pp. 173–190. ISSN: 0167-739X. DOI: 10.1016/j.future.2018.05.046. URL: http://www.sciencedirect.com/science/article/pii/S0167739X17329205.

Rhazali, Yassine, Youssef Hadi, and Abdelaziz Mouloudi (2016). "Criteria Based Evaluation for Transforming CIM to PIM". In: International conference JSSA'16, pp. 1–3. DOI: 10.13140/RG.2.1.3453.0165f. URL: https://hal.inria.fr/hal-01336330/file/JSSA%202016.pdf.

RID (2019). *Intergovernmental Organisation for International Carriage by Rail (RID)*. URL: https://otif.org/en/?page_id=1105.

Rifi, Nabil et al. (2017). "Towards using blockchain technology for IoT data access protection". In: *17th International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pp. 1–5. DOI: 10.1109/ICUWB.2017.8251003.

Rogers, EM (2003). "Diffusion of innovations". In: *Journal of Minimally Invasive Gynecology*.

Rozier, Kristin Y. (2011). "Linear Temporal Logic Symbolic Model Checking". In: *Computer Science Review* 5.2, pp. 163–203. ISSN: 15740137. DOI: 10.1016/j.cosrev.2010.06.002. URL: https://www.sciencedirect.com/science/article/abs/pii/S1574013710000407?via%3Dihub.

Salah, K. et al. (2019). "Blockchain-based Soybean Traceability in Agricultural Supply Chain". In: *IEEE Access*, pp. 1–11. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2918000. URL: https://ieeexplore.ieee.org/document/8718621/.

Sample, Char and Kim Schaffer (2013). "An Overview of Anomaly Detection". In: *IT Professional* 15.1, pp. 8–11. ISSN: 1941-045X. DOI: 10.1109/MITP.2013.7.

SAP-DG (2020). *Dangerous Goods Management (EHS-DGP) - SAP Help Portal*. URL: https://help.sap.com/viewer/47b7039ad8c44808aafbedb12029db94/6.18.11/en-US.

SCBestPractices, Ethereum (2016). *Known Attacks - Ethereum Smart Contract Best Practices*. URL: https://consensys.github.io/smart-contract-best-practices/known_attacks/.

Scherer, Mattias (2017). "Performance and Scalability of Blockchain Networks and Smart Contracts". en. In: Master of Science Programme in Computing Science and Engineering, Umea University, p. 46. URL: https://www.diva-portal.org/smash/get/diva2:1111497/FULLTEXT01.pdf.

Schmidt, Douglas C (2006). "Model-Driven Engineering". In: *IEEE Computer* 39.2, p. 9. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.9720&rep=rep1&type=pdf.

Schätz, Bernhard (2009). "Formalization and Rule-Based Transformation of EMF Ecore-Based Models". In: *Software Language Engineering*. Lecture Notes in Computer Science. Springer, pp. 227–244. ISBN: 978-3-642-00434-6. DOI: 10.1007/978-3-642-00434-6_15.

Science, Verge (2018). *(134) 88,000 tons of radioactive waste – and nowhere to put it - YouTube*. https://www.youtube.com/watch?v=YgVyPwhkoJs.

SCSafety (n.d.). *SCSafety*. URL: https://www.sitepoint.com/smart-contract-safety-best-practices-design-patterns/.

SCUpgrade (2019). Smart-Contract-Upgrade. URL: https://github.com/clearmatics/smart-contract-upgrade.

Seam-IBM (n.d.). *Seam-IBM*. https://www.forbes.com/sites/rogeraitken/2017/01/07/ibms-blockchain-consortium-with-the-seam-deploys-hyperledger-for-cotton-trading/.

Seidewitz, E. (2003). "What models mean". In: *IEEE Software* 20.5, pp. 26–32. ISSN: 1937-4194. DOI: 10.1109/MS.2003.1231147.

Selic, B. (2003). "The pragmatics of model-driven development". In: *IEEE Software* 20.5, pp. 19–25. ISSN: 0740-7459. DOI: 10.1109/MS.2003.1231146. URL: http://ieeexplore.ieee.org/document/1231146/.

Sergey, Ilya, Amrit Kumar, and Aquinas Hobor (2018). "Scilla: a Smart Contract Intermediate-Level LAnguage". In: *arXiv:1801.00687 [cs]*. URL: http://arxiv.org/abs/1801.00687.

Serpac (2020). *5 ways to transport dangerous goods: what aspects to consider*. https://info.serpac.it/blog-en/5-ways-to-transport-dangerous-goods-what-aspects-to-consider.

Sharma, Ritu and Manu Sood (2011). "Cloud SaaS: Models and Transformation". In: *Advances in Digital Image Processing and Information Technology*. Vol. 205. Series Title: Communications in Computer and Information Science. Springer, pp. 305–314. ISBN: 978-3-642-24054-6 978-3-642-24055-3. DOI: 10.1007/978-3-642-24055-3_31. URL: http://link.springer.com/10.1007/978-3-642-24055-3_31.

Shi, Peng and Jimmy Lin (2019). "Simple BERT Models for Relation Extraction and Semantic Role Labeling". In: *arXiv:1904.05255 [cs]*. arXiv: 1904.05255. URL: http://arxiv.org/abs/1904.05255.

Sigwart, Marten et al. (2019). "Blockchain-based Data Provenance for the Internet of Things". In: *9th International Conference on the Internet of Things*. Bilbao Spain: ACM, pp. 1–8. ISBN: 978-1-4503-7207-7. DOI: 10.1145/3365871.3365886. URL: https://dl.acm.org/doi/10.1145/3365871.3365886.

Silva, Alberto Rodrigues da (2015). "Model-driven engineering: A survey supported by the unified conceptual model". In: *Computer Languages, Systems & Structures* 43, pp. 139–155. ISSN: 14778424. DOI: 10.1016/j.cl.2015.06.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1477842415000408.

Singh, Amritraj (2020). "Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities". In: *Computers & Security* 88, p. 17. DOI: DOI:10.1016/j.cose.2019.101654. URL: https://www.journals.elsevier.com/computers-and-security.

Singh, Sukhpal and Inderveer Chana (2016). "A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges". In: *Journal of Grid Computing* 14.2, pp. 217–264. ISSN:

1570-7873, 1572-9184. DOI: 10.1007/s10723-015-9359-2. URL: http://link.springer.com/10.1007/s10723-015-9359-2.

Skuchain (2018). *Skuchain*. URL: https://www.skuchain.com/.

Solidity, Ethereum (2016). *Solidity Documentation*. URL: https://solidity.readthedocs.io/en/v0.5.12/.

Somers, Toni M and Klara Nelson (2001). "The Impact of Critical Success Factors across the Stages of Enterprise Resource Planning Implementations". In: *34th Hawaii International Conference on System Sciences*, p. 10. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.6418&rep=rep1&type=pdf.

Sousa, Joao, Alysson Bessani, and Marko Vukolic (2018). "A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform". In: *48th International Conference on Dependable Systems and Networks (DSN)*, pp. 51–58. ISBN: 978-1-5386-5596-2. DOI: 10.1109/DSN.2018.00018. URL: https://ieeexplore.ieee.org/document/8416470/.

Sreenivasaiah, Pradeep Kumar and Do Han Kim (2010). "Current Trends and New Challenges of Databases and Web Applications for Systems Driven Biological Research". In: *Frontiers in Physiology* 1. ISSN: 1664-042X. DOI: 10.3389/fphys.2010.00147. URL: http://journal.frontiersin.org/article/10.3389/fphys.2010.00147/abstract.

Srinivas, J, K Venkata Subba Reddy, and Dr A MOIZ Qyser (2012). "Cloud Computing Basics". In: *International Journal of Advanced Research in Computer and Communication Engineering* 1.5, p. 6. ISSN: 2278 –1021.

Srinivasan, S. (2014). *Cloud Computing Basics*. Springer. ISBN: 978-1-4614-7699-3. URL: https://www.springer.com/gp/book/9781461476986.

Staples, M et al. (2017). "Risks and opportunities for systems using blockchain and smart contracts". In: *CSIRO), Sydney*.

Suchenia, Anna et al. (2017). "Supporting BPMN Process Models with UML Sequence Diagrams for Representing Time Issues and Testing Models". In: *Artificial Intelligence and Soft Computing*. Lecture Notes in Computer Science. Springer, pp. 589–598. ISBN: 978-3-319-59059-2 978-3-319-59060-8. DOI: 10.1007/978-3-319-59060-8_53. URL: http://link.springer.com/10.1007/978-3-319-59060-8_53.

Sukhwani, H. et al. (2018). "Performance Modeling of Hyperledger Fabric (Permissioned Blockchain Network)". In: *17th International Symposium on Network Computing and Applications (NCA)*, pp. 1–8. DOI: 10.1109/NCA.2018.8548070.

Sukhwani, Harish et al. (2017). "Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric)". In: *36th Symposium on Reliable Distributed Systems (SRDS)*. Hong Kong, Hong Kong: IEEE, pp. 253–255. ISBN: 978-1-5386-1679-6. DOI: 10.1109/SRDS.2017.36. URL: http://ieeexplore.ieee.org/document/8069090/.

Sultana, Tanin and Khan A. Wahid (2019). "Choice of Application Layer Protocols for Next Generation Video Surveillance Using Internet of Video Things". In: *IEEE Access* 7, pp. 41607–41624. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2907525.

SyncFab (2020). *SyncFab*. URL: https://syncfab.com/.

Syriani, Eugene et al. (2013). "AToMPM: A Web-based Modeling Environment". In: *16th International Conference on Model Driven Engineering Languages and Systems*, p. 5. URL: http://ceur-ws.org/Vol-1115/demo4.pdf.

T-Mining (2017). *Port Of Antwerp*. URL: https://www.portofantwerp.com/en/news/antwerp-start-t-mining-develops-blockchain-solution-safe-efficient-container-release.

Tallon, Paul P., Kenneth L. Kraemer, and Vijay Gurbaxani (2000). "Executives' Perceptions of the Business Value of Information Technology: A Process-Oriented Approach". In: *Journal of Management Information Systems* 16.4, pp. 145–173. ISSN: 0742-1222. DOI: 10.1080/07421222.2000.11518269. URL: https://doi.org/10.1080/07421222.2000.11518269.

Tang, Changbing et al. (2020). "Incentivizing Honest Mining in Blockchain Networks: A Reputation Approach". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 67.1, pp. 117–121. ISSN: 1549-7747, 1558-3791. DOI: 10.1109/TCSII.2019.2901746. URL: https://ieeexplore.ieee.org/document/8653347/.

Tao, Fei et al. (2018). "Digital twin-driven product design, manufacturing and service with big data". In: *The International Journal of Advanced Manufacturing Technology* 94.9, pp. 3563–3576. ISSN: 1433-3015. DOI: 10.1007/s00170-017-0233-1. URL: https://doi.org/10.1007/s00170-017-0233-1.

Tao, Fei et al. (2019). "Digital Twin in Industry: State-of-the-Art". In: *IEEE Transactions on Industrial Informatics* 15.4, pp. 2405–2415. ISSN: 1941-0050. DOI: 10.1109/TII.2018.2873186.

Taylor, I (2017). *Maersk and IBM launch blockchain-based cross-border supply chain solution*. URL: https://postandparcel.info/79170/news/e-commerce/maersk-and-ibm-launch-blockchain-based-cross-border-supply-chain-solution/.

Team, Polkadot (2017). *Polkadot: Vision for a Heterogeneous Multi-chain Framework*. URL: https://www.win.tue.nl/~mholende/seminar/references/ethereum_polkadot.pdf.

Telegraph (2016). *Mafia, toxic waste and a deadly cover up in an Italian paradise: 'They've poisoned our land and stolen our children'*. https://www.telegraph.co.uk/news/0/mafia-toxic-waste-and-a-deadly-cover-up-in-an-italian-paradise-t/.

Tendermint (2021). *Tendermint*. URL: https://tendermint.com/.

Tendermint-SDK (2017). *Ethereum on Tendermint using Cosmos-SDK!* URL: https://github.com/cosmos/ethermint.

Teslya, N. and I. Ryabchikov (2017). "Blockchain-based platform architecture for industrial IoT". In: *21st Conference of Open Innovations Association (FRUCT)*, pp. 321–329. DOI: 10.23919/FRUCT.2017.8250199.

Textor, Andreas, Jeanne Stynes, and Reinhold Kroeger (2010). "Transformation of the Common Information Model to OWL". In: *Current Trends in Web Engineering*. Lecture Notes in Computer Science. Springer, pp. 163–174. ISBN: 978-3-642-16984-7 978-3-642-16985-4. DOI: 10.1007/978-3-642-16985-4_15. URL: http://link.springer.com/10.1007/978-3-642-16985-4_15.

Tian (2014). *Slimcoin-project*. URL: https://github.com/slimcoin-project/slimcoin-project.github.io/blob/master/whitepaperSLM.pdf.

Tian, Feng (2017). "A supply chain traceability system for food safety based on HACCP, blockchain Internet of things". In: *International Conference on Service Systems and Service Management*. ISSN: 2161-1904, pp. 1–6. DOI: 10.1109/ICSSSM.2017.7996119.

Tijan, Edvard et al. (2019). "Blockchain Technology Implementation in Logistics". In: *Sustainability* 11.4, p. 1185. ISSN: 2071-1050. DOI: 10.3390/su11041185. URL: http://www.mdpi.com/2071-1050/11/4/1185.

Tixier, J. et al. (2002). "OSIRIS: software for the consequence evaluation of transportation of dangerous goods accidents". In: *Environmental Modelling & Software* 17.7, pp. 627–637. ISSN: 13648152. DOI: 10.1016/S1364-8152(02)00025-7. URL: http://linkinghub.elsevier.com/retrieve/pii/S1364815202000257.

Tjoa, A Min, Li Xu, and Sohail Chaudhry (2006). "Research and practical issues of enterprise information systems". In: *International Conference on Research and Practical Issues of Enterprise Information Systems*. DOI: 10.1007/978-3-319-99040-8. URL: https://www.springer.com/gp/book/9783319990392.

TKI-Dinalog (2017). *Blockchain and logistics project TKI-Dinalog—Logistics and FinTech.* URL: http://logisticsandfintech.com/blockchain-and-logistics-project-tki-dinalog/.

Tolmach, Palina et al. (2021). "A Survey of Smart Contract Formal Specification and Verification". In: *ACM Computing Surveys* 54 (7), pp 1–38. DOI: https://doi.org/10.1145/3464421. URL: https://dl.acm.org/doi/fullHtml/10.1145/3464421.

Torretta, Vincenzo et al. (2017). "Decision support systems for assessing risks involved in transporting hazardous materials: A review". In: *Safety Science* 92, pp. 1–9. ISSN: 09257535. DOI: 10.1016/j.ssci.2016.09.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925753516302119.

Treleaven, P., R. Gendal Brown, and D. Yang (2017). "Blockchain Technology in Finance". In: *Computer* 50.9, pp. 14–17. ISSN: 1558-0814. DOI: 10.1109/MC.2017.3571047.

Tribis, Youness, Abdelali El Bouchti, and Houssine Bouayad (2018). "Supply Chain Management based on Blockchain: A Systematic Mapping Study". In: *MATEC Web of Conferences*. ISSN: 2261-236X. DOI: 10.1051/matecconf/201820000020. URL: https://www.matec-conferences.org/10.1051/matecconf/201820000020.

Truyer, Frank (2006). *The Fast Guide to Model Driven Architecture*. URL: https://www.omg.org/mda/mda_files/Cephas_MDA_Fast_Guide.pdf.

Tsankov, Petar et al. (2018). "Securify: Practical Security Analysis of Smart Contracts". In: *SIGSAC Conference on Computer and Communications Security*. DOI: https://doi.org/10.1145/3243734.3243780. URL: https://dl.acm.org/doi/abs/10.1145/3243734.3243780.

UNECE (2017). *European Agreement concerning the International Carriage of Dangerous Goods by Road*. URL: https://www.unece.org/fileadmin/DAM/trans/danger/publi/adr/adr2017/ADR2017e_web.pdf.

UNEP (1989). *Basel Convention on the control of transboundary movements of hazardous wastes and their disposal*. URL: http://www.basel.int/TheConvention/Overview/tabid/1271/Default.aspx.

VanderWeele, Tyler J. and James M. Robins (2007). "Directed Acyclic Graphs, Sufficient Causes, and the Properties of Conditioning on a Common Effect". In: *American Journal of Epidemiology* 166.9, pp. 1096–1104. ISSN: 1476-6256, 0002-9262. DOI: 10.1093/aje/kwm179. URL: https://academic.oup.com/aje/article-lookup/doi/10.1093/aje/kwm179.

Velte, Toby, Anthony Velte, and Robert Elsenpeter (2009). *Cloud Computing, A Practical Approach*. USA: McGraw-Hill, Inc. ISBN: 978-0-07-162694-1.

Verwijmeren, Martin (2004). "Software component architecture in supply chain management". In: *Computers in Industry* 53.2, pp. 165–178. ISSN: 01663615. DOI: 10.1016/j.compind.2003.07.004. URL: http://linkinghub.elsevier.com/retrieve/pii/S0166361503001854.

Vidales, Miguel Ángel Sánchez, Ana Fermoso García, and Luis Joyanes Aguilar (2005). "From The Platform Independent Model (Pim) To The Final Code Model (FCM) According To The Model Driven Architecture (MDA)". In: *IADIS International Conference on Applied Computing*, p. 5. URL: http://www.iadisportal.org/digital-library/from-the-platform-independent-model-pim-to-the-final-code-model-fcm-according-to-the-model-driven-architecture-mda.

Vo, Minh Hoang Lien and Quang Hoang (2020). "Transformation of UML class diagram into OWL Ontology". In: *Journal of Information and Telecommunication* 4.1, pp. 1–16. ISSN: 2475-1839, 2475-1847. DOI: 10.1080/24751839.2019.1686681. URL: https://www.tandfonline.com/doi/full/10.1080/24751839.2019.1686681.

Voas, Jeffrey M (2016). *Networks of 'things'*: NIST SP 800-183. Gaithersburg, MD: National Institute of Standards and Technology. DOI: 10.6028/NIST.SP.800-183. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-183.pdf.

Voirin, Jean-Luc (2017). *Model-based System and Architecture Engineering with the Arcadia Method*. Elsevier.

Vyper (2018). URL: https://vyper.readthedocs.io/en/v0.1.0-beta.13/.

VyperDetails (n.d.). URL: https://vyper.readthedocs.io/en/stable/.

W3C (2020). *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)*. URL: https://www.w3.org/TR/owl2-syntax/.

Walmart and IBM (2017). *Walmart and IBM*. URL: https://bitcoinmagazine.com/articles/walmart-testing-blockchain-technology-for-supply-chain-management.

Wan, Paul Kengfai, Lizhen Huang, and Halvor Holtskog (2020). "Blockchain-Enabled Information Sharing Within a Supply Chain: A Systematic Literature Review". In: *IEEE Access* 8, pp. 49645–49656. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2980142. URL: https://ieeexplore.ieee.org/document/9032112/.

Wang, Huaimin et al. (2018a). "Blockchain challenges and opportunities: a survey". In: *International Journal of Web and Grid Services* 14.4, p. 352. ISSN: 1741-1106, 1741-1114. DOI: 10.1504/IJWGS.2018.10016848. URL: http://www.inderscience.com/link.php?id=10016848.

Wang, Jingyuan et al. (2017). "No Longer Sleeping with a Bomb: A Duet System for Protecting Urban Safety from Dangerous Goods". In: *23rd International Conference on Knowledge Discovery and Data Mining*. New York, USA, pp. 1673–1681. ISBN: 978-1-4503-4887-4. DOI: 10.1145/3097983.3097985. URL: http://doi.acm.org/10.1145/3097983.3097985.

Wang, N., X. Huang, and D. Wei (2016). "Route selection for dangerous goods based on D numbers". In: *Chinese Control and Decision Conference (CCDC)*, pp. 6651–6656. DOI: 10.1109/CCDC.2016.7532194.

Wang, Wenbo et al. (2018b). "A Survey on Consensus Mechanisms and Mining Management in Blockchain Networks". In: *IEEE Access*. DOI: 10.1109/ACCESS.2019.2896108. URL: https://ieeexplore.ieee.org/document/8629877.

Wang, Yingli, Jeong Hugh Han, and Paul Beynon-Davies (2019a). "Understanding blockchain technology for future supply chains: a systematic literature review and research agenda". In: *Supply Chain Management: An International Journal* 24.1, pp. 62–84. ISSN: 1359-8546. DOI: 10.1108/SCM-03-2018-0148. URL: https://www.emerald.com/insight/content/doi/10.1108/SCM-03-2018-0148/full/html.

Wang, Yuepeng et al. (2019b). "Formal Specification and Verification of Smart Contracts for Azure Blockchain". In: *arXiv:1812.08829 [cs]*. arXiv: 1812.08829. URL: http://arxiv.org/abs/1812.08829.

WasteCollection (2020). *Waste collection and transportation permit*. URL: https://guichet.public.lu/en/entreprises/urbanisme-environnement/dechets-subst-dangereuses/transport-dechets/transport-collecte-dechets.html.

Weber, Ingo et al. (2016). "Untrusted Business Process Monitoring and Execution Using Blockchain". In: *Business Process Management*. Vol. 9850. Cham: Springer, pp. 329–347. ISBN: 978-3-319-45347-7 978-3-319-45348-4. DOI: 10.1007/978-3-319-45348-4_19. URL: http://link.springer.com/10.1007/978-3-319-45348-4_19.

Werner, Hartmut (2008). *Supply Chain Management: Grundlagen, Strategien, Instrumente und Controlling*. Gabler Lehrbuch. ISBN: 978-3-8349-0504-8.

WikiTezos (2021). *Tezos White Paper - Tezos Agora Wiki*. URL: https://wiki.tezosagora.org/whitepaper.

WISER (n.d.). *WISER Home*. URL: https://wiser.nlm.nih.gov/.

Witthaut, Markus et al. (2017). "Smart Objects and Smart Finance for Supply Chain Management". In: *Logistics Journal : referierte Veröffentlichungen* 2017.10. ISSN: 1860-7977. DOI: 10.2195/lj_NotRev_witthaut_en_201710_01. URL: http://www.logistics-journal.de/not-reviewed/2017/10/4610.

Wohrer, M. and U. Zdun (2018). "Smart contracts: security patterns in the ethereum ecosystem and solidity". In: *International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pp. 2–8. DOI: 10.1109/IWBOSE.2018.8327565.

Wohrer, Maximilian and Uwe Zdun (2018). "Design Patterns for Smart Contracts in the Ethereum Ecosystem". In: *International Conference on Internet of Things (iThings)*. Halifax, NS, Canada: IEEE, pp. 1513–1520. ISBN: 978-1-5386-7975-3. DOI: 10.1109/Cybermatics_2018.2018.00255. URL: https://ieeexplore.ieee.org/document/8726782/.

Wood, Gavin (2014). *Ethereum: A Secure Decentralised Generalised Transaction Ledge*. URL: https://ethereum.github.io/yellowpaper/paper.pdf.

Wu, Haoyan et al. (2017). "A Distributed Ledger for Supply Chain Physical Distribution Visibility". In: *Information* 8.4, p. 137. ISSN: 2078-2489. DOI: 10.3390/info8040137. URL: http://www.mdpi.com/2078-2489/8/4/137.

XACML 3.0 (n.d.). URL: http://www.datypic.com/sc/xacml30/e-xacml_Policy.html.

Xenvis (n.d.). *Xenvis*. URL: https://www.ess.co.at/XENVIS/x01.html.

Xia, Jiang and Liu Yongjun (2017). "Trust Evaluation Model for Supply Chain Enterprises under Blockchain Environment". In: *7th International Conference on Social Network, Communication and Education (SNCE)*. ISBN: 978-94-6252-386-9. DOI: 10.2991/snce-17.2017.129. URL: http://www.atlantis-press.com/php/paper-details.php?id=25883096.

Xu, Xiwei et al. (2016). "The Blockchain as a Software Connector". In: *13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*. Venice, Italy: IEEE, pp. 182–191. ISBN: 978-1-5090-2131-4. DOI: 10.1109/WICSA.2016.21. URL: http://ieeexplore.ieee.org/document/7516828/.

Xu, Xiwei et al. (2017). "A Taxonomy of Blockchain-Based Systems for Architecture Design". In: 2017 IEEE International Conference on Software Architecture (ICSA). Gothenburg, Sweden, pp. 243–252. ISBN: 978-1-5090-5729-0. DOI: 10.1109/ICSA.2017.33. URL: http://ieeexplore.ieee.org/document/7930224/.

Xu, Yang et al. (2018). "Towards Secure Network Computing Services for Lightweight Clients Using Blockchain". In: *Wireless Communications and Mobile Computing* 2018, pp. 1–12. ISSN: 1530-8669, 1530-8677. DOI: 10.1155/2018/2051693. URL: https://www.hindawi.com/journals/wcmc/2018/2051693/.

Yaga, Dylan et al. (2018). *Blockchain technology overview*. National Institute of Standards and Technology. DOI: 10.6028/NIST.IR.8202. URL: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf.

Yang, Jian and Hong Shen (2019). "Blockchain Consensus Algorithm Design Based on Consistent Hash Algorithm". In: *20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. Gold Coast, Australia: IEEE, pp. 461–466. ISBN: 978-1-72812-616-6. DOI: 10.1109/PDCAT46702.2019.00090. URL: https://ieeexplore.ieee.org/document/9029081/.

Yang, Yuchen et al. (2017). "A Survey on Security and Privacy Issues in Internet-of-Things". In: *IEEE Internet of Things Journal* 4.5, pp. 1250–1258. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2694844. URL: http://ieeexplore.ieee.org/document/7902207/.

Yang, Zheng (2018). "Formal Process Virtual Machine for Smart Contracts Verification". In: *International Journal of Performability Engineering*. DOI: 10.23940/ijpe.18.08.p9.17261734. URL: http://www.ijpe-online.com/EN/10.23940/ijpe.18.08.p9.17261734.

Yaqoob, I. et al. (2017). "Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges". In: *IEEE Wireless Communications* 24.3, pp. 10–16. ISSN: 1558-0687. DOI: 10.1109/MWC.2017.1600421.

Yasumoto, Keiichi, Hirozumi Yamaguchi, and Hiroshi Shigeno (2016). "Survey of Real-time Processing Technologies of IoT Data Streams". In: *Journal of Information Processing* 24.2, pp. 195–202. ISSN: 1882-6652. DOI: 10.2197/ipsjjip.24.195. URL: https://www.jstage.jst.go.jp/article/ipsjjip/24/2/24_195/_article.

Ye, Congcong et al. (2018). "Analysis of Security in Blockchain: Case Study in 51%-Attack Detecting". In: *5th International Conference on Dependable Systems and Their Applications (DSA)*, pp. 15–24. ISBN: 978-1-5386-9266-0. DOI: 10.1109/DSA.2018.00015. URL: https://ieeexplore.ieee.org/document/8563187/.

Yu, Guangsheng et al. (2018). "An Optimized Round-Robin Scheduling of Speakers for Peers-to-Peers-Based Byzantine Faulty Tolerance". In: *Globecom Workshops (GC Wkshps)*, pp. 1–6. ISBN: 978-1-5386-4920-6. DOI: 10.1109/GLOCOMW.2018.8644251. URL: https://ieeexplore.ieee.org/document/8644251/.

Yuan, Yong and Fei-Yue Wang (2016). "Towards blockchain-based intelligent transportation systems". In: *19th International Conference on Intelligent Transportation Systems (ITSC)*. ISSN: 2153-0017, pp. 2663–2668. DOI: 10.1109/ITSC.2016.7795984.

Zografos, K. G., G. M. Vasilakis, and I. M. Giannouli (2000). "Methodological framework for developing decision support systems (DSS) for hazardous materials emergency response operations". In: *Journal of Hazardous Materials* 71.1-3, pp. 503–521. ISSN: 0304-3894. DOI: 10.1016/s0304-3894(99)00096-5.

Zografos, Konstantinos G. and Konstantinos N. Androutsopoulos (2008). "A decision support system for integrated hazardous materials routing and emergency response decisions". In: *Transportation Research Part C: Emerging Technologies* 16.6, pp. 684–703. ISSN: 0968090X. DOI: 10.1016/j.trc.2008.01.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X08000211.

Zyskind, Guy, Oz Nathan, and Alex Pentland (2018). "Enigma: Decentralized Computation Platform with Guaranteed Privacy". In: *New Solutions for Cybersecurity*. DOI: 10.7551/mitpress/11636.003.0018. URL: https://web.media.mit.edu/~guyzys/data/enigma_full.pdf.

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

université PARIS-SACLAY

Titre : Utilisation de la technologie blockchain pour l'amélioration de la confiance dans les processus de logistique et de transport

Mots clés : Blockchain, Logistique, Confiance, Flux de travail, IoT, Smart Contract

Résumé : Cette thèse aborde le problème général du transport sûr et sécurisé des marchandises dangereuses (TMD). Le TMD est très compliqué à gérer en raison des risques pour l'environnement et la vie humaine. Actuellement, il souffre d'un manque d'efficacité, de confiance et de transparence. Dans cette thèse, nous proposons une nouvelle méthode pour spécifier les aspects du flux de travail du TMD en considérant toutes les étapes du processus du TMD pendant tout son cycle de vie. Cette méthode vise à faciliter les spécifications du système de gestion du flux de travail TMD qui est entièrement basé sur les cadres réglementaires existants assurant la conformité, la confiance et la transparence de tous les processus sous-jacents. La méthode de conception de système proposée est basée sur l'approche dite d'architecture dirigée par le modèle (MDA) et l'améliore pour prendre en compte les propriétés de la blockchain. La première étape est l'analyse formelle du processus de TMD et son alignement avec les cadres réglementaires. La méthode de conception proposée vise, à ce stade, à permettre la définition et la vérification formelles de la conception du système au regard des cadres réglementaires. Les étapes suivantes de la méthode s'appuient fortement sur la transformation de modèle qui est un aspect important de la méthode de conception proposée. La transformation du modèle permet de découvrir automatiquement les composants du système homologue et les interactions autorisées. La dernière étape de l'ensemble des transformations du modèle est la spécification des profils de jumeaux numériques pour toutes les parties prenantes potentielles. Toutes les interactions dans le monde réel entre les parties prenantes sont transformées en interactions dans le monde numérique, tandis que les interactions avec l'environnement sont réalisées grâce à l'utilisation de l'IoT. L'approche proposée permet les interactions entre les composants des systèmes (jumeaux numériques, dispositifs IoT, etc.) uniquement si cela est conforme au cadre réglementaire. Grâce à la technologie blockchain, notre méthode de conception permet d'améliorer la confiance et la transparence dans le processus de TMD du point de vue des collaborations entre les parties prenantes. Les capacités technologiques des contrats intelligents sont également une pierre angulaire de la solution proposée. Cette thèse contribue également à améliorer la sémantique des contrats intelligents pour capturer les spécifications de gestion de la chaîne d'approvisionnement ainsi que les spécificités des marchandises dangereuses en termes de transport. Les concepts dynamiques liés à la gestion de la chaîne d'approvisionnement des marchandises dangereuses tels que les contraintes temporelles et géographiques, la certification numérique, la détection des anomalies et les contrats intelligents multiparties, la gestion des urgences et le partage des responsabilités ont été abordés au niveau du contrat intelligent. En particulier, cette thèse propose d'appliquer la logique temporelle pour la spécification et la vérification formelles des contrats intelligents. Cette thèse propose une approche intégrée pour la blockchain et l'IoT afin de prendre en charge les aspects dynamiques dans la chaîne d'approvisionnement des marchandises dangereuses. Les données collectées à partir de divers dispositifs IoT le long de la chaîne d'approvisionnement physique (marchandises, véhicules, frontières des pays, etc.) sont transmises à la blockchain et traitées ensuite par le système en suivant la logique de flux de travail qui a été spécifiée et en déclenchant automatiquement les smart contracts connexes et les actions correspondantes. La dernière contribution de cette thèse est la mise en œuvre d'un système de preuve de concept pour valider les différents aspects de la contribution, à savoir la méthode de conception, l'assurance de confiance et de transparence, et le déclenchement automatique des actions et des flux d'informations.

Title : Using the Blockchain Technology for Trust Improvement of processes in Logistics and Transportation

Keywords : Blockchain, Logistics, Trust, Workflow, IoT, Smart Contract

Abstract : This thesis address the general problem of safe and secure transport of dangerous goods (TDG). The TDG is very complicated to manage because of risk for the environment and human life. Currently, it suffers from a lack of efficiency, trust, and transparency. In this thesis, we propose a novel method to specify the workflow aspects of TDG by considering all TDG process stages during its entire lifecycle. This method aims to facilitate the specifications of the TDG workflow management system that is entirely based on existing regulatory frameworks ensuring the compliance, trust, and transparency of all underlying processes. The proposed system design method is based on the so-called model-driven architecture (MDA) approach and enhancing it to consider blockchain properties. The first stage is the formal analysis of the process of TDG and its alignment with the regulatory frameworks. The proposed design method aims, at this stage, to allow the formal definition and verification of the design of the system with regard to the regulatory frameworks. The next stages of the method rely strongly on the model transformation that is a salient aspect of the proposed design method. Model transformation allows to automatically discover peer system components and authorized interactions. The last stage of the whole model transformations is the specification of digital twin profiles for all potential stakeholders. All the interactions in the real world between stakeholders are transformed into interactions in the digital world, while the interactions with the environment are achieved through the use of IoT. The proposed approach enables interactions between components of the systems (digital twins, IoT devices, etc.) only if this is compliant with the regulatory framework. Thanks to blockchain technology, our design method allows improving trust and transparency in the process of TDG from the perspective of stakeholder collaborations. Smart contract technological capabilities are also a cornerstone of the proposed solution. This thesis also contributes to improving the semantic of smart contracts to capture supply chain management specifications as well as dangerous goods specificities in terms of transportation. Dynamic concepts related to the supply chain management of dangerous goods such as time-related and geographic constraints, digital certification, anomaly detection and multi-party smart contract, managing emergencies, and shared responsibility have been addressed at the level of the smart contract. In particular, this thesis proposes applying temporal logic for the formal specification and verification of smart contracts. This thesis proposes an integrated approach for blockchain and IoT to support the dynamic aspects in the supply chain of dangerous goods. Data collected from various IoT devices along the physical supply chain (goods, vehicles, country borders, etc.) are transmitted to the blockchain and further processed by the system following the workflow logic that was specified and automatically triggering related smart contracts and corresponding actions. The last contribution in this thesis is the implementation of a proof-of-concept system to validate the different aspects of the contribution, namely the design method, the trust and transparency assurance, and the automatic triggering of actions and information flows.

Maison du doctorat de Université Paris-Saclay
2e étage, aile ouest, École normale supérieure Paris-Saclay
4 avenue des Sciencs
91190 Gif-sur-Yvette, France