Enhancing the traceability of B2B logistic chains using blockchain, IoT and Deep Learning

Mohamed Ahmed Mohamed

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAS013

Thèse de doctorat

Amélioration de la traçabilité des chaînes logistiques B2B à l'aide de la Blockchain, l'IoT et le Deep Learning

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 École doctorale IP Paris (ED IP Paris)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Évry, le 27 octobre 2021, par

MOHAMED AHMED MOHAMED

Composition du Jury :

Salima BENBERNOU Professeure, Université de Paris	Présidente
Nazim AGOULMINE Professeur, Université d'Évry	Rapporteur
Manuele KIRSCH PINHEIRO Maître de Conférences HdR, Université Paris 1	Rapporteuse
Chantal TACONET Maître de Conférences HdR, Télécom SudParis	Directrice de thèse
Amel BOUZEGHOUB Professeure, Télécom SudParis	Co-encadrante de thèse
Mohamed OULD MOHAMED LEMINE Docteur, Directeur Général Adjoint, MyTower	Co-encadrant de thèse
Sophie CHABRIDON Professeure, Télécom SudParis	Co-encadrante de thèse

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAS013

Thèse de doctorat

Enhancing the traceability of B2B logistic chains using blockchain, IoT and Deep Learning

PhD thesis of The Institut Polytechnique de Paris
prepared at Télécom SudParis

Doctoral school n°626 IP Paris Doctoral School (ED IP Paris)
Ph.D. Specialty: Computer Sciences

Thesis presented and defended in Évry, on October 27, 2021, by

MOHAMED AHMED MOHAMED

Members of the jury :

Salima BENBERNOU Professor, Université de Paris	President
Nazim AGOULMINE Professor, Université d'Évry	Reviewer
Manuele KIRSCH PINHEIRO Associate Professor, Université Paris 1	Reviewer
Chantal TACONET Associate Professor, Télécom SudParis	Thesis supervisor
Amel BOUZEGHOUB Professor, Télécom SudParis	Thesis co-supervisor
Mohamed OULD MOHAMED LEMINE PhD, COO of MyTower	Thesis co-supervisor
Sophie CHABRIDON Professor, Télécom SudParis	Thesis co-supervisor

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisors, Chantal TACONET, Amel BOUZEGHOUB et Sophie CHABRIDON, who throughout their support and kindness helped me to achieve this project.

I would like also to express my deepest gratitude to my industrial supervisor, Mohamed OULD, for his trust and support during the whole project. Many thanks also, to the MyTower company for supporting this thesis.

My thanks go also to the thesis reviewers Manuele KIRSCH PINHEIRO and Nazim AGOULMINE for their relevant and instructive reports. I would like also to say thanks to the jury president, Salima BENBERNOU, for accepting to be president of the thesis jury.

I could not end these acknowledgments without thanking my parents and my wife for their unconditioned supports.

I would like also to dedicate this work to my children's.

Finally, I would like to thank my colleagues at MyTower, Télécom Sud Paris and all my friends.

Acronyms

AI Artificial Intelligence

B2B Business to Business

B2C Business to Customer

C2C Customer to Customer

DL Deep Learning

EDI Electronic Data Interchange

GPS Global Positioning System

IoT Internet of Things

IIoT Industrial Internet of Things

IPFS InterPlanetary File System

LPWAN Low Power Wide Area Networks

ML Machine Learning

PBFT Practical Byzantine Fault Tolerance

PoET Proof of Elapsed Time

PoW Proof of Work

PoS Proof of Stake

RAFT Replicated And Fault Tolerant

RFID Radio Frequency IDentification

SGX Software Guard Extensions

TEE Trusted Execution Environment

Résumé

Les systèmes d'information des entreprises connaissent aujourd'hui une évolution rapide. Dans le contexte de la chaîne logistique, cette évolution est marquée par l'introduction des nouvelles technologies comme l'Internet des Objets. Puisque la chaîne logistique implique plusieurs intervenants, elle exige le partage des données entre les intervenants pour assurer la traçabilité des produits tout au long de la chaîne logistique.

Dans leurs architectures actuelles, les systèmes de traçabilité sont centralisés chez l'un des intervenants de la chaîne logistique. Par conséquence, ils ne garantissent pas le partage sécurisé des données de traçabilité et ne permettent pas d'assurer l'accord des intervenants sur les données partagées et la bonne application des règles de traitement convenues pour ces données.

Par ailleurs, la qualité des données de l'Internet des Objets reste un frein au développement des nouvelles architectures de traçabilité. La prise en compte de la qualité de données de ces objets permettra d'assurer la confiance des intervenants dans les données collectées et de faciliter l'automatisation du processus de collecte des données de traçabilité.

L'introduction de l'internet des objets comme nouvelle source de données dans ces architectures génère un volume important de données, et pour tirer profit de ce gros volume de données, il faut assurer un traitement efficace et intelligent des données collectées. Cela permettra d'améliorer les décisions prises par le système de traçabilité.

Plusieurs travaux ont été proposés dans la littérature en utilisant la blockchain pour surmonter les problèmes susmentionnés. Cependant, peu de travaux se sont intéressés à cette problématique de traçabilité dans les chaînes logistiques B2B, caractérisées par les besoins de limitation d'accès aux données et les contraintes de performance sur le temps de réponse de ces architectures.

L'objectif principal de cette thèse est d'aller au-delà de l'état de l'art actuel et de proposer une architecture de traçabilité basée sur la blockchain et l'Internet des Objets et adaptée aux besoins des chaînes logistiques B2B.

Pour répondre aux besoins susmentionnés du contexte logistique B2B, la nouvelle architecture proposée dans le cadre de cette thèse utilise les blockchains permissives qui offrent une gestion des droits d'accès aux données et permettent de répondre aux exigences de performances des intervenants de la chaîne logistique.

Nous utilisons également les smart contracts pour implémenter le processus de traçabilité. Ceci permet d'assurer l'accord de l'ensemble des intervenants sur la bonne exécution de ce processus tel qu'il a été convenu. Afin de faciliter le déploiement du smart contract proposé dans différents contextes de traçabilité des chaînes logistique sans avoir besoins des développe-

ments spécifiques, le smart contract proposé dans le cadre de cette thèse est générique. Il gère également les points de suivi logistique qui sont utilisés largement aujourd'hui dans le domaine logistique. Il permettra aussi de prendre des décisions automatiques basées sur des données collectées automatiquement par les objets connectés.

La qualité des données collectées automatiquement par les objets connectés a un impact direct sur la qualité des décisions prises automatiquement par le système proposé. Pour assurer la qualité des données collectés et par conséquence, la qualité des décisions, notre proposition inclut un module de qualification des données des objets connectés. Ce module fournit aux intervenants des données de haute qualité et un contrôle et suivi fins de la qualité des données basés sur les exigences qualité des intervenants.

De plus, l'Internet des Objets génère un volume important de données et pour assurer un traitement efficace et intelligent de cet important volume de données, le smart contract de traçabilité proposé est renforcé avec des capacités d'apprentissage en utilisant l'apprentissage profond.

En outre, toutes les propositions de la thèse ont été évaluées et leurs évaluations montrent des résultats prometteurs pour le déploiement de l'architecture de traçabilité proposée dans la chaîne logistique afin d'améliorer toujours et encore la traçabilité.

Abstract

Nowadays, company information systems are witnessing a very fast evolution. In the logistic chain context, this fast evolution is characterized by the introduction of new technologies such as the Internet of Things. Since the logistic chain involves multiples stakeholders, it requires data sharing among all these stakeholders to ensure products traceability in the whole logistic chain. Traditional traceability systems are used by the stakeholders for traceability data sharing. However, these traditional systems are centralized and do not guarantee the secure sharing of data and the stakeholders agreement on the shared data and its processing rules. Many works have been proposed in the literature using blockchain to overcome the above issues. The main objective of this thesis is to go beyond the current state of the art and propose a blockchain-IoT based traceability architecture adapted to the B2B logistic chain context. In addition, the IoT data quality is a hindrance to the development of this kind of traceability architectures. To overcome this issue and ensure the stakeholders trust in the collected data and facilitate the automation of the traceability data collection process, the proposed architecture includes an IoT data qualification module providing the stakeholders with high data quality and fine data quality control and monitoring based on the stakeholders quality requirements. Moreover, the IoT generates a huge data volume and to ensure an efficient and intelligent data management of this huge data volume, the proposed architecture is boosted with learning capabilities using Deep Learning. Furthermore, all the thesis propositions have been evaluated and their evaluation shows promising results for the deployment of the proposed traceability architecture in the logistic chain to help the stakeholders in their traceability daily life struggle.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Context

Goods moving is not a recent human activity, as witnessed by the famous Silk Road [13]. Like all the other human activities, it has been overwhelmed by the last centuries communication and transport revolution.

The transport domain revolution has contributed to the transportation cost decrease and allowed the transport of raw materials to be transformed far from their sources and to be distributed as finished or semi-finished products all around the globe. The activity behind this process is called in our days the supply chain. Similarly, the communication revolution has accelerated the information flow sharing. Therefore, it has participated in the development of efficient supply chain.

The International Organization for Standardization (ISO) defines the supply chain as "*a linked set of resources and processes that begins with the sourcing of raw material and extends through the delivery of products or services to the end user across the modes of transport. It may include vendors, manufacturing facilities, logistics providers, internal distribution centers, distributors, wholesalers, and other entities that lead to the end user*" [59]. In this thesis, we focus on the Logistics part of the supply chain.

The logistics term is originated from the military domain. It has been used in 1811 by Wilhelm Müller in his book "The elements of the science of war". In this book, he stated that "*Logistics embraces all the details of moving and supplying armies*" [98]. Despite the growth of the supply chain domain, the logistics is still considered today as the most important part of the supply chain domain.

The Council of Supply Chain Management Professionals (CSCMP), defines the logistics as "*the process of planning, implementing, and controlling procedures for the efficient and effective transportation and storage of goods including services, and related information from the point of origin to the point of consumption for the purpose of conforming to customer requirements*" [34].

The term logistic chain designates "*a dual network consisting of goods and information flows down-and upstream*" [37].It is responsible of goods movement, from a point of origin to a destination point. To achieve the goods movement mission, the logistic chain relies on transportation, storage, and other services. Therefore, the logistic chain involves multiple stakeholders. The ISO defines stakeholder as a "person or entity having a vested interest in the organization's performance, success, or the impact of its activities" [59].

In this thesis, we focus on the activity related to the transport operation and the sharing of all its related data among all the stakeholders. A transport operation involves at least three stakeholders: a *shipper* (at the origin of the transport request), a *carrier* (in charge of transport operation) and a *consignee* (the recipient of the transported shipment). Many other stakeholders can also be involved in this process, among them: Logistic Service Providers, Customs, Insurance companies and Banks. We refer to an object handled by the stakeholders in the transport operation as a *shipment*.

A shipment is defined by [60] as "*Household Goods (HHG) items transported and/or stored under the terms of a single bill of lading, waybill, or contract of carriage irrespective of the quantity or number of containers, packages, or pieces*". In this thesis, we use the term *shipment* to designate any object entrusted to the carrier by the shipper, to be forwarded to the consignee. All the involved stakeholders need to share traceability data about the shipment progress in the logistic chain.

The shipments movements inside the logistic chain are accompanied by information flows to ensure an efficient execution of the logistic chain process, users' visibility of ongoing shipments movements and process transparency.

Due to the growing number of shipments to be moved, today, the logistic chain faces many challenges. The shipments require more sophisticated physical tools to be moved. The shipments movements also generate tremendous information flows that require more elaborated logistic chain information systems to accompany the shipments movements and ensure a full traceability of these movements inside the logistic chain.

The traceability is defined by ISO [58] as "*the ability to trace the history, application, or location, including for a product the origin of materials and parts, the processing history, and the distribution and location of the product after delivery*".

Nowadays, the traceability is no longer a need it is a requirement in many domains, especially in food industry. Recently there were many scandals, such as Bovine Spongiform Encephalopathy (BSE) which mainly affected the United Kingdom between 1986 and the 2000s [42] and the horsemeat scandal in 2013 which affected many European countries [104]. These scandals prove the necessity of advanced traceability tools to guarantee the product traceability and quality for the customers and make easy the industrial product recall and authorities audit and investigations processes.

To achieve this traceability in the Business to Business (B2B) logistic chain, it is important

to collect data about the shipment transport and storage conditions, such as the temperature and the delivery dates agreed between the stakeholders. For this purpose, traceability systems are used to track the shipment movements and trace the collected traceability data.

In this thesis, we refer to the data collected during shipment transit as *traceability data*, the system in charge of collecting, saving and sharing those data as *traceability system* and the whole process of data collection and processing as the *traceability process*.

Many discoveries have marked the history of the logistic chains, such as the Barcode invention in 1952 [96] which was a game-changer in the warehouse and the inventory management. In the domain of transport, the invention of containers in 1956 [92] was a stepping-stone, especially for the maritime transport. The Radio Frequency IDentification (RFID) tags invention in 1973 [123] overwhelmed the tracking management in the whole logistic chain and many other domains.

The reliability of a traceability system depends on the quality of its input data and the data collection process (manual or automatic). Traditionally, the traceability data collection in the logistic chain is based on manual tools such as faxes, phone calls, and emails. The traceability data is collected in a centralized system on one of the stakeholders' sides. With the growing number of goods to be moved in the logistic chain physical flows, faxes, phone calls and email tools show rapidly their limits. The stakeholders could no longer rely only on these tools for data collection.

The introduction of new Information and Communication Technologies (ICT) in the traceability process is considered as a competitive advantage for companies [87]. Among these technologies, the Internet of Things (IoT) and the blockchain are considered as the most promising for the logistic domain according to [64] and [6]. As depicted in Figure 1.1, we can see the growing academic interest in IoT and blockchain for the logistics domain.

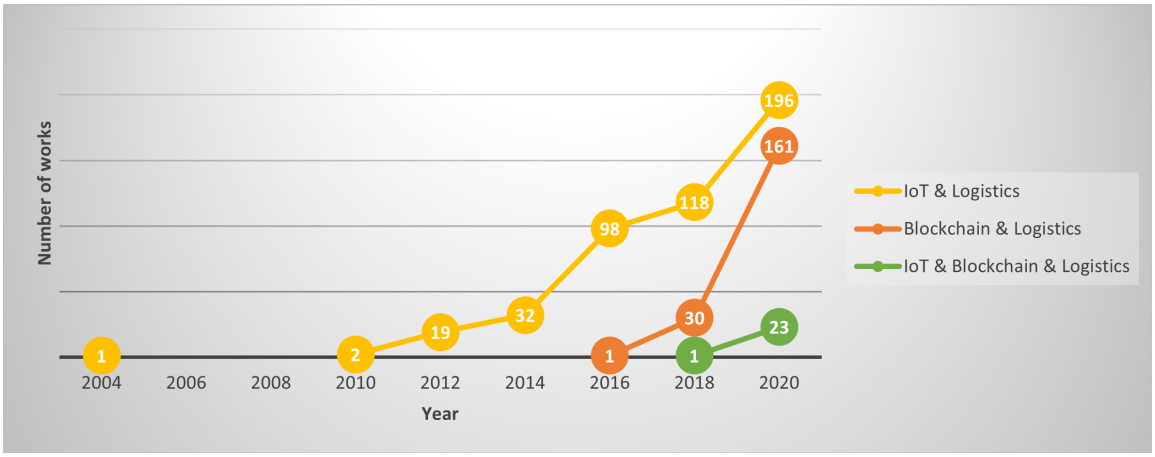


Figure 1.1: Research trends results using Web of Science[1] search tool with the keywords IoT, blockchain and logistics

Before going further in the discussion of new technologies integration in traceability sys-

[1] https://www.webofknowledge.com

tems, it is important to understand these systems. The following section presents the architecture of traditional traceability systems and their limitations.

1.2 Traditional Traceability Architectures

In logistic information systems, the three-tier architecture is widely used. As indicated by its name, this architecture is composed of three tiers [5], as depicted in Figure 1.2.

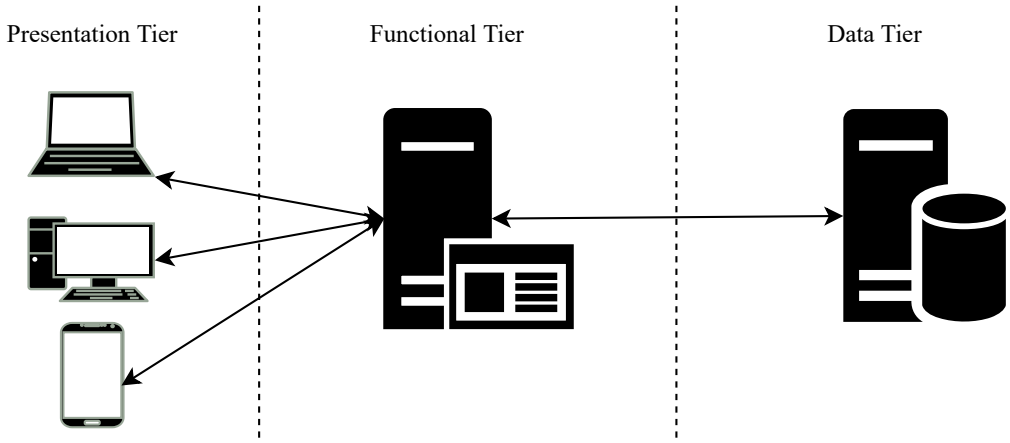


Figure 1.2: Three-Tier Architecture

- The presentation tier is responsible of the data presentation for the final user and the user interface;
- The functional tier is responsible of business rules, data processing and the communication with the presentation and data tiers;
- The data tier is responsible of data storage and management.

This thesis focus on the research and development issues related to the functional tier. The presentation and data tiers are out of the scope of this thesis.

As depicted in Figure 1.3, in the existing logistic traceability architectures, each stakeholder has its own traceability system. They use communication technologies such as Electronic Data Interchange (EDI) and web services to exchange the traceability data.

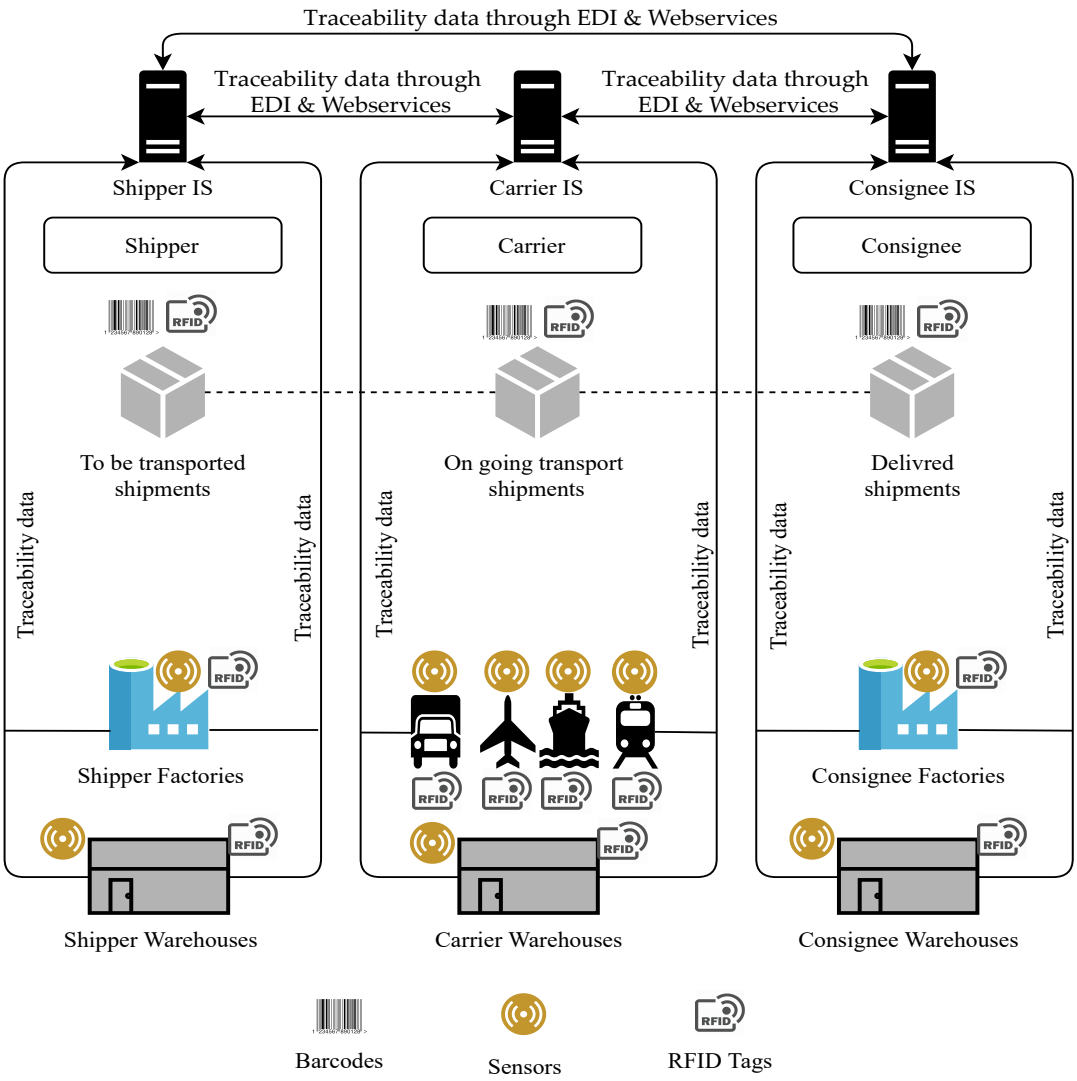


Figure 1.3: Traceability systems traditional architecture

The traceability data is classified in three types according to [126]: Master, Transactional and Condition Status. Master data are permanent, such as Origin, Destination, Weight etc. Transactional data are related to the shipment milestones progress, such as Departure, Arrival etc. Condition Status data are related to the shipment transport conditions, such as temperature, humidity etc.

As a responsible of the transport operation, the Carrier collects and centralizes the big part of the traceability data. From the waiting for Pickup Shipper notification until the final delivery to the Consignee.

The Shipper collects and centralizes data about the shipment preparation, waiting for pickup and pickup operations.

The Consignee collects and centralizes data about the delivery operation. He also identifies the status of the shipment at delivery and reports any non-compliance with the negotiated delivery terms, for example, a damaged shipment.

1.3 Limitations of Traditional Traceability Architectures

The stakeholders work together for the good functioning of the logistic chain. In this order, they need to share traceability data collected about the shipment progress in the whole logistic chain. However, the traditional traceability architectures present several limitations for the good accomplishment of data sharing.

1.3.1 Centralization

One of the big challenges of the traditional traceability architectures is: the centralization of data on one of the stakeholder sides. When the other stakeholders need to access the data, either they establish an EDI or webservice link with the IS of the data holder, or they ask him for an access to this data through any available medium (website for example).

There are many risks related to this data centralization issue. The data holder represents a single point of failure. If this system goes down, the traceability data that it holds is no longer available. Also, for the other stakeholders, there is no guarantee of the correct application of the agreed data collection and management rules.

1.3.2 Passive Data Collection Process

In traditional traceability architecture, the shipment data is collected using barcode scanning, RFID tags reading, Sensor values gathering or manually by the stakeholder workers.

Those passive data collection methods limit the data collection covering. For example, an RFID reader is needed to get data from the RFID tags. Consequently, the data collection possibilities are limited to the points of the logistic chain in which RFID readers are installed.

1.4 MyTower

The PhD took place in the company MyTower. MyTower is a company specialized in logistic software edition. Currently, the company main marketed solutions are shipper Transport Management System (TMS) and Global Trade Management (GTM) software. The TMS, as indicated by its name, helps the shippers in the management of all their transport processes, the search and booking of a transport, until delivery and transport freight management. It also integrates Key Performance Indicators (KPI) for transport process performance monitoring and improvement. The GTM is used for custom process management.

The TMS integrates a Track & Trace (TT) module connected to the carrier systems to collect shipment traceability data sent by these systems. The TMS is developed using the traditional architecture of traceability systems presented above. It is deployed in a Software as a Service

(SaaS) mode handled by MyTower or on promise on the shipper side. Consequently, it suffers from the limitations related to the centralization and the passive data collection issues.

The thesis objectives are to explore the questions related to the limitations of the existing TT module architecture. How decentralized, secured, and trusted architecture of the TT module can be implemented? How IoT sources can be integrated in this kind of architecture to improve data collection? How can the collected IoT data quality be ensured to ensure the stakeholders trust in the system and to facilitate the automation of the traceability process? How can an efficient and intelligent data processing be implemented to take advantage of all the collected data and to help the stakeholders in the traceability process improvement decisions?

To highlight real needs from a specific logistic chain context, we present in the next section, the thesis use case related to the medical equipment cold chain.

1.5 Medical equipment cold chain use case

The medical equipment cold chain is an emblematic example of the B2B logistic chain. For some of the transported equipment, temperature monitoring is required. Because of this specific constraint, it is handled by specific transport means. The shipper specifies the required transport conditions, such as temperature interval depending on the product type. Then, the carrier uses temperature-controlled trucks, containers, and/or boxes to ensure the product transport under the required temperature interval.

We have chosen this use case for two reasons. Firstly, the requirement of transport conditions monitoring, such as shipment transport temperature that needs an active data collection process. Secondly, we worked with a MyTower customer specialized in the production of medical equipment and we were able to discuss with this customer about their traceability needs for this specific cold chain context. We refer to this customer in the thesis as the shipper.

In this context, the shipper main traceability needs are tracking, tracing, secure sharing of the collected data and transparency in the whole traceability process. For the tracking, the shipper needs an active data collection process that gives the visibility on ongoing shipment transport operation. This process should not depend only on the data sent afterword by the carrier. The collected data needs to be traced, shared with the other stakeholders (the carrier and the consignee) transparently and in a secured manner. The quality of the collected data is not only a customer need but a requirement for the adoption, the trust, and the stakeholder adherence to the target traceability solution.

For the shipper, some of the equipment, such as perishable medical diagnostic kits used in blood tests, need to be transported under strict conditions with a temperature between a minimum of +2 and a maximum of +8 °Celsius. The non-compliance with this temperature interval may render the medical diagnostic kits unusable. The stakeholders should be notified of any temperature non-compliance.

For visibility and transparency purpose, the stakeholders need to securely share all the traceability data created manually or collected automatically by the IoT data sources. The stakeholders need also to be sure that the traceability data processing is conform to the agreed, and the traceability data availability should not depend on the availability of one of the stakeholder systems.

In the traditional traceability architectures presented in Section 1.2, the stakeholders use a milestones-based traceability system. The milestones are tracking points agreed with the carrier to monitor the shipment progress in the logistic chain. For example: Pickup, Departure, Arrival, Customs and Delivery as shown in Figure 1.4.

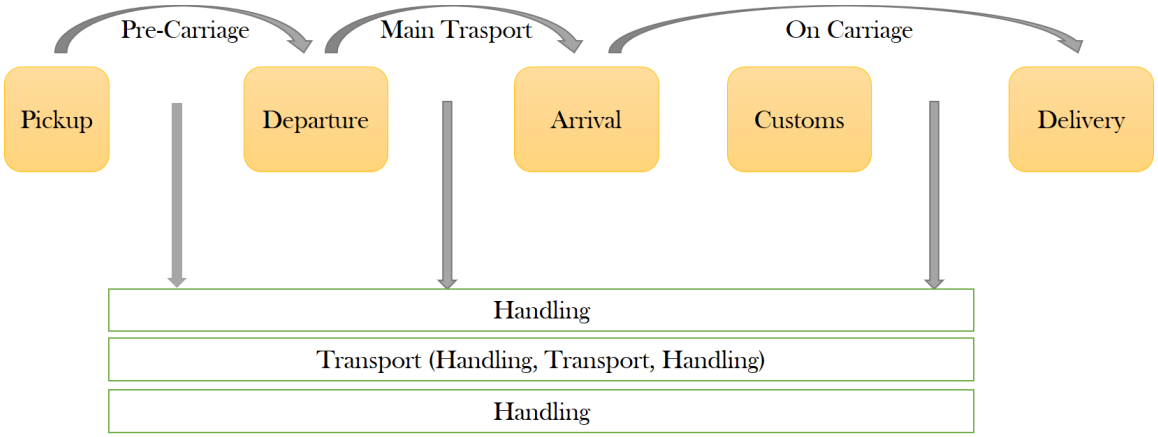


Figure 1.4: Examples of logistics transport milestones

The carrier should communicate about the state of each agreed milestone to the other stakeholders. The milestones are set manually in the system, and the degree of automation of data collect about these milestones depends on the degree of digitalization of the carrier.

It is worth noting that when the stakeholders trust and adhere to the data collected and sent by the IoT data sources, we will no longer need the milestones manual data collection. All the needed traceability data about the transport operation progress will be provided by the IoT data sources, and the progress milestones could be inferred automatically from the IoT data.

The shipper is responsible for the shipment creation in the traceability system, with all the data required by the carriers for the good execution of the transport operation, such as the origin, destination, transport temperature thresholds, milestones and IoT data reception interval.

In this scenario, we focus on the management of traceability related incidents. Those incidents are of three types depending on their detection and declaration process.

The first type are the incidents that should be detected automatically by the traceability system, based on the data sent by the carrier or the IoT data sources, such as the non-compliance with the negotiated transport temperature interval or the negotiated milestone date.

The second type are the incidents that could be declared manually by the stakeholders such as material damage.

The last type are the incidents related to the data quality. The target system should handle this kind of incident to incentivize the stakeholders to improve their source data quality according to a list of agreed data quality rules.

1.6 Research questions

From the above sections, we can highlight three issues in the current state of the art of traceability systems that we propose to address in this PhD. In the next subsections, we briefly present each issue and the associated proposition.

1.6.1 Decentralization

The centralization of the existing traceability systems on one stakeholder side (the shipper in the thesis use case) represents a risk for transparent data sharing among all transport operation stakeholders, data availability, and stakeholder agreement on the correct application of the agreed data processing rules. Additionally, the traceability data collection process relies essentially on the data provided by the carrier, which is a real issue for data availability.

The advent of new tools such as EDI and Webservices has accelerated the processes of data collection, data sharing capabilities among all the logistic chain stakeholders and improved the user visibility and the logistic chain traceability. However, the EDI and Webservices sharing methods have created a large traceability system that depends on traceability data shared among the stakeholders but without any guarantee that these data will be processed in the same manner on each stakeholder side, in compliance with the data processing rules agreed between the stakeholders. In case of discrepancy issue between the stakeholders on the shared traceability data, there are no clear and transparent processes to resolve it.

The decentralization of the traceability system architecture could help to overcome the above issues, by allowing transparent data sharing and data availability among all the stakeholders. However, the target decentralization technology should have the ability to ensure the data and processes security and the stakeholders agreement.

From that statement we identify the first research question handled by this PhD.

(RQ1) How can we ensure a **decentralized**, secured processing and a transparent sharing of the collected **traceability data** while integrating IoT data sources and guaranteeing the **stakeholders' agreement** on all the collected traceability data and its processing rules?

According to [6], the blockchain is a promising technology for the logistic domain. It helps in the development of distributed and secured architectures, and its consensus mechanism ensures stakeholders agreement on the processes and the collected data. In this thesis, we

propose to implement a blockchain based traceability architecture to answer the **RQ1**.

1.6.2 Data quality

In this work, we combine blockchain with IoT to provide the target architecture with field data collected automatically from the logistic chain. The combination of blockchain and IoT provides the stakeholders with transparency and visibility of the logistic chain operations.

Moreover, due to the limited resources of IoT objects and consequently their limited capacity to secure and verify the quality of their data, the IoT data should not be integrated directly in the target traceability architecture. This could lead to unsound decisions, such as "incident detection" based on unsound IoT data. For example, the shipment temperature collected during the transport operation should reflect the real transport temperature. Otherwise, it could be source of incident loss or false incident alerts. In both cases the stakeholder transport condition visibility and transparency requirements are not met, and the full traceability process is seriously affected.

Another issue in the existing traceability systems is that with the growing adoption of the IoT for the data collection, an automation of the data collection process and traceability decisions is possible. However, the quality of the collected IoT data is a big hindrance to the development of these automated systems. The data quality means that the following data facets are controlled and monitored by the stakeholders and ensured by the target traceability architecture. The traceability system should ensure that the collected data reflects the reality of the shipment transport conditions. Also, to ensure the full goods traceability, there should be no gap in the collected IoT data. In case of a traceability data provided by multiple sources, the target system should ensure the stakeholders agreement on the final data to be considered. Additionally, to improve the tracking process and consequently the stakeholders' visibility, the data should be compliant with the agreed data providing time interval. Moreover, to identify easily the elements affecting the collected IoT data quality, a fine control and monitoring of the IoT data quality is required by the stakeholders. Furthermore, as an important part of the traceability process, the IoT data quality control and monitoring processes should be shared securely and transparently among all the stakeholders to guarantee their agreement on the IoT data qualification rules and ensures the correct application of the agreed data quality measurement methods.

In this context, the second research question is:

(RQ2) How the **IoT data quality** could be ensured for traceability data in **decentralized architectures?** What are the methods to be used to measure and monitor the collected IoT data quality while ensuring the **stakeholders' agreement** on the correct application of these methods?

We propose in this thesis to integrate in the target traceability architecture an IoT data qualification process, to answer the **RQ2**. This proposition has many advantages: not only, it provides a quality degree to each shipment related IoT event and a performance measure of its

associated data source, but it also helps the users to choose the most trustworthy data source and facilitates the detection of damaged ones to repair or replace them.

1.6.3 Intelligent management of the collected data

The IoT integration in the target traceability architecture generates a huge volume of data related to the shipment progress in the logistic chain. This provides the stakeholders with a fine shipments' traceability.

In addition, the huge volume of traceability data generated by the IoT constitutes a gold mine for the traceability system decisions improvement. However, due to its volume, it could not be handled neither by human operators, nor by classic traceability systems.

To take advantage of all the collected data for the traceability system decisions improvement, the target traceability architecture requires an efficient and intelligent management of the huge volume of traceability data generated by the IoT. For example, the collected traceability data could be used to train Artificial Intelligence (AI) algorithms to predict traceability incidents and anticipate the to-do actions to avoid the predicted incidents. The daily goods movement and transportation costs forecasting are also among the possible AI use cases in the target traceability architecture.

Hence, our last research question targets this issue.

(RQ3) How can we ensure an **efficient** and **intelligent** processing of all the collected **traceability data**, while taking advantage of all this data for **traceability improvement decisions?**

To answer this last question, we propose in this thesis to integrate DL mechanism in the target traceability architectures.

1.7 Contributions

This thesis relies on the blockchain integration with the IoT and Deep Learning (DL) to answer the above-research questions and develop a decentralized, secured, transparent, trusted, automated and intelligent traceability system. In this context, the main contributions of this thesis are the following:

Blockchain-IoT based Architecture We propose a blockchain-IoT based architecture to handle the traceability data and ensure a secure and a transparent traceability process executed in the same manner among all the traceability process stakeholders. The proposed architecture introduces a generic traceability smart contract handling contractual milestones, incidents and IoT data and sources' management. It is evaluated using a permissioned blockchain, and the evaluation results show promising results for the proposed architecture.

IoT Data Qualification Process In the blockchain-IoT traceability architecture, we also propose to handle the collected IoT data quality to improve the resulting decisions and facilitate the system automation. To achieve the IoT data quality control and monitoring, we propose to handle four relevant IoT data quality facets for the logistic chain context. Firstly, the Accuracy to ensure that collected IoT data represents the reality of shipment conditions. Secondly, the Completeness to ensure that there is no gap in the collected IoT data. Thirdly, the Consistency to ensure the users agreement on the traceability data collected from multiples sources. Finally, the Currentness to ensure that the collected data is timely valid. For each IoT data quality facet, we propose in this thesis a corresponding measurement method to quantify, control and easily monitor the target architecture IoT data quality. For the fine control of the data quality, we propose also an IoT data quality model that ensure the stakeholders with a data quality visibility at each level of the manipulated objects. All the proposed IoT data quality facets and the quality model are implemented in the blockchain to ensure the stakeholders agreement on the correct application of these data qualification methods. The proposed IoT data qualification process is evaluated using a publicly available dataset, and the evaluation results show clearly how this module can be used to control and ensure the data quality, without a considerable impact on the time related to the IoT data integration in the blockchain.

Deep Learning for Intelligent Data Management Furthermore, to cope with the huge volume of data collected by the IoT, we propose to integrate AI techniques in the target architecture for efficient data processing, analyze and to push forward the development of proactive systems with predictive capabilities instead of traditional reactive traceability systems. As one of the most efficient advanced in the AI domain, we propose in this thesis to integrate a Deep Learning (DL) model in the target architecture. This will enhance the proposed traceability architecture with self-learning capabilities and will ensure an efficient and intelligent processing of all the available data without the need of any human intervention. In this context, we propose an adequate method to ensure the stakeholders' agreement on the DL model training, update, and output result processes. The DL-Blockchain integration proposition is evaluated on traceability incidents prediction on a real logistic chain traceability dataset.

1.8 The Manuscript Plan

The rest of this manuscript is structured as follows.

Chapter 2 provides the required background knowledge about the main technologies used in the thesis proposition, namely the blockchain and the IoT. For each one of these technologies, we present a detailed explanation of its underlying components, its application and contributions in the logistic domain and their related challenges.

Chapter 3 addresses the limitations of traceability architectures. After a literature review of blockchain, IoT and blockchain-IoT based traceability architectures, the chapter details the blockchain-IoT based architecture proposed in this thesis to answer the **RQ1** and ensure a decentralized and secure sharing of traceability data and transparent traceability process execution among the logistic chain stakeholders. The chapter also presents an implementation of the proposed architecture and an evaluation of this architecture based on the stakeholders' requirements.

Chapter 4 presents a study of the state-of-the-art of IoT data qualification and the thesis proposition to answer the **RQ2** using an IoT data qualification module for blockchain based logistic chain traceability architectures. In the proposed module, we address the quality facets related to the IoT data Accuracy, Completeness, Consistency and Currentness using a measurement methods implemented inside the traceability smart contract to ensure the stakeholders agreement on these data qualification methods.

Chapter 5 discusses the works related to the AI and blockchain integration and answers the **RQ3** by presenting the thesis proposition of DL integration inside the blockchain chain for efficient data processing and to equip the traceability blockchain based system with big data analyze and prediction capabilities. A use case of traceability data incident prediction is used to evaluate and show the relevancy of the chapter proposition.

Chapter 6 concludes this manuscript through a review of the contributions used to answer the thesis research questions and a presentation of some perspectives for future works.

Chapter 2

Background

For a better understanding of the thesis contributions, it is important to introduce some background elements related to the concepts and paradigms used in the thesis propositions.

In this chapter, Section 2.1 introduces the blockchain and its main underlying technologies such as asymmetric cryptography, hash function, consensus mechanism, distributed ledger and smart contracts. In Section 2.2, we present the IoT, its usage context and some of the issues related to its integration in logistic chain traceability blockchain-based architectures. Finally, Section 2.3 concludes this background chapter.

2.1 Blockchain

The blockchain paradigm has been introduced by Satoshi Nakamoto with Bitcoin Cryptocurrency in 2008 [99]. It uses a secured and shared tamper-proof ledger to store transactions.

Furthermore, it was firstly proposed to handle cryptocurrencies, but it rapidly shows its capacity to be used in many other domains, such as: citizenship services, health care, commercial, business, and industrial applications, according to [121] and [27]. The logistic chain with its multiple stakeholders was among the first use cases of the blockchain technology outside the cryptocurrency domain.

However, blockchain technology faces many challenges in its long road for industrial adoption. According to [6], among its main challenges, there are: the development of blockchain standards, overcome its current technical limitations, organization and culture related to this technology and its sharing paradigm.

In this thesis, we propose to explore the implementation of blockchain-based architecture, to ensure the stakeholders' agreement on traceability data, its processing rules, and the resulting decisions. We also study how the IoT could be integrated in the blockchain architecture to improve data collection. Furthermore, we investigate how blockchain based data quality

verification can be integrated in this architecture to ensure the quality of the collected IoT data and the transparency of the data quality rules. Finally, for efficient data processing, intelligent and automated improvement decisions, we also investigate how can the blockchain be used to ensure the stakeholders' agreement on DL models training, update and prediction processes.

To understand the blockchain technology and how the security and the immutability of its distributed ledger are achieved, we introduce below some of its underlying mechanisms, namely: the asymmetric cryptography, the hashing, and the consensus.

2.1.1 Asymmetric Cryptography

The asymmetric cryptography uses a combination of two different keys: a public and a private key, to ensure a secure communication between two parties. For these two keys, at least one should not be computable from the other, as stated by [116].

The public key is shared with all the communication parties and can be used to encrypt messages destinated to the public key owner. It is also used to verify that a message has been encrypted by the corresponding private key owner.

The private key is only known by its owner and is used to encrypt messages destinated to the other parties. The private key owner uses it also to decrypt messages received from the other parties and encrypted with its public key.

2.1.2 Cryptographic Hash Function

A cryptographic hash function is a function that takes an input i and gives an output o of fixed size, verifying that finding i from o is a hard problem and there is not another $i' \neq i$ that could generate the same output o, as stated by [36].

Hash functions are very useful in verifying data integrity, no matter what the data size is. The data is compressed in a fixed size output, and to verify that a data has not been altered, we just need to compare the known output with the hash result of the data to be verified. If the result is the same, this means that the data is the same as the origin, otherwise the data has been altered.

2.1.3 The Consensus Mechanism

The consensus mechanism is the core component of the blockchain. It defines the next block generation policy and ensures the blockchain participants agreement on blocks order.

The next block in the blockchain is generated by a leader elected among the blockchain nodes. There are two mechanisms for the leader election. The first one is the *lottery* known also as the *Nakamoto Consensus*. It is based on the Proof of Work (PoW). The node that resolves

the PoW puzzle is elected as a leader for the next block generation. The second mechanism is the *vote*. It uses many vote rounds to find a consensus among the blockchain nodes on the next leader election.

It is worth noting that the consensus mechanism is responsible of the next blockchain generation leader. However, there is another mechanism that is responsible of securing the chaining of blocks in the blockchain distributed ledger. This mechanism will be presented in the next section.

Among the main used consensus algorithms, there are:

- **Proof of Work (PoW)**: It is the most famous consensus mechanism due to its usage by Bitcoin and Ethereum. In PoW, to generate the next block, the blockchain participants try to resolve a puzzle by finding a nonce value verifying that when added into the block, the block hash result starts with a number of required zero bits. This activity is called the mining, and the miners with more CPU (Central Processing Unit) power are more likely to generate the next block due to the huge calculation made in search of the nonce value. Consequently, this consensus mechanism is energy inefficient [91]. However, this algorithm has proven its scalability in large scale blockchain networks such as Bitcoin and Ethereum.

- **Proof of Stake (PoS)**: To address the energy inefficiency problem of the PoW, the PoS has been introduced firstly by Peercoin [119]. It proposes to link the mining power to the percentage of coins owned by the miner. This reduces considerably the energy consumption and improves the security of the consensus mechanism.

- **Practical Byzantine Fault Tolerance (PBFT)** [29]: In this consensus mechanism, a node broadcasts a message and when it receives $2F + 1$ responses with the same value from the other nodes, the response value is considered valid. Such as F is the maximum number of tolerated byzantine nodes, which is a third of all the network nodes in the PBFT. Therefore, the agreement is reached more quickly than in the PoW, with less energy consumption. The PBFT is largely used by permissioned blockchains such as Hyperledger Fabric [14].

- **Replicated And Fault Tolerant (RAFT)** [101]: It is divided into two main steps. The first one is the leader election and the second one is the logs replication. In the leader election step, one node is selected to act as a leader and in case of crash a new leader is elected. In the log replication step, the leader accepts commands from clients, replicates its logs to the other network nodes and in case of conflict, the node content is overridden with the leader logs. This algorithm guarantees a high transaction throughput. However, it is not designed to handle byzantine faulty nodes. The RAFT is supported by some of permissioned blockchains such as Hyperledger Fabric [14].

- **Proof of Elapsed Time (PoET)**: It has been proposed by Intel[1] to address the byzantine faulty nodes problem in blockchain network. It relies on the use of a Trusted Execution

Environment (TEE) to select the miner that will generate the next block. A function is executed in the TEE generating random timer values for the participant nodes, and the node with the lowest timer value wins the next block generation. This reduces consequently the required amount of calculation for the next block generation, however it relies on the TEE such as the Intel Software Guard Extensions (SGX) implementation which is restrictive for the algorithm execution on network with different hardware architectures. This algorithm is used by Hyperledger Sawtooth [71], a permissioned blockchain implemented by Intel.

In the B2B logistic chain context, the PoW algorithm is not usable due to its huge resource consumption and the time required to validate a transaction. For example, in the Bitcoin blockchain network, a transaction could be considered as valid when 6 blocks have been generated after the transaction block. The Bitcoin blocks are generated every 10 minutes. This means that the user should wait 60 minutes (1 hour) after its transaction block generation, to consider this transaction as valid.

The PoS links the mining power to the number of coins which is not compatible with the traceability needs. Indeed, in the logistic chain traceability context, we do not need coins to handle the traceability data. Furthermore, all the stakeholders are at the same level in the traceability process. There is not a stakeholder that has more power in this process than the others.

Since the number of the B2B logistic chain stakeholders is limited and their identities are known, the risk of byzantine failure is limited, and consequently there is no need for PBFT like algorithms.

Furthermore, in the target traceability architecture, the IoT collected data will generate a huge number of transactions. Therefore, we focus in this thesis on the use of decentralized and secured consensus algorithms that will guarantee a high throughput for traceability related transactions, such as the RAFT and the PoET. It is worth noting that in the evaluations we did not have an SGX environment and we just used a simulated version of the PoET.

2.1.4 The Distributed Ledger

The Distributed Ledger is a register that holds all the blockchain transaction history, and each blockchain participant holds a copy of this ledger.

In the distributed ledger, transactions are stored in blocks and each block contains a reference to its parent block. This reference is a hash of the parent block, as depicted in Figure 2.1. This creates a list of chained blocks and gives to the technology its name blockchain. The first block in the blockchain is called the Genesis block. It contains the blockchain initial configuration.

[1] https://www.intel.com

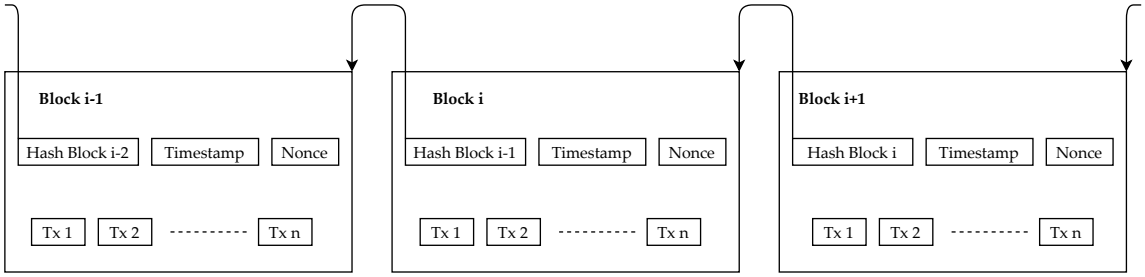


Figure 2.1: Blockchain

The above-described ledger is hard to alter by a malicious user, because any modification in a block content will result in a different hash for this block and this will break the local copy of the user, but not all the other participant copies and they will easily detect this data alteration attempt, which will be refused by the blockchain consensus mechanism.

Each blockchain participant has a pair of public and private keys. The public key is the user id in the blockchain. The private key is used to sign transactions and transactions' blocks submitted by the user, and this facilitates the identification of transactions and blocks origin. The other participants use the user public key to verify transactions and blocks origin.

2.1.5 Blockchain Implementations

The blockchain is implemented as permissionless such as Bitcoin [99] and permissioned such as Hyperledger Fabric [14]. On one hand, permissionless blockchains do not have any control on transactions read/write accesses. On the other hand, permissioned blockchains have integrated mechanisms to control users' transactions read/write accesses, and these accesses are granted to a limited and known number of users.

The blockchain could be deployed in public, consortium, or private environment [23].

- **Public blockchain:** read and write access in the blockchain are open to everyone. Any user in the world can join the network, read, and write transactions in the blockchain. It also guarantees the user's anonymity. This type of deployment is more adapted to Business to Customer (B2C) or Customer to Customer (C2C) contexts.
- **Consortium blockchain:** a list of pre-selected nodes control read and write access in the blockchain. It is useful in a context of multiples organizations with data secure share needs, as in the B2B logistic chain context.
- **Private blockchain:** only one organization controls all the rights in the blockchain

In the B2B logistic chain context, the blockchain implementation choice depends on the stakeholders requirements.

On one hand, public blockchains can be used by companies that don't have any problem

to work with anonymous suppliers or consumers or share their logistic chain on a publicly accessible ledger. This mode of work raises juridical and ethical issues for companies; however, it is a fully transparent work mode.

On the other hand, private blockchain is deployed only by one of the stakeholders but take advantage of blockchain secured and distributed architecture. In this context, the blockchain is controlled only by its deployer which is a real problem for transparency, trust, and the system availability for the other stakeholders.

Therefore, the consortium blockchain implementation is more adapted to the logistic chain B2B context in which all the stakeholders' identities are known, and the blockchain is controlled not only by one stakeholder entity, but by a consortium of stakeholders. The blockchain nodes are also deployed for each one of the stakeholders. This guarantees more transparency, stakeholders' trust, and system availability on each stakeholder side.

2.1.6 Smart Contracts

As stated by [120], smart contract designates the hard coding of all contract clauses in a hardware or software to be executed automatically in a secured and distributed environment. In the blockchain, the smart contract are self-executed programs that run on the top of blockchain.

They are used to develop business application on the top of the blockchain. Bitcoin proposed a script language for smart contract development; however, it was very limited. In the end of 2013, Ethereum [22] came with an integrated framework for smart contract development. Since, it has become a standard in blockchain implementations to integrate the support of smart contracts. The recent permissioned blockchains support smart contracts development in many of wide used languages such as Java in Hyperledger Fabric [14] and Python in Hyperledger Sawtooth [71].

In the logistic domain, the smart contracts could be used to write the business rules agreed between the logistic chain stakeholders. Hence, these rules will be shared and executed transparently among all the logistic chain stakeholders.

Like any other pieces of code, smart contracts are vulnerable to bugs and attacks [102]. On one hand, the bugs will just require the stakeholders' agreement to deploy a new version of the smart contract including the bugs corrections. On the other hand, in the B2B context, attacks risks are reduced due to the limited number of stakeholders and their identities known by each other. Each stakeholder is responsible of the transactions originated from his blockchain nodes.

Further details and discussions about smart contracts for logistic chain traceability will be presented in the literature study of Chapter 3.

2.1.7 Blockchain Pros And Cons

Pros

Among the main advantages of the blockchain, there are: the elimination of the need for a trusted third party to carry out transactions, the transparency and immutability through the shared registry and the fact that the transactions cannot be deleted or altered, also the blockchain data are of high quality because they are complete, consistent, dated and widely available. It is also worth noting that by using the shared ledger of transactions, the risks of data loss or its unavailability due to any failure, are largely reduced.

In the logistic domain, the blockchain helps in the secure and transparent data sharing among all the logistic chain stakeholders. It also brings the stakeholders trust in the logistic information system through the shared and tamper-proof transaction ledger. Hence, it ensures traceability, reduces the costs and risks, facilitates the management and improves the sustainability and the flexibility as stated by [76] and [17].

Cons

One of the biggest cons of the blockchain is the time it takes for a transaction to be validated. For example, it can take up to several hours on the Bitcoin platform, due to the size of the Bitcoin network. The use of permissioned blockchain such as Hyperledger Sawtooth for example, will limit this delay [25], because it allows the use of the PoET consensus algorithm, which is one of the fastest in terms of response time and the least greedy in terms of resources.

Additionally, the access to the logistic chain data must not be opened to everyone, this problem can also be resolved via the use of permissioned blockchains that offer a management of the blockchain access and the transactions execution rights.

Moreover, the consensus algorithms used in the blockchain, in particular the PoW, are very greedy in terms of calculations and therefore in energy consumption. Also, the redundancy of data and the redundancy of calculations required each time to decide whether a new block can be added to the blockchain, are greedy in energy consumption.

Finally, the blockchain is a complete paradigm shift, it means moving from a centralized to a decentralized network. This may lead to problems of adoption and integration of this technology by customers in existing ecosystems.

Although the blockchain technology remains at its infancy, many of its identified cons are related to the permissionless implementation. The development of permissioned blockchains opens new opportunities for the adoption of this technology in the logistic domain, especially in B2B contexts. The adoption of this technology accelerates in the domains in which the traceability and transparency are legal and final users' requirements as for example in Pharmaceutical [114, 19, 80] and Food [84, 28, 21] supply chains.

2.2 Internet of Things (IoT)

The term IoT has been used in first time by Kevin ASHTON, in 1999 [15]. There is no universal definition of this term, but the main idea of the IoT is that everyday objects can be equipped with the ability to sense the environment, communicating with other objects, and send the sensed data through the internet, as stated by [130]. Additionally, the IoT is a value chain of hardware, connectivity, software infrastructures in the cloud and applications and services that are implemented around this value chain.

In terms of connectivity, today, there are emerging networks for IoT called Low Power Wide Area Networks (LPWAN), such as: Sigfox, LoRa and Weightless [11]. These networks are designed specifically for the IoT as they are low energy protocols. Also, they allow two-way communication between objects and the outside world, using the LPWAN network and the network provider's cloud for internet connectivity.

The IoT technology provides the users with visibility and transparency, using near real time data collected from the field. It is used in many domains, such as Smart Infrastructure (homes, buildings, energy grids), Health Care, Supply Chain, Logistics and Societal Applications, according to [130]. Also, authors in [16] state that the IoT could help in facing many societal challenges such as: Health, Food security and Smart transport; through the monitoring of people health, smart farming and the management of logistic issues.

However, the adoption of the IoT technology faces many challenges related to resilience, security and trust [16]. Additionally, according to [97], the standardization of IoT communication protocols and the sensor energy supplying are among the main IoT technical challenges. The security of objects used to collect the IoT data is also a big concern, as stated by [51]. These objects have limited resources and could not ensure the security of their collected data.

In the logistic chain context, the Radio-Frequency Identification has been used to identify and track shipments. In the cold chain, data loggers have also been used to monitor the shipment transport conditions. In both cases, the data visibility was limited, and users could not get any data about the shipments when the transport operation is ongoing. Therefore, there is a real need to use new connected solutions to improve the traceability data collection mechanisms.

The integration of the IoT in the logistic chain improves the stakeholder's visibility, the tracking, and the traceability of the transported shipments through the entire logistic chain. This integration is about "sensing and sense making" as stated by [64]. Sensing through the whole logistic chain monitoring and sense making by taking advantage of the huge amount of data generated by the IoT for the logistic chain improvement decisions. Many recent works propose to use this technology especially for traceability improvement as in [7], [83] and [42].

Nevertheless, the integration of IoT collected data in blockchain based architecture faces many challenges related to the tremendous volume of data generated by the IoT, the blockchain

immutability nature and its automation capabilities using smart contracts.

Due to the data volume generated by the IoT and the blockchain immutability, the quality of the IoT data to be integrated in the blockchain need to be ensured. Furthermore, the data quality rules, as the others logistic traceability rules, need to be shared among the logistic chain stakeholders

Moreover, the blockchain automation capability using smart contract emphasizes the IoT data quality issue. Automatic decisions could be enshrined in the smart contract and should not be taken based on unsound data. Therefore, the IoT data qualification needs to be explored for blockchain based architectures.

In the logistic chain traceability context, the collected IoT data should reflect the real transport conditions status. It should also be received at the right time. Furthermore, all the agreed and consequently expected data should be received. Additionally, this data must be concordant with the other stakeholder's data if it concerns the same transport operation. Further discussions about the IoT data quality will be presented in Chapter 4.

2.3 Conclusion

In this chapter, we detailed the main components used in the thesis propositions, namely the blockchain and the IoT. For each component, we presented its underlying technologies and the main issues related to its introduction in the target logistic chain traceability architecture.

The next chapters present the contributions of this thesis. This starts by the blockchain-IoT based traceability logistic architecture in the following chapter.

Chapter 3

Blockchain-IoT Based Logistic Traceability Architecture

3.1 Introduction

The logistic chain traceability stakeholders need visibility and transparency about the shipments transport conditions and progress in the whole logistic chain. For this purpose, traceability systems are used to collect and handle shipments traceability data, from the pickup by the carrier until the delivery to the consignee.

Traceability systems evolved with the introduction of new information technologies. In the logistic chain, this digitization process [70] has changed the companies and the workers processes and life.

This chapter describes the introduction of blockchain and IoT in traceability architectures, to create a new generation of traceability systems, ensuring the secure and transparent sharing of traceability data among stakeholders. In this new generation of traceability systems, the main part of the traceability data is collected in near real time using IoT connected objects.

Many recent works have proposed to use this combination of blockchain and IoT to develop new traceability systems, such as those presented in Hinckeldeyn and Jochen [53], Wen et al. [128] and Hasan et al. [50]. However, in those works, there is no focus on the enterprise context. This context requires the usage of permissioned blockchain with an access reserved only to the stakeholders. For supply chain digitization companies such as MyTower, the proposed traceability smart contract needs to be generic. This will help in the reuse of this smart contract for any logistic chain context, without need of more development. The traceability smart contract genericity is not covered by the existing works in state of the art. Additionally, the milestones management needs to be integrated in the proposed traceability systems. The milestones are widely used in our days to handle the shipment tracking, and their integration facilitates new traceability systems adoption. Moreover, the IoT data sources, the transport conditions and

the traceability incidents generic management are important part of the traceability process that are not considered yet in the literature.

There are two main challenges for proposing blockchain-IoT based traceability architecture. Firstly, the selection of a blockchain implementation adapted to the enterprise context as discussed in Section 2.1.5 of Chapter 2. Finally, the implementation of traceability rules in a generic smart contract, especially the management of IoT data sources, transport conditions, milestones, and incident detection and management rules.

The proposed architecture evaluation is presented in Section 3.5 to prove the ability of the proposed architecture to be deployed in real life production scenarios.

The main contributions of this chapter are:

- The proposition and the implementation of a logistic traceability architecture based on blockchain and IoT and adapted to the enterprise context;
- A Generic Traceability Smart Contract handling IoT data sources, transport conditions, contractual milestones and incident detection and management.

The rest of this chapter is organized as follows: in Section 3.2 we present the chapter research questions. In Section 3.3 we study the literature of blockchain and IoT based traceability architectures. Section 3.4 presents the proposed blockchain-IoT architecture. The proposed architecture implementation and evaluation results are presented in Section 3.5. Finally, Section 3.6 concludes this chapter.

3.2 Research Questions

To overcome the traditional traceability systems limitations that have been presented in Chapter 1 and answer the thesis **RQ1**, this chapter addresses the following sub-research questions:

- **RQ1.1:** How can we conceive decentralized logistic traceability architectures? The decentralized architecture should guarantee a secure data sharing and stakeholders' agreement on the shared traceability data;
- **RQ1.2:** How IoT data sources could be introduced in this decentralized traceability architecture? The IoT provides traceability architecture with filed traceability data collected automatically in near real time;
- **RQ1.3:** How can we propose generic logistic traceability solution that could be easily reused in many logistic traceability context, with an integrated handling of transport conditions, milestones, and incident detection and management?

The work on these research questions helps in the development of new generation of logistic traceability decentralized and secured architectures, with automatic filed data collection

capabilities.

3.3 State-of-the-art of Blockchain and IoT for Logistic Traceability

This section presents and discusses the existing works in the state-of-the-art using IoT and blockchain to handle traceability data.

The logistic chain with its multiple stakeholders is a promising domain for blockchain technology application, as stated by [117], [62] and [106]. With the advent of this technology, and its secure sharing architecture, the actors of the logistic chain domain start to think about the use of this technology, to meet their traceability data secure sharing needs [6].

In the logistic chain the use of the blockchain allows the secure sharing of not only the traceability data but also all the rules agreed by the stakeholders for the data handling and ensure the correct application of those rules on each stakeholder's side using blockchain smart contract. Also, it helps in the meeting of the key supply chain management objectives such as cost, quality, speed, dependability, risk reduction, sustainability and flexibility as stated by [76]

The blockchain is used in some existing works to share manual collected traceability data. This is a first step of the integration of the blockchain technology in the logistic chain. In this thesis, it is proposed also to integrate in the blockchain filed traceability data collected automatically by IoT connected objects. The IoT connected objects provide the proposed traceability architecture with secured, accurate and fresh traceability data [64]. This is a big step forward in the logistic chain digitization.

The combination of blockchain and IoT to tackle the traceability problem in logistic chains is a recent research trend. In the literature, several solutions have been proposed using blockchain, IoT or their combination. These works are studied based on the following criteria:
(C1): Blockchain for traceability data: Does the work uses blockchain to handle traceability data
(C2): IoT for traceability data collection: Does the work uses IoT objects to collect some or all of the traceability data?;

In the following sections we discuss the works related to **C1**, **C2** and the combination of these two criteria (**C1&C2**).

3.3.1 Blockchain for Traceability Data (C1)

The blockchain usage in logistic chain provides the stakeholders with a distributed, secured, and trusted architecture for traceability data sharing.

The logistic chain with its multiple stakeholders was among the first use cases of the blockchain technology outside the cryptocurrency domain.

In the literature, many works propose to use blockchain to tackle the logistic chain traceability issue. In this subsection, the focus is on the works that use only the blockchain. The combination of blockchain and IoT is studied later in a dedicated subsection.

The selected works are studied based on the following sub criteria:
(C1.1): Generic traceability smart contract how the collected traceability data are handled in the blockchain? Are they just saved in the blockchain or there is a smart business process implemented on the top of the blockchain to handle those data? The implemented smart contract is it generic? This means that it can be used for other logistic traceability context. Are transport conditions handled by the smart contract?;
(C1.2): Contractual milestones: the contractual milestones are widely used in traditional traceability system. It is important to handle them to facilitate the adoption of the proposed traceability solution, before passing to the fully automated traceability system using the auto collected IoT data;
(C1.3): Incidents management: incidents are elements of the daily life in the logistic chain and the lack of secured and transparent process for their management affects seriously the data quality of traceability systems;
(C1.4): Blockchain implementation: the blockchain implementation is a determinant criterion in the target traceability architecture. We focus here on the blockchain implementation choice in the proposed works. Is it permissionless (e.g., Bitcoin and Ethereum) or permissioned (e.g., Hyperledger Fabric)? Due to the integrated access management support, the high transactions throughput, permissioned blockchains are more adapted to the B2B logistic chain use cases.

After the definition of the related works study criteria, we start the study and discussion of these works.

To achieve information sharing and synchronization in the supply chain, Chang et al. [31] proposed to re-engineer the traceability business process based on the blockchain. Using the proposed solution, they promise real time tracking, costs reduction, and payment lead time reduction using digital currency. In addition, they proposed to introduce control points for B2B supply chain scenarios by modifying the data structure without giving more details on how to do that. Authors in [31] proposition includes on-chain and off-chain data management process. On-chain data designate the data saved inside the blockchain and off-chain are the data saved outside the blockchain. This is an interesting proposition to alleviate the blockchain data charge. However, in the B2B logistic context, all the data agreed between

the stakeholders should be enshrined in the blockchain ledger for traceability and further audit purpose. Off-chain data technique could be used in future work for file storage, and the hash of sensitive files such as the Proof of Delivery (PoD) document should be stored inside the blockchain to ensure the document tamper-proof. Also, in B2B logistic context, the usage of permissioned blockchain with a limited numbers of stakeholders improves the blockchain transactions throughput and eliminates transactions fees used by permissionless blockchains. Finally, it is worth noting that the payment of logistic transactions between the stakeholders are out of the scope of this thesis. Consequently, we did not discuss the [31] payment process proposition.

Imeri and Khadraoui [57] proposed a blockchain based solution to handle the traceability issue in the Transport of Dangerous Goods (TDG) among all its stakeholders. The authors' proposition includes a smart contract for shipment emergency case management. However, as stated by the authors, it is used only for incident notifications. It does not cover the whole incident lifecycle management, from the incident creation till its closing by all the incident concerned stakeholders. Additionally, the authors' proposition includes a portal for data access management. Based on this access management, a stakeholder can decide to share a data with only one other specific stakeholder, and the other stakeholders will not see this shared data.

To tackle the food safety traceability problem, a combination of blockchain and EPCIS (Electronic Product Code Information Services), is proposed by Lin et al. [84]. This combination addresses the issue of trust transfer among the supply chain stakeholders. The authors also used an enterprise-level smart contract to handle sensitive data management issue. In addition, they proposed to address the blockchain data explosion problem, using a dynamic on-chain & off-chain data management. Finally, the authors implemented a prototype of their proposition using Ethereum. In the conclusion, authors in Lin et al. [84] state that their implementation could not handle huge data volume and suffers from throughput issue due the unoptimized consensus algorithm.

Westerkamp et al. [129] model the manufacturing process in a supply chain traceability system using Non-Fungible Tokens (NFT). The proposed mechanism aims to preserve the traceability of product transformations, by addressing the supply chain common issue of physical goods projection onto digital representation. For each type of goods, authors propose to implement a smart contract responsible of generating the unique NFT. Then, a transformation recipe is created with the tokens to be transformed as input and a new token as output corresponding to the new product. The authors proposition also handles certified goods that can be used equally in new product creation process. They also handle a list of roles involved in the product transformation such as resources suppliers, producers, and logistic and retail firms. Finally, the authors presented a prototypical implementation of the proposed traceability system using Ethereum smart contract. The shipment transport involves multiple stakeholders, implies responsibility transfer between them and there is no transformation of shipment during the whole transport operation. Consequently, the traceability solution proposed by [129]

could not be used to cover the shipment transport traceability process. However, it can be used in earlier logistic stages such as shipment preparation at the packaging step as stated by the authors.

Yong et al. [131] proposed a traceability solution for vaccine supply chain. Their proposed solution handles three main processes of vaccine supply chain. Firstly, the vaccine production in compliance with the Good Manufacturing Practice (GMP). Secondly, the vaccine lot release. Finally, the vaccine inoculation to a recipient. Also, the authors proposed an Ethereum based implementation of their solution including virtual coins for vaccine payment handling. Additionally, the proposition includes some intelligent recommendation modules for vaccine demand forecasting, vaccine production enterprise credibility evaluation and intelligent inoculation to assist the potential vaccine recipient. The proposed vaccine traceability solution detects vaccine expiration incidents. However, these incidents are related to the vaccine lifecycle management rather than the supply chain process.

Cui et al. [35] proposed a blockchain-based framework to provide traceability for electronic devices in the supply chain. The proposed framework can use Electronic Chip ID (ECID) or Physically Unclonable Functions (PUF) for unique devices' identification. Then, the generated unique ID is used to initialize and register the device in the blockchain. In addition, the framework ensures device ownership transfer for traceability purpose. Finally, the authors presented an implementation of their proposed framework using Hyperledger Fabric smart contract. However, the traceability framework proposed by the authors is not applicable for shipments which are not only electronic devices. This framework could be useful to track the IoT objects used to collect the traceability data.

For reducing overall cargo unit transport time and improving the logistic data sharing among the stakeholders, *The SmartLog* project proposed a blockchain based architecture. The target transport time for this blockchain pilot project is "in accordance with the EU's targets for road, rail, air and water transport networks in the Baltic/North Sea region" [95]. In first time, the project proposes to connect some port management systems for port operators' visibility improving, speed increasing and cost saving. According to the author in [95], *The SmartLog* blockchain will store transactions related to the intermodal containers movements and status. However, there is no details about the technical implementation of this project. Knowing that this project was *The SmartLog* project was intended to end in the summer of 2019, there is no information about its status in august 2021.

In 2018, UPS[7] applied for a patent to use blockchain in shipment tracking [8]. The patented system proposes to go beyond the tracking by handling all the shipment process, from the transport operation to the payment using cryptocurrencies, such as Bitcoin and Ethereum. Also, the system automatically determines a shipment transportation plan through the available transportation networks. However, there is no more information in the patent description about the technical implementation, the used smart contract, contractual milestones, or incidents management.

[7]United Parcel Service https://www.ups.com/

Table 3.1 summarizes the studied works in this subsection and how they meet the identified criteria.

References of research works	**Generic traceability smart contract (C1.1)**	**Contractual milestones (C1.2)**	**Incident management (C1.3)**	**Blockchain implementation (C1.4)**
Chang et al. [31]	N/A	B2B control points	N/A	N/A
Imeri and Khadraoui [57]	N/A	N/A	Smart contract for incident notification	N/A
Lin et al. [84], Westerkamp et al. [129] and Yong et al. [131]	N/A	N/A	N/A	Permissionless (Ethereum)
Cui et al. [35]	N/A	N/A	N/A	Permissioned (Hyperledger Fabric)
This work	Generic smart contract with generic transport conditions	Contractual milestones management	Incidents auto detection and qualification in the smart contract	Permissioned (Hyperledger Fabric)

Table 3.1: Blockchain related works comparison

All the above discussed works mention the usage of blockchain for traceability data management. However, there is no reference in these works to the IoT usage for traceability data collection. Consequently, their propositions are limited in terms of traceability data collection capabilities and could not provide the users with fresh data collected automatically from the logistic chain as we propose in this chapter.

3.3.2 IoT for Traceability Data Collection (C2)

The usage of IoT objects to collect shipment traceability data provides traceability system stakeholders with fresh filed data about shipment progress in the logistic chain. It also improves the performances in terms of delivery time, the available traceability data quality, reduce the risk of shipment loss and accelerating the logistic chain digital transformation process. Consequently, performance, visibility and transparency throughout the entire logistic chain are

improved.

This subsection focusses on the works using the IoT without blockchain. The works combining blockchain and IoT will be studied in the next subsection.

In the literature there are many works using the IoT for traceability purpose. As an example of these works, there is the *Luggage tracker* project [7] developed by Sigfox[5] and Louis Vuitton[6]. This project allows passenger to track their luggage in major world airports. It uses an IoT tag embedded in Louis Vitton luggage to collect data about the luggage position. The collected data is communicated to the cloud using Sigfox Network.

Li et al. [83] proposed a Real Time Monitoring Traceability System for a bacteria cold chain. Their solution is based on the use of the IoT to monitor bacteria storage and transportation progress in the cold chain. For this purpose, the authors designed and developed an electronic device composed of the following modules: sensing, information processing, GPRS wireless communication and power supply. Using TLINK Cloud Service Platform, the authors also developed a mobile and PC software to provide the users with a cold chain real-time environment monitoring.

Gialelis et al. [44] proposed a traceability platform based on the usage of low cost IoT node. On one hand, the proposed IoT node is composed of a microcontroller and radio module, a power supply module and an input/output module for sensors connecting. The evaluation of the proposed module shows a coverage distance of around 25 Km, a low energy consumption that could maintain the node endure for approximately 3.5 years, and a global cost of 40€ per IoT node. On the other hand, the traceability platform is composed of two modules. A first module for data aggregation and transmission and a second module for data processing. In addition to the product visibility and quality assurance, the traceability platform provides the users with many services such as an increased traceability.

However, the afore-discussed works relies on centralized solutions to handle the collected traceability data. Consequently, their traceability solutions suffer from availability and users trust issues. To overcome these issues, we propose in this thesis to combine the IoT technology with the blockchain to develop secured, decentralized, and trusted traceability solution.

3.3.3 Blockchain-IoT Traceability Architectures (C1 & C2)

The combination of blockchain and IoT is a recent research trend. In the logistic traceability domain, this combination provides the traceability stakeholders with fresh filed data collected automatically, processed, and shared securely among all these stakeholders.

In the literature, many works propose to work with this combination to deal with the problems of traceability data collection and secure sharing.

[5] https://www.sigfox.com/
[6] https://www.louisvuitton.com/

For an efficient track and trace of goods, easy customs formalities and counterfeit detection, a Smart Storage Containers prototype is proposed by Hinckeldeyn and Jochen [53]. Their proposition is based on the use of connected smart storage containers and blockchain smart contracts. The smart containers use weight sensors to detect the available product quantity. When this quantity flows below a defined threshold, the smart container orders automatically the needed product quantity. After the reception of the ordered quantity, the smart container automatically verifies the order weight and triggers the payment of the order using cryptocurrencies. The authors implemented a prototype of their proposition using Ethereum smart contract and a Raspberry Pi for the smart container control. Finally, the authors highlighted several limitations to their proposed prototype, such as the limited used hardware for the blockchain, the centralization of the IoT used protocol (MQTT) and blockchain scalability and smart contract verification. However, authors in Hinckeldeyn and Jochen [53] proposed to handle the storage, which is an important part of the logistic chain, but different from the transport operation in terms of process and requirements. The transport operation involves multiple stakeholders with their own connected IoT objects used to collect the traceability data. Additionally, the Ethereum blockchain implementation suffers from scalability limitations as stated by the authors. It also did not handle users' access rights. Consequently, it is not adapted to the B2B logistic chain context.

Wen et al. [128] proposed a privacy compliant traceability solution. It is based on the use of Attribute-Based Encryption (ABE). This technique allows the users with the right role attributes and that satisfies the access policy to decrypt the encrypted data. The authors implemented their proposition using Ethereum. However, the focus of authors in [128] was on data privacy rather than the traceability process. Their proposed approach could be integrated in the works that use permissionless blockchains to handle the shared data privacy. In the B2B logistic chain context, the permissionless blockchains are not suitable for the aforementioned reasons.

The shipment management system by Hasan et al. [50] is a combination of IoT-enabled shipment and Ethereum smart contracts. The IoT-enabled container monitors the shipments transport conditions using IoT sensors. The smart container handles the notifications conditions in case of non-compliance with the agreed transport conditions. The smart contract handles the transport operation creation and tracing. It also traces the non-compliance notifications. The authors presented an implementation of their proposed traceability solution using Raspberry Pi and Ethereum smart contracts. However, their proposition handles only some limited package status and transport conditions that could not cover all different contexts specific needs in terms of status and transport conditions. Additionally, the non-compliance notifications handling at the smart containers level presents a security and agreement issue on these notifications' conditions. In this chapter, we propose to handle these notifications at the smart contract level. Also, the smart contract creation for each shipment presents an efficiency issue for the blockchain smart contracts management. For the B2B logistic chain context, we propose to create one digital smart contract for each physical collaboration contract signed between a shipper, a carrier, and a consignee.

In the Agri-Food supply chain Caro et al. [25] proposed AgriBlockIoT, a solution for traceability management. This solution is a layered combination of IoT and blockchain to create a transparent, fault tolerant and controllable ledger. The authors proposition also explores the integration of edge devices such as gateways as blockchain nodes. In the evaluation, the authors compared Ethereum and Hyperledger Fabric and show that Hyperledger Fabric has a better latency, network, and CPU load performances in comparison to Ethereum. However, Caro et al. [25] results were preliminaries, and they did not show any advanced usage of smart contracts to control business logic such as incident related process as proposed in this chapter.

A reference implementation of a blockchain-based logistics monitoring system was designed and implemented by Helo and Hao [52] for parcels tracking. They proposed an architecture of four layers. A bottom layer composed of IoT objects and responsible of real-time data collection from Global Positioning System (GPS), RFID, sensors, and barcodes. A second layer responsible logistic operations data. A third layer handling the business logic such as logistic monitoring and access management. A top layer in charge of users' management. The authors select Ethereum to develop their proposed architecture. However, as stated before, this implementation is not suitable for the B2B logistic chain context, due to its transaction throughput, transaction gas fees and its permissionless nature.

For Soybean supply chain traceability, Salah et al. [111] proposed an architecture based on the usage of Ethereum smart contracts. They proposed to use IoT-enabled containers and packages to collect data about the shipment quality and conditions. The main objective of the authors' proposed approach is to guarantee the traceability of Soybean from the farmer to the final consumer. To handle the traceability related files (images, videos etc.), the authors proposed to use InterPlanetary File System (IPFS) which is a distributed file system and adapted to the combination with blockchain based system. We propose to use the same tools in future works to handle the B2B logistic chain traceability files. However, the authors architecture was proposed specifically to handle the Soybean supply chain process and could not be applied to handle other processes with different requirement and different supply chain processes. Additionally, there was no results about the evaluation of the proposed architecture to prove its capacity to be used in real world scenario.

In the domain of Food Supply Chain (FSC), Casino et al. [28] proposed to model the food traceability using a blockchain based approach. They used Ethereum for the implementation of their proposition. Additionally, their approach includes the usage of IPFS to store collected IoT data and alleviate the blockchain data charge for performance improvement. They also proposed to use smart contracts only for the FSC downstream (wholesalers, distributors, and retailers) and not for the FSC upstream (farmers, food producers and manufacturers). The IPFS usage is a good idea to implement the above discussed on-chain and off-chain approach. However, it should be used carefully to store only the data that are generally less disputed by the blockchain stakeholders such as files and metadata. In contrast, all the data agreed between the stakeholders and that could be used to generate traceability incident, such as

transport conditions collected by the IoT, should be stored in the blockchain. Finally, the authors proposed to implement only the downstream process using smart contracts. However, the whole traceability process involves multiple stakeholders from the farmers to the retailers, and they need to ensure the correct application of the agreed traceability rules. Consequently, the whole traceability process should be implemented using smart contracts for traceability rules trust and transparency, especially traceability incident handling related rules.

To track eggs from farm to consumer, Bumblauskas et al. [21] proposed an architecture based on Hyperledger Sawtooth blockchain implementation. Their architecture captures traceability and engagements data at every step of the supply chain using IoT. It also includes a proxy authentication server to handle stakeholders access rights levels and ensure that users access only to authorized data. However, in [21], the proposed traceability solution focus only on traceability data and not on the traceability process and its different stakeholders' interactions. This is a final consumer centered view and do not consider the supply chain stakeholders traceability process agreement and transparency needs. It is also a limited usage of the blockchain and especially its smart contract future. Beside the blockchain traceability feature, we propose in this chapter to take advantage of smart contracts to implement the B2B logistic chain traceability process.

IBM Corporation[7] and GTD Solution[8] proposed *TradeLens* which is a supply chain platform based on Hyperledger Fabric [3]. It provides users with real time access to shipping documents and tracking data including IoT sensors data such as temperature and container weight. According to [3] report, *TradeLens* also handles shipping milestones and exceptions. However, there are no details about how this handling is accomplished.

Table 3.2 summarizes the studied works in this subsection and how they meet the identified criteria.

[7] https://www.ibm.com/
[8] https://www.gtdsolution.com/

References of research works	**Generic traceability smart contract (C1.1)**	**Contractual milestones (C1.2)**	**Incident management (C1.3)**	**Blockchain implementation (C1.4)**
Hinckeldeyn and Jochen [53]	Generic smart contract for smart storage	N/A	N/A	Permissionless (Ethereum)
Wen et al. [128]	Generic smart contract (without transport conditions)	N/A	N/A	Permissionless (Ethereum)
Hasan et al. [50]	Smart contract with hard coded shipment status and transport conditions	N/A	N/A	Permissionless (Ethereum)
Caro et al. [25]	N/A	N/A	Recording detected anomalies	Permissionless and Permissioned (Ethereum and Hyperledger Sawtooth)
Helo and Hao [52], Salah et al. [111] and Casino et al. [28]	N/A	N/A	N/A	Permissionless (Ethereum)
Bumblauskas et al. [21]	N/A	N/A	N/A	Permissioned (Hyperledger Sawtooth)
This work	Generic smart contract with generic transport conditions	Contractual milestones management	Incidents auto detection and qualification in the smart contract	Permissioned (Hyperledger Fabric)

Table 3.2: Blockchain and IoT related works comparison

In brief, in the above studied works, there was no work which meet all the related works study criteria. Consequently, we decided to implement in this chapter a traceability architecture with a generic smart contract (C1.1), handling contractual milestones (C1.2), with incident management (C1.3) and based on a permissioned blockchain (C1.4) for B2B logistic chain context.

3.4 The Proposed Blockchain-IoT Architecture for Logistic Traceability

The main purpose of the proposed architecture is to provide the stakeholders with a fully automated traceability system. For this, it is needed to increase the user trust in the collected traceability data.

In order to meet the above requirements, the target architecture depicted in Figure 3.1 is proposed. This architecture allows the secure sharing and handling of traceability data, through the introduction of two components: the blockchain and the IoT data sources. These components are introduced on the top of the traditional architecture presented in Chapter 1 (Figure 1.3). In the target architecture, the blockchain component replaces the EDI and web-services exchanges used in traditional traceability architectures.

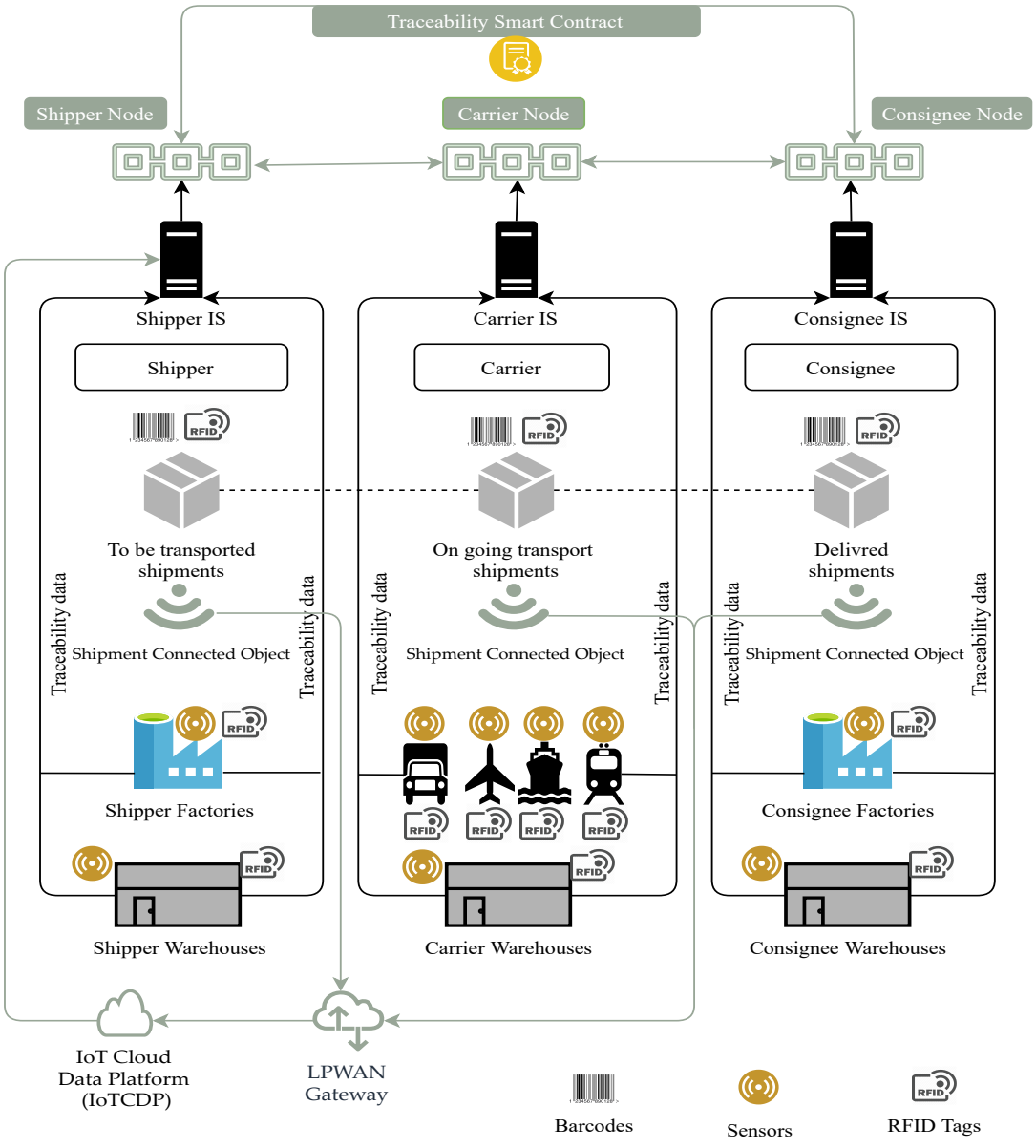


Figure 3.1: Traceability systems target architecture

To ensure the collection, transmission, and consolidation of IoT data in the target architecture, we propose a three layers model.

A first layer for perception allowing to collect shipment transport conditions data, such as: a GPS position to locate a parcel or a container, a temperature and/or a humidity level to monitor a goods packaging state.

A second transmission layer based on LPWAN networks depending on the network choice and the network card integrated in the object. This layer ensures the transmission of the collected data through the object network and the provider's cloud network.

A third application layer, which is the heart of this architecture. This layer is based on blockchain and allowing to clean, consolidate, analyze, and present the collected traceability data to the stakeholders.

The two main components of the proposed traceability architecture are presented in the following sections.

3.4.1 IoT Data Sources

In the proposed traceability architecture, the IoT data could be received from many IoT data sources. Each stakeholder could assign one or many of its IoT data sources to a shipment in which it has a stakeholder role, at any time during the shipment progress in the logistic chain. The only condition to do so is that the IoT data source and the shipment have been already created in the smart contract.

The assignment of an IoT data source to a shipment is for a limited period. Every data source assigned to a shipment sends IoT data about the shipment transport conditions at a fixed time interval defined in the shipment smart contract instance. If a data related incident is detected by the smart contract, it is automatically affected to the IoT data source owner declared in the smart contract. The smart contract has a detailed description of the IoT data source specifications collected at the data source creation in the smart contract.

In the proposed architecture, the shipper has a principal IoT data source which is the shipment connected object accompanying the shipment. The role of this object is to collect data about the shipment transport conditions, throughout the transport operation. To send the collected data to the shipper IS (Information System), the connected object uses an LPWAN (Low Power Aera Network) network Gateway which transmits the received messages to the IoT Cloud Data Platform (IoTCDP) before their reception in the shipper IS.

The shipper IS sends to the shipper node the received messages including the connected object id of the messages. This connected object id is used by the smart contract to link the received IoT messages to the right shipment in the smart contract. In this context, the data is pushed by the IoT object. The pull/push of data from/to the connected object is out of the

scope of our work. The shipment connected object collects data about the shipment pickup, transport and delivery conditions.

Each stakeholder could declare other IoT data sources, such as IoT data sources related to factories, warehouses, transport vehicles etc. In general, every data source that can collect and send automatically measures about the shipments could be declared by the stakeholder as an IoT data source. Moreover, all the IoT data sources, except the shipment connected object, help to collect data about the shipment conditions in a specific segment of the transport operation. Only, the shipment connected object that accompanies the shipment continues to collect data about the shipment transport conditions during the whole transport operation.

3.4.2 Traceability Smart Contract

The logistic chain stakeholders need to share and ensure the agreement on traceability data sources management and handling rules, and the decisions taken automatically by the traceability system. According to [86], this data sharing fluidifies the data flow and makes easy the logistic chain management, through decision making process based on filed data collected in near real-time from the logistic chain. To meet these needs, the proposition is to implement a blockchain traceability smart contract.

The proposed traceability smart contract holds the shipment and IoT data sources management rules. This means that it allows the data sources and shipment creation, update, and the assignment of an IoT data sources to a shipment.

Regarding the traceability decisions, they are automatically taken by the smart contract based on the collected data. These decisions are related to the shipment incident detection in case of non-compliance of the received data with the agreed transport conditions.

Moreover, the proposed smart contract handles contractual logistic milestones created manually and agreed by the stakeholders, to monitor the shipment progress in the logistic chain. It is worth noting that these milestones are implemented in the smart contract only to facilitate the stakeholder's adoption of the proposed solution. The objective of the proposed architecture is that finally the stakeholders use only the data collected automatically by the IoT data sources instead of the manual contractual milestones, to automate the whole traceability process.

Transactions In The Proposed Blockchain Based Architecture

In the proposed traceability architecture, the blockchain is used to save all the transactions at every step of the shipment progress in the logistic chain. Each write or update action in the ledger is considered as a transaction that is enshrined in the distributed ledger. For example, the creation of and IoT data source or the reception of an IoT data event. Also, the shipment progress milestones are considered as transactions to be traced because they mark a shift in the shipment legal responsibility between the logistic chain stakeholders.

Table 3.3 depicts milestones' transactions examples and the stakeholders of each milestone, in a standard transport operation scenario.

Transaction	**The stakeholders**
Pickup	The shipper and the carrier
Delivery	The carrier and the consignee

Table 3.3: Traceability Transactions Examples

For each transaction, Table 3.4 depicts some examples of the data to be saved in the blockchain.

Data	**Type**
The transaction unique identifier	Alphanumeric
The transaction Date & hour	Timestamp
Shipment number	Alphanumeric
Stakeholders' IDs	Alphanumeric list

Table 3.4: Traceability Transactions Data

The Proposed Traceability Smart Contract Class Diagram

To handle the shipment traceability in the logistic chain, we propose the class diagram depicted in Figure 3.2. It is worth noting that for some classes, such as the Shipment, the properties shown in the diagram are only a subset of all the class properties.

The **Shipment** represents the object entrusted by the shipper to the carrier in order to be delivered to the consignee. It is the main entity of the class diagram. The stakeholders involved in the shipment transport operation are saved in its stakeholders list.

The **Assignment** is used to handle the IoTDataEvent received during the relationship period between a Shipment and an IoTDataSource. This period runs from the timestamp sartAssiTime to the timestamp endAssiTime. In contrast to the Shipment-IoTDataSource relationship created manually by the users, the end of this relationship is set automatically by the traceability smart contract, when the delivery to the consignee is confirmed.

The **ShipmentMilestone** marks the shipment progress in the logistic chain. It has a list of stakeholders involved directly in the accomplishment of the milestone and a negotiated Date agreed between these stakeholders. The non-compliance of the milestone $actualDate$ with its $negotiatedDate$ results in a shipment incident.

The **ShipmentCondition** depicts the agreed transport conditions. It is worth noting that in this thesis we focus on the monitoring of bounded measures with a min-max interval requirement, such as min-max temperature, which is the main measure used in the thesis use

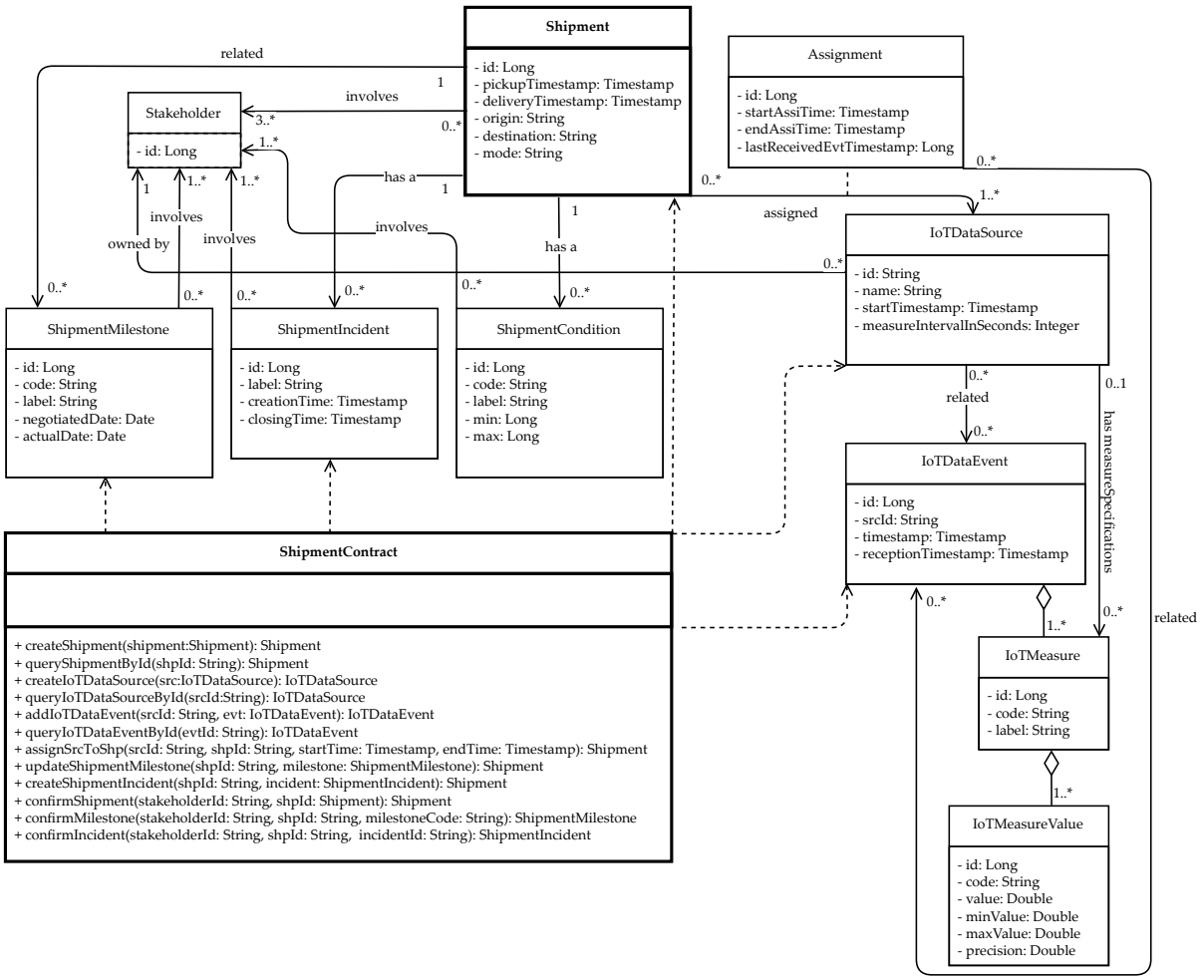


Figure 3.2: Traceability Smart Contract Class Diagram

case.

The **ShipmentIncident** handles the shipment transport conditions related incidents. These incidents are of two types. The first type are incidents created automatically when a non-compliance with the milestone agreed date is detected. The second type are incidents related to the non-compliance with the agreed transport conditions, such as transport temperature.

The **ShipmentContract** class contains the implementation of all the traceability smart contract methods. This class depends on the classes linked to it through dotted lines, as depicted in Figure 3.2. It uses those classes as input in its methods.

After this description of the proposed smart contract classes diagram, we describe in the next section the main methods of this smart contract.

The Proposed Traceability Smart Contract Main Methods

The *ShipmentContract* class main methods are: *createShipment*, *createIoTDataSource*, *assignIoT-Datasource*, *addIoTEvent* and *updateMilestone*.

The *createShipment* method is called by the shipment initiator to create a new shipment. It takes as argument a description of the shipment to be created with all its related elements: description, milestones, transport conditions and stakeholders.

The *createIoTDataSource* as indicated by its name, this method is used by the stakeholders to create IoT data sources. It takes as arguments a detailed description of the data sources and all its related measures specifications.

The *assignSrcToShp* method is used to create the temporal relationship between the data sources and shipments. This relationship is for the period specified by its start and end timestamp. It is at the maximum for the whole shipment progress time in the logistic chain, such as the relationship between the shipment connected object and the shipment.

The *addIoTDataEvent* method, depicted in Algorithm 1, is called by the stakeholder IS when a new IoT data event is received from an existing IoT data source. It takes as argument the blockchain transaction context ctx and the received $IoTDataEvent$ $event$. The method adds the event to the IoT data sources event list. After this operation, the event is sent to all the shipments related to the data source. Also, the event measures are verified to create eventually a shipment incident in case of non-compliance with the agreed transport conditions.

Algorithm 1: addIoTDataEvent

Input :

$srcId$; // The IoT data source id

$event$; // The IoT Data event

$reqComp \leftarrow$ `GetCompanyFromContext()` ; // Get the requestor company

$stateShps \leftarrow$ `GetShipmentBySrcId(`$srcId$`)` ; // Get the shipment assigned to the event source from the blockchain ledger

foreach *Shipment shp* ***in*** *stateShps* **do**

  if $reqComp \in shp.stakholders$ **then**

    `AddEventToShpCondEvt(`$shp.shipmentCondition$,evt`)`;

    $updatedShipments.add(shp)$;

    $isOutRangeEvent \leftarrow$ `IsEventOutOfShipmentConditionRange(`$shp.shipmentCondition$,evt`)`;

    if $isOutRangeEvent$ **then**

      `GenerateShipmentConditionNonComplianceIncident(`$shp.shipmentCondition$,evt`)`;

    end

  end

  else

    `ThrowUnauthorizedUpdateError()`;

  end

end

Output $updatedShipments$ **:**

The *updateMilestone* method depicted in Algorithm 2, is used for manual update of the shipment declared milestones by the stakeholders. As indicated previously, this manual method is used temporarily before the full automation of the traceability system using the received IoT data events.

Algorithm 2: updateMilestone

Input :

$shpID$; // The id of the shipment to be updated

$mlstToUpdate$; // The milestone to be updated

$reqComp \leftarrow$ `GetCompanyFromContext()` ; // Get the requestor company

$stateShp \leftarrow$ `GetShipmentById(`$shpID$`)` ; // Get the shipment from the blockchain ledger

$stateMlst \leftarrow$ `GetMilestoneByCode(`$stateShp$, $mlstToUpdate.code$`)`;

if $reqComp \in stateMlst.stakholders$ **then**

  $stateMlst.actualDate \leftarrow mlstToUpdate.actualDate$;

  if $mlstToUpdate.actualDate$ *after* $stateMlst.negotiatedDate$ **then**

    `GenerateMilestoneNonComplianceIncident(`$stateMlst$`)`;

  end

  $updatedMilestone \leftarrow stateMlst$;

end

else

  `ThrowUnauthorizedUpdateError()`;

end

Output $updatedMilestone$ **:**

The manual confirmation methods such as *confirmShipment*, *confirmMilestone* and *confirmIncident* are proposed in the smart contract to engage the stakeholder's responsibility in each one of these actions. However, it is worth noting that this confirmation aligns with the manual contractual milestones' usage. There is no confirmation in the automated scenario that use IoT data. In this scenario, the confirmation is done automatically based on the automatically collected IoT data.

The stakeholders' interaction with the smart contract

On each stakeholder side, there are two types of users, the administrators and the simple users. Administrators are in charge of the smart contract deployment and the blockchain access management.

Each user has its own pair of public and private keys. Their public key are used as their identifier in the blockchain network. The private keys are keeped secretly by the users and they use them to sign their transactions.

Traceability Smart Contract Conclusion

In brief, the above-described smart contract and its related classes do not hold any reference to a specific logistic chain context. Consequently, it meets the genericity criterion (C1.1) and could be used in any logistic chain context. Additionally, the contractual milestones management criteria (C1.2) is fulfilled through the handling of a generic milestone class and an $updateMilestone$ method in the smart contract. Finally, the incident management have been integrated in the IoT events integration and milestones update methods, to meet the incident management criterion (C1.3). The proposed traceability architecture evaluation is presented in the following section.

3.5 Evaluation of the Proposed Blockchain-IoT Traceability Architecture

This section presents an implementation of the proposed Blockchain-IoT traceability architecture. Some performance tests and results are also presented, to prove the ability of the proposed architecture to be deployed in real life production scenarios. Finally, the proposed architecture is evaluated based on performance test results.

3.5.1 Evaluation Environment

The evaluation tests were done on a Virtual Machine (VM) on which all the evaluation components have been deployed using Docker. The evaluation test machine has the following characteristics (Table 3.5):

Machine element	**Details**
Operating System (OS)	Ubuntu 18.04.4 desktop amd64
Central Processing Unit (CPU)	4 CPU Intel(R) Core™ i7-8565U
Random Access Memory (RAM)	8G
Virtual Disk	50G

Table 3.5: Test Machine Characteristics

3.5.2 Hyperledger Fabric Based Implementation

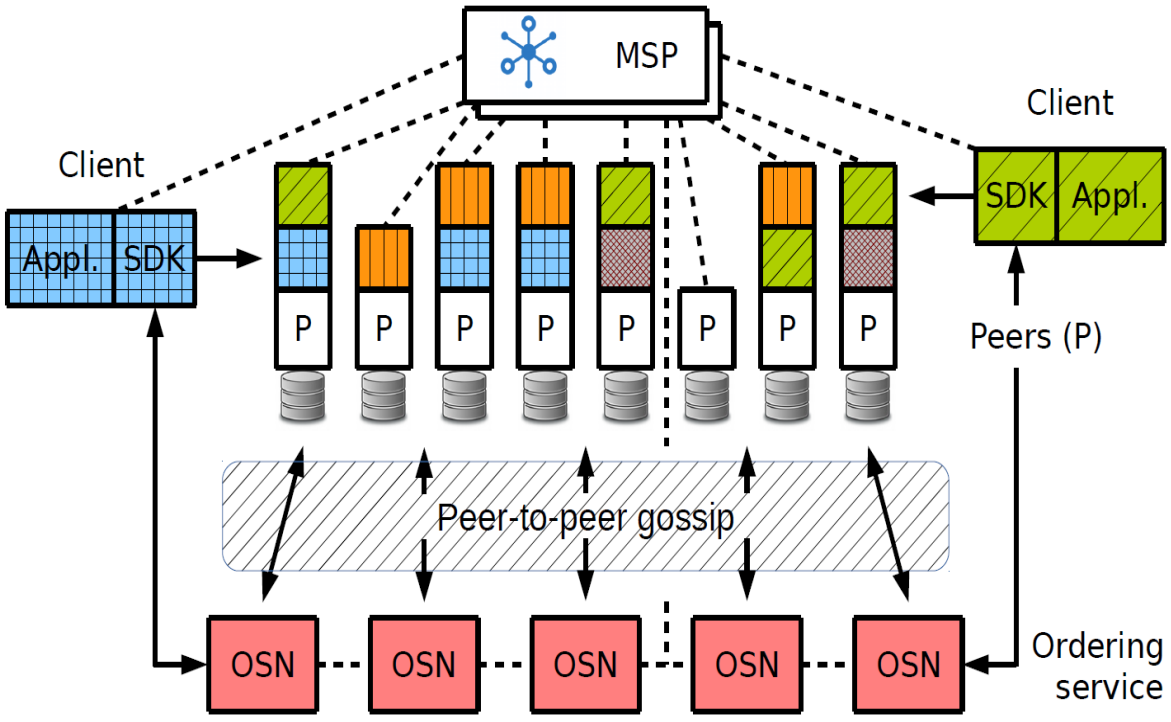
The proposed architecture has been implemented using Hyperledger Fabric or simply Fabric. Fabric is a permissioned blockchain implementation designed for enterprise purposes. It presents many advantages in comparison to other permissioned blockchain implementations, among them: a parametrized consensus protocol, a node architecture based on the notion of organization to establish a trust model more adapted to the enterprise context and the support

of Java, Javascript and Go languages for smart contracts writing [14]. Fabric addresses the limitations of order-execute architectures used by previous blockchain implementations.

In order-execute architecture, the transactions are ordered first using the consensus mechanism and then executed in the same order on all the network peers. This raises issues related to transaction sequential execution which limits the blockchain transactions throughput, the handling of non-deterministic transactions and smart contract execution confidentiality.

Fabric proposed an execute-order-validate architecture to overcome the above limitations. The Proposed architecture executes sequentially only related transactions, otherwise, transactions are executed in parallel, which significantly improves the transaction throughout. It also supports the development of smart contracts in generic languages such as Java, Javascript and Go. Moreover, Fabric proposes to install and execute smart contracts in an isolated environment only on concerned peer sides; however, the blockchain state is shared among all the blockchain peers. This ensures the smart contract execution confidentiality, while taking advantage of the shared and secured data ledger.

The Fabric main components are the Membership Service Provider (MSP) in charge of maintaining the node identities, the Ordering Service Nodes (OSN) responsible of channels' management and transactions ordering in these channels, and the Chaincodes which are the smart contracts in the Fabric terms. Figure 3.3 depicts an example of Fabric network, with a federated MSPs and Running multiple chaincodes.



Source: [14]

Figure 3.3: Fabric Network Example

Table 3.6 depicts the software versions that have been used for the development and the

deployment of the traceability smart contract.

Software	**Version**
Hyperledger Fabric Docker Images Tag	1.4.6
Hyperledger Fabric COUCH Docker Images Tag	0.4.20
Hyperledger Fabric Java Chaincode	1.4.3
Hyperledger Fabric Gateway Java	1.4.1
Docker	19.03.6
Java	1.8.0
Eclipse IDE	2019-12 (4.14.0)

Table 3.6: Test software versions

For this evaluation, as depicted in Figure 3.4, a Hyperledger Fabric architecture has been implemented, with three stakeholders interacting with the blockchain: a shipper, a carrier and a consignee.

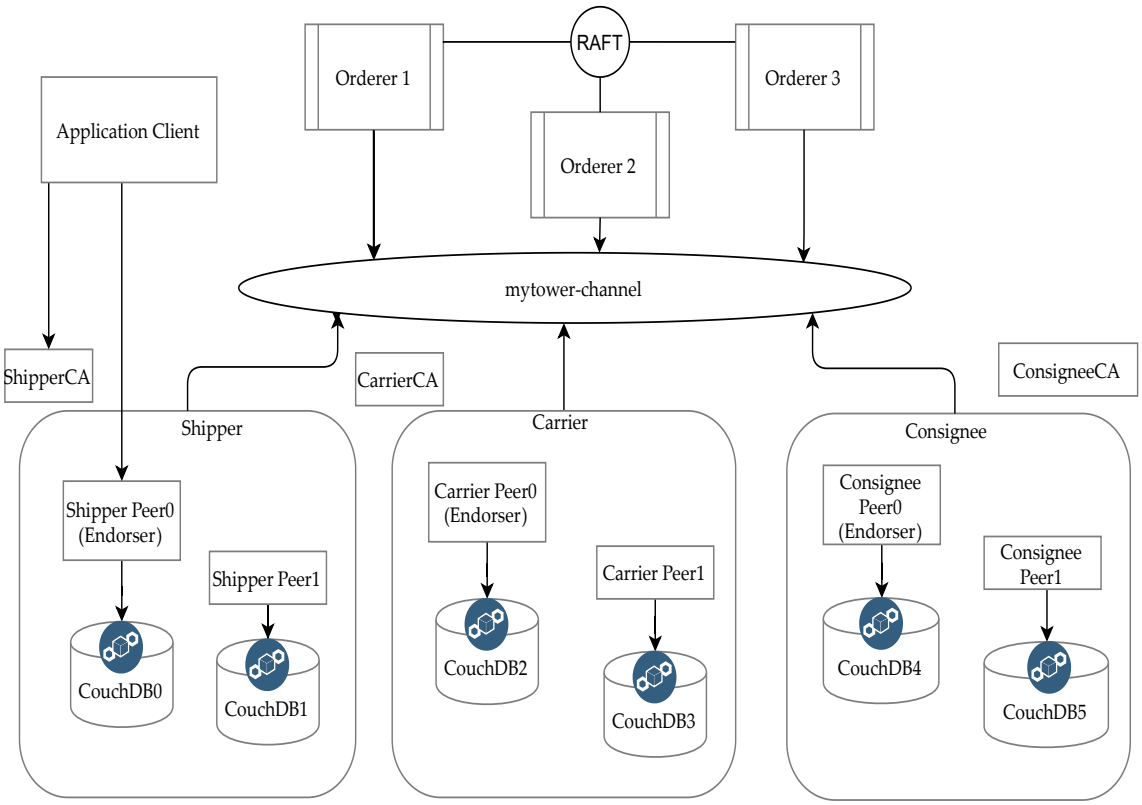


Figure 3.4: Hyperledger Fabric evaluation architecture

The stakeholders have been created as independent Hyperledger Fabric organizations. Each organization hosts the following components: *(1)* Certificate Authority, responsible of the organization user certificates management; *(2)* Two peers, with a local CouchDB database for each peer. One of the two peers is designated as the endorser peer, which is responsible of the correct execution of the smart contract on the organization side.

All the endorser peers are connected to a channel called « mytower-channel ». The transaction order is handled by a RAFT [101] cluster of three ordering nodes, as depicted in Figure 3.4.

3.5.3 Test and evaluation

We don't evaluate the Hyperledger Fabric performance itself, this subject has already been treated by [14] and [132]. The evaluation objective is rather to test the performance of the proposed architecture for its usability in logistic chain traceability system.

To let a sufficient time for the stakeholders blockchain nodes synchronization, the Hyperledger Fabric block creation timeout has been set to 1 second and the maximum number of transactions per block to 15. This means that after the reception of a new transaction, the system will trigger the block creation either after a waiting time of 1 second or after a total number of 15 new transactions is reached.

The test starts by enrolling a network admin on the shipper side. This enrollment consists of requesting the Shipper Certificate Authority (ShipperCA) for the admin identity creation in the network. The generated public key and private key for the admin are saved on the client test machine file system. Then, a second request is sent to the ShipperCA using the admin identity to register a new simple user that will be used in the test of the smart contract methods. As for the admin, the new user identity is saved on the Application Client test machine file system.

Performance Definition

The performance of a software solution is among the most important evaluation criteria. There are many aspects to be considered in the performance criterion, such as the per request response time, the number of requests per time unit and so on. In the B2B logistic chain context, the number of stakeholders is limited, and the shipment requests are initiated only by the shipper. The number of per time unit request is also limited. For example, in the thesis use case, the shipper has an average of 412 shipments per day, as will be presented in the evaluation dataset of Chapter 5. Consequently, we focus on the per request response time as our performance measure metric.

The architecture proposed in this chapter focuses on the functional tier of the three-tier architecture, as presented in Figure 1.2 of Chapter 1. Consequently, the response time to be measured and evaluated is for this functional tier. However, this response time includes the response time of the data tier.

According to the response time performance classification established by [108], we can classify the blockchain based architecture in the category "Complex or Ambiguous Search or Save Operations", due to the blockchain required calculations and replications operations. This category targets a response time of maximum 5 seconds.

Because of this thesis focus on functional tier in the three-tier architecture, we propose to define a maximum target response time of 3 seconds for this functional tier. This represents 60% of the global maximum target response time. Therefore, the remaining target maximum response time for the presentation tier is 2 seconds to do the basic operation of data presentation. This basic operation should take less than 2 seconds as stated by [108].

Based on our final users' requirements and the work of [108] and [135], we defined an acceptable per request time response of 3 seconds for the proposed architecture. This maximum response type is applicable to the create, read, and update requests.

Performance Evaluation Results

Due to a lack of testing data, we have used auto generated batches of data to test the proposed architecture. We have generated three batches of objects to be tested (e.g., Shipment and IoTDataEvent). A first batch of 500 objects, a second batch of 1000 objects, and a last batch of 2000 objects. This helps to monitor the response time evolution according to the batch number of objects evolution. The batches are launched sequentially in the same order for each round. A test round is a process in which we start the test environment, we launch all the test batches, we collect the test results, we stop all the docker containers and we clean the test environment. Then, we could start a new test round.

Tables 3.7, 3.8 and 3.9 summarize the test results for the main smart contract methods (i.e. *createShipment*, *createIoTDataSource*, *assignSrcToShp*, *addIoTEvent* and *updateMilestone*) for the first, second and last round respectively. The columns represent the number of objects in the test batch. For each batch, we present the Average (AVG) and the Standard Deviation (STD) response time in seconds. The result precision is two digits after the decimal point, rounding off from the third decimal place.

Method	**500**		**1000**		**2000**	
	AVG (s)	**STD (s)**	**AVG (s)**	**STD (s)**	**AVG (s)**	**STD (s)**
createShipment	1.24	0.03	1.25	0.02	1.25	0.05
createIoTDataSource	1.16	0.02	1.16	0.02	**1.15**	**0.01**
assignSrcToShp	1.18	0.02	1.18	0.03	1.18	0.02
addIoTEvent	1.19	0.02	1.2	0.05	1.19	0.02
updateMilestone	1.17	0.02	1.16	0.02	**1.3**	**0.05**

Table 3.7: The Smart Contract Test Results (Round 1)

Method	500		1000		2000	
	AVG (s)	**STD (s)**	**AVG (s)**	**STD (s)**	**AVG (s)**	**STD (s)**
createShipment	**1.47**	0.06	1.46	0.05	1.25	0.02
createIoTDataSource	1.29	0.05	1.26	0.04	**1.16**	**0.02**
assignSrcToShp	1.3	0.05	1.3	0.04	1.19	0.03
addIoTEvent	1.34	0.05	1.28	**0.08**	1.19	0.02
updateMilestone	1.32	0.04	1.18	0.03	1.17	0.02

Table 3.8: The Smart Contract Test Results (Round 2)

Method	500		1000		2000	
	AVG (s)	**STD (s)**	**AVG (s)**	**STD (s)**	**AVG (s)**	**STD (s)**
createShipment	1.26	0.04	1.26	**0.02**	**1.46**	0.05
createIoTDataSource	1.16	0.03	**1.16**	0.02	1.3	0.04
assignSrcToShp	1.17	0.02	1.19	0.02	1.3	0.05
addIoTEvent	1.19	0.02	1.29	**0.07**	1.19	0.04
updateMilestone	1.17	0.04	1.31	0.04	1.32	0.05

Table 3.9: The Smart Contract Test Results (Round 3)

The average (AVG) response time of the five tested methods did not exceed 1.47 seconds and the maximum standard deviation (STD) was around 0.08 seconds. These response time results meet largely the maximum response time of 3 seconds defined in the above section.

However, further tests are needed to confirm the performance of this architecture in a real distributed environment with large network constraints and more stakeholders, since those elements could impact the architecture performance.

3.5.4 Discussion

In this evaluation section, we have presented an implementation of the proposed IoT-Blockchain based traceability architecture using Hyperledger Fabric. The evaluation shows promising results for the architecture test and deployment in logistic chain traceability real life scenario. In comparison to the state-of-the-art works, this architecture has the advantages of generic smart contract handling contractual milestones and incident management using permissioned blockchain.

It is worth noting that the performance tests done in this evaluation do not include parallel blockchain calls. All the calls to the blockchain are done sequentially. This can explain the low transaction throughput of the proposed architecture (more than 1 second per operation) in comparison to some works in the state of the art as the 0.021 second required to set a value in a Hyperledger Sawtooth blockchain based architecture according to [25]. However, this low

throughput meets largely the performance requirements for the target use cases. It is also better than the throughput results of more than 16 seconds for Ethereum [25]. The proposed architecture response time performance could be improved by parallel calls tests and the work on fine tuning of Hyperledger Fabric settings such as the block generation time out and the maximum number of transactions per block.

3.6 Conclusion

In this chapter, a literature study of logistic chain traceability architectures has been presented. This literature study helped in the identification of centralization and passive data collection as the main limitations of the existing traceability architectures. A new Blockchain-IoT based architecture has been presented to address the identified limitations, with a generic traceability smart contract handling IoT data sources, transport conditions, milestones and incident management. The proposed architecture has been evaluated using the evaluation criteria identified from the stakeholders' requirements. This evaluation gives promising results for the deployment of the proposed architecture in real life logistic scenarios.

The blockchain based architectures is a new paradigm in the computer science domain and more specifically in logistic chain traceability systems. Its adoption will face stakeholders' reticence in first time due to its intrusive character. This could easily be overcome by explaining to the stakeholders the benefits of this technology, starting by the secure sharing and the tamper-proof of blockchain data and processes which facilitate the development of reliable traceability systems.

To implement a fully automated traceability systems and get stakeholders adherence, the collected IoT data quality needs to be ensured. This issue will be addressed in the next chapter.

Chapter 4

IoT Data Qualification In Blockchain Based Traceability Architectures

4.1 Introduction

The advent of blockchain technology and smart contracts helps in the development of new traceability systems. Such systems allow the logistic chain stakeholders to achieve the secure and transparent sharing of traceability data, using the blockchain secured and distributed ledger. In addition, smart contracts allow the stakeholders to share data handling and decision-making rules, to ensure that the same agreed rules are applied by all the stakeholders.

Increasingly, IoT devices are used to automatically collect field data. Those data are used both for traceability purpose and to take automatic decisions, such as the creation of *shipment* incidents, when one or more of the negotiated *shipment* transport conditions are not respected. As a result, the human intervention is limited in the process, as well as process error probability.

In traditional traceability architecture, the existing data sharing methods such as EDI and Webservices do not provide the stakeholders with a good data quality. Gharehgozli et al. [43] gave some examples of the identified data quality problems in intermodal transportation. Among them, duplicated, wrong and timely inaccurate data.

New traceability architectures have been proposed in the literature combining smart contracts and IoT for the development of trusted [24] and automated systems. However, the IoT data quality is a hindrance to the development and adoption of this new generation systems.

The data quality is an important topic in information systems [20]. It becomes critical

when the resulting decision of the information could engage the responsibility of one of the information system stakeholders. This is the case of the traceability system to be proposed.

The existing works in the literature propose to integrate the IoT data directly into the smart contract, as in [50], [21], [28] and [128]. This could lead to unsound decisions taken by the smart contract based on erroneous data collected and sent directly to the smart contract by the IoT data sources.

To overcome this issue, we propose to implement in the smart contract an IoT data qualification module to define and control the quality of the collected IoT data and ensuring the transparency and the stakeholders' agreement on the data qualification rules. The proposed quality module handles a selection of IoT data quality dimensions adapted to the logistic chain context, to measure different facets of the data quality. Also, the visibility of the data quality is improved with the management of different quality levels in the manipulated objects, and the introduction of data quality incidents in addition to the existing traceability incidents.

Moreover, the proposed IoT data quality module does not only increase the integrated data quality, but also the stakeholder's trust and adherence to the resulting automatic decisions.

The main contributions of this chapter are threefold:

- The literature review of IoT data qualification highlights that the data quality of a system is assessed by means of several dimensions. Considering the logistic chain properties, the first contribution is to identify the most relevant IoT data qualification dimensions and provide measurement methods for each of them.
- To help the stakeholders to get an end-to-end visibility of the data quality and to identify the quality issues causes, the second contribution aims at measuring the data quality at four levels: IoT data events, IoT data sources, shipments and IoT data sources-shipments associations.
- To ensure the stakeholders agreement on the traceability data, the data qualification rules, and the decisions taken based on the data, such as the creation of incidents, the third contribution consists in integrating the data qualification measurement methods in a traceability smart contract.

The rest of the chapter is organized as following. Section 4.2 highlights the research questions related to the IoT data qualification. In Section 4.3, we present the evaluation criteria used to validate that the proposition meet the stakeholders' requirements. The works related to the IoT data qualification are discussed in Section 4.4. Section 4.5 describe the proposed IoT data qualification approach. The proposition evaluation is presented in Section 4.6. Section 4.7 concludes the chapter.

4.2 IoT Data Qualification Research Questions

In this chapter, we address the thesis **RQ2** through six sub-research questions related to the IoT data quality in the logistic chain context.

RQ2.1: How accurate are the data? In other words, do the data reflect the reality of the *shipment* transport operation? Measuring data accuracy avoids the use of unreliable data.

RQ2.2: Are the data complete? Indeed, the existence of gaps in the collected data may affect the *shipment* traceability.

RQ2.3: Are the data consistent? The consistency issue arises when the collected data assigned to a *shipment* comes from several sources with possibly discrepancies leading to incidents. In this case, an agreement could be defined to tolerate a minimum deviation between the data, for example, a gap of 0.5 °C in the temperature may be considered as acceptable.

RQ2.4: Are the data timely valid? That is, are the data compliant with the receiving window agreed between the stakeholders? The non-respect of this interval may significantly affect the stakeholder's visibility and the required transparency of ongoing transport operations.

Each above question reflects a facet (dimension) of the quality process that this chapter addresses and thus the main contributions of this chapter are to propose quality measures for each dimension identified as relevant in the logistic chain context namely: accuracy, completeness, consistency and currentness.

In addition, to the above quality dimensions questions, there is a concern about quality granularity. **RQ2.5:** How can the system provide different levels of quality: data events, IoT data sources and per *shipment* performances? This high precision quality monitoring facilitates the identification at the right time of the data sources that need to be repaired or removed.

Finally, there is a question concerning transparency. **RQ2.6:** How can the data and the data quality measurement rules be shared securely among the stakeholders to ensure their agreement on the correct application of these rules? To address this issue, we propose to implement the above quality measures into a smart contract, to ensure the agreement of all the stakeholders on the correct application of the proposed quality measures.

4.3 IoT Data Qualification Evaluation Criteria

In this evaluation we worked on the existing traceability architecture presented and evaluated in Chapter 3. For data quality management and evaluation, we define in this chapter the thresholds and the indexes. The thresholds are the values that are defined by the stakeholders to express their data quality level requirements. The indexes are the output of the data quality measurements methods that will be defined and detailed in Section 4.5.

The thresholds and indexes values are expressed in values ranging from 0 to 1, 0 for no quality and 1 for the strict quality control. In Table 4.1, we establish a classification of these values to help in the presentation and the analysis of data quality evaluation results.

Data Quality Thresholds and Indexes Interval	**Label**	**Code**
$[0, 0.5)$	Poor quality	P
$[0.5, 0.7)$	Low quality	L
$[0.7, 0.9)$	Good quality	G
$[0.9, 1]$	High quality	H

Table 4.1: Data Quality Thresholds And Indexes Classification

4.4 State-of-the-art of IoT Data Qualification

Data quality is not a recent research topic. The first data quality studies concerned databases. Many data quality aspects have been considered such as the accuracy, consistency, and reliability to improve the quality of data inputs into databases and handle databases incompatibility and time critical delivery data [81].

With the advent of the IoT as new data sources, the existing data quality studied aspects needed to be extended to the specificities of those new data sources. The data collected from IoT data sources need to be controlled even more due to the limited capacity of these sources to ensure the security and the quality of their data. The "Never trust user input" should evolve to "Never trust things input", as stated by Karkouch et al. [69].

Moreover, the emergence of blockchain opens new opportunities for systems that involve multiple stakeholders. The logistic chain domain, which involves multiple stakeholders, provides relevant use cases for this technology [118], especially for traceability purpose [106]. The blockchain promotes the development of smart logistics [62], using smart contracts, as presented in Chapter 3.

Before providing a literature review, it is important first to define some terms used in the domain of data quality and their meaning in the logistic context.

4.4.1 Data Quality Definitions

Data quality dimensions are attributes representing a single aspect of the data quality, as stated by Richard Y. Wang [109]. In this chapter, we consider the following data quality dimensions: *accuracy*, *completeness*, *consistency* and *currentness*.

The *accuracy*, as stated by ISO-25012 [61], refers to: "the degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event

in a specific context of use". In the logistic traceability context, it is difficult to know if a received measurement reflects the real *shipment* situation, especially when the *shipment* transport operation is ongoing. However, we can define an accuracy measurement method based on the received measure and the measure source specifications.

The *completeness*, according to ISO-25012 [61], corresponds to "the degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use". In the logistic traceability context, the completeness depicts the fact that all the expected events have been received by a data source or a *shipment* according to the update interval agreed by all the stakeholders.

The *consistency*, according to ISO [61], refers to "The degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use". It is also referred to as concordance in some works [85]. In the logistic traceability context, the *consistency* dimension corresponds to the degree of coherence between IoT data events sent by different IoT data sources and related to the same *shipment*.

The *currentness* was defined by ISO [61] as: "The degree to which data has attributes that are of the right age in a specific context of use". It is also referred to as timeliness, currency, freshness, delay or contemporaneous, in some works [85, 80]. In the logistic traceability context, an event is considered of the right age when it is received at the expected time according to the update interval agreed by the stakeholders and defined in the smart contract.

4.4.2 Related Works Study Criteria

In this chapter, we study the works related to the IoT data quality issue according to three criteria:
(C1) Quality dimensions: what are the quality dimensions used and what are their corresponding measurement methods;
(C2) Quality levels: what levels of quality visibility are provided and what is the proposed quality model;
(C3) Blockchain smart contracts for data quality management: are blockchain smart contracts used to handle the agreed data quality rules?

Quality Dimensions (C1)

The IoT data quality issue has been addressed using data quality dimensions. For this purpose, the traditional data quality dimensions [109] have been used and adapted to the IoT context needs [85].

The definition of the IoT quality dimensions and their corresponding measurement methods facilitates their usage and application in the target IoT based systems. Due to the lack of works on IoT data using quality dimensions in the logistic chain context, we selected some

representative related works from other domains.

Many of the existing works show the interest of using those quality dimensions for IoT data quality handling. In each work, the authors selected the dimensions relevant to their domain and defined the corresponding measurement methods for the selected quality dimensions.

Li et al. [82] defined and measured the Currency, Availability, and Validity metrics in a pervasive environment (IoT context) and the problem of data expiration (data no longer usable). For the Currency measurement, the authors used a linear decline function $max[0, (1 - \frac{Age}{T^{exp}})]$, where Age is the difference between current and update time and T^{exp} is the valid lifetime provided by domain experts. To scale the data Currency, authors introduced the concept of data Volatility. It is the update probability between the last update and the current time. However, it is worth noting that, in the traceability context, the data do not expire. It is important to get all the data for traceability purpose even though the data received late will have a poor currentness quality index. The authors also proposed the Availability dimension which is "the percentage of the time that there is an unexpired data object provided by the source" [82]. This is a mixture of the Completeness and the Currentness dimensions. In this chapter, we choose to separate the two dimensions for fine monitoring of the data quality. Additionally, the authors data Validity dimension proposition corresponds to our Accuracy dimension in which we used only static rules for Accuracy measurement. Finally, the authors highlight the possible usage of dynamic rules for data validation which is a good idea and we need to study in future work how it can be applied in the logistic context.

Sicari et al. [115] proposed a quality-aware and secured architecture handling: Timeliness, Completeness, Accuracy and Precision. The author's Timeliness measurement proposition focus on the time interval between the data sampling and the data reception by their middleware. For Timeliness measurement, authors used the same Currentness measurement method discussed above in [82]. In their Completeness measurement method, the authors used a general method of the number of collected values in comparison to the number of expected values. We will use the same measurement method in our proposition. However, we will detail it for every quality measurement level. While for the Accuracy measurement, the authors proposed to measure each value Accuracy based on a reference value, in our proposition we propose to also integrate the measure dimensions in the reference values comparison to consider multi-dimensional measure values.

The *Valid.IoT* framework for determining the quality of heterogeneous information sources was proposed by Kuemper et al. [77], using the dimensions of Completeness, Timeliness, Plausibility, Artificiality and Concordance. For the Completeness, the authors used a standard measurement method based on the missed and expected values. To measure the Timeliness, the authors used a method based on a reward punishment algorithm. In our proposition, we use a close method for the Currentness based on a quality index updated with every new integrated IoT data event. The authors' Concordance measurement method consider only the sensors' locations. In the logistic chain context, the sensors Concordance relationship is defined not only by their assignment to the same shipment but also by the measurement time interval

defining the received events relationships. This means that the collected IoT data events are measuring the same transport conditions monitored at the same defined time interval. The Plausibility is an interesting quality metric that need to be handled in future work to consider the measurement context knowledge. However, in the logistic chain traceability context, we do not use interpolated data due to traceability requirements of real collected data. Consequently, the Artificiality quality metric is not applicable. Finally, authors in [77] proposed an interface to handle the quality metric requests at an atomic level of quality vector, a full quality vector and an extended quality document. This is more an aggregation of the collected quality metrics than the definition of quality measurement levels with real corresponding business entities as in our quality measurement levels proposition.

To ensure a real-time data allocation and data quality in multiple partitions collection and storage, Kolomvatsos [74] proposed a real time data pre-processing mechanism, using Fuzzy Logic and handling the Accuracy dimension. The author's proposition is developed for distributed data storage system that could include different data at every node of the distributed system. This is completely different from our blockchain based architecture which should maintain the same data at every node copy of the distributed ledger.

In the domain of Ambient Assisted Living (AAL) systems, Kara et al. [67] proposed a quality evaluation model. Their approach is based on the definition and execution of quality metrics and the use of fuzzy logic to evaluate the metrics and decide of the data quality level. There are many data quality dimensions in the authors proposition, such as the Accuracy, the Completeness, and the Precision. However, there were no details about the methods used to measure these dimensions.

In the same precedent domain, Erazo-Garzon et al. [39] defined, measured, and evaluated the quality of data collected from an intelligent pillbox, using seven data quality dimensions, among them the Accuracy, the Completeness and the Currentness. For the Accuracy and the Completeness measurement, the authors highlight the usage of correctly detected events and events used by the AAL system. However, there were no details in the paper about how these notions of correctly detected and used events are measured. To measure the Currentness, the authors include in their formula the maximum time in which the data becomes "old" or outdated. As discussed before, these notion of "old" or outdated do not apply in the logistic traceability context. In addition, authors in [39] defined fixed quality metrics weights which is an interesting idea to consider the metrics importance for the data consumer. In our data quality module, we propose to include a parameterized weight for each handled data quality metric.

All the above discussed works use some of or all our required IoT quality dimensions. However, their measurements methods do not meet our needs of dimensions definition and measurement at different levels: data event, data source and *shipment*.

Quality Levels (C2)

In the logistic chain context, the stakeholders need to be provided with a full quality visibility at different levels of the manipulated objects. This is the second study criterion (C2). It is helpful for the data quality management and simplifies the investigation in case of discrepancy between the stakeholders IoT data sources. Some works proposed data quality models to handle this issue.

Fagúndez et al. [41] work on a data quality model to assess sensors data quality in the health domain, using the dimensions of Accuracy, Completeness, Freshness, and Consistency. In their proposition, authors designed a conceptual schema for data window and its attached data quality information. In this schema the authors highlight the dimensions application at the stream window level. However, they did not provide details about the used measurement methods and the applicability of the used dimension at the data event, or the data source levels.

A generic data quality metamodel for data stream management was proposed by Karkouch et al. [68]. This generic model handles the data quality dimensions of Accuracy, Timeliness and Completeness. However, there were no details in their proposition about the measurement methods of these data quality dimensions, neither in the generic model nor in specific implementation for the evaluation. Also, their dimensions application was limited to the data item level (data event), and they do not cover the other levels of data quality measurement such as the data source for example. Additionally, in the evaluation of their work, the authors used only the Accuracy and Completeness dimensions. Besides, the above discussed elements about Karkouch et al. [68] work, their data model is a good start to implement our target IoT data quality model.

The above cited models do not meet the logistic chain context needs. On the one hand, the data sources in the logistic chain are reused and assigned to different *shipments* in different transport operations. On the other hand, to meet the criterion (C2) in our proposition, we provide the stakeholders with a full visibility of the data quality at different object levels, using an adequate quality model.

Blockchain Smart Contracts for Data Quality Management (C3)

Traceability data and its quality measurment methods need to be shared securely among the stakeholders to ensure their agreement on the data quality and the correct application of the agreed data quality measurement methods. This is the third study criterion (C3). The following representative works from the literature propose IoT-blockchain based architectures to handle this issue.

With the advent of smart devices, crowdsensing platforms are emerging to collects and share the smart devices sensors' data. There are many recent works, proposing to use the blockchain

in order to improve the quality of the collected data, such as the works of Gu et al. [47], Nguyen and Ali [100], Wei et al. [127], Cheng et al. [33], Huang et al. [55], Zou et al. [136] and Javaid et al. [65]. Their propositions are based essentially on users' reviews, reputation, and reward mechanisms to incentivize the users to improve the quality of their provided data. Those mechanisms are not applicable in the logistic chain context, in which the stakeholders are known and responsible of their provided data.

Casado-Vara et al. [26] proposed an IoT data quality framework based on the use of a blockchain, in the context of smart homes. Their solution uses a vote method based on the game theory to ensure the accuracy and the consistency of data collected among multiple sensors. Using cooperative coalition, "sensors are forced to cooperate to evaluate whether the temperature of the central sensor is correct in relation to their neighbourhood" [26]. To find a cooperative temperature, authors proposed to calculate an average temperature of all sensors, then the absolute value of the difference between each sensor temperature and the average temperature is calculated. This difference is used to establish a confidence interval and only sensors temperature that belong to this interval will be considered in the coalition voting process. However, their proposed approach does not involve multiple stakeholders, each having its own data sources with different technical characteristics. Consequently, it could not be used in the logistic chain context.

In the context of a fish farm, Hang et al. [48] proposed a blockchain based architecture to ensure agriculture data integrity. Their proposed fish farm architecture is composed of four components: fish farm, blockchain network, data storage, and end-user. They proposed a fish farm management smart contract deployed in the blockchain component. This smart contract includes an outlier filtering. However, there is no details in the article about the method used for outlier filtering. Outside the blockchain, the authors used an outlier filter to removes measurements beyond the expected values and predict future values. This outlier filter uses a Kalman filter algorithm.

Leal et al. [80] presented the European-Union-funded SPuMoNI project for end-to-end traceability and data integrity, in the domain of pharmaceutical manufacturing. This project addressed the problem of temporal and multi-source variability using probability distribution methods. Also, it used multi-agent system for data integrity checks close to the data source. In addition, it used Ethereum based architecture to ensure the end-to-end traceability of pharmaceutical products and processes. However, there were no details in the article about the methods used to quantify and measure the data quality in the SPuMoNI project.

It is worth noting that in the logistic chain context, we do not need to estimate sensor measurement data, so we should just report these data as they are sent by sensors. If some data are missed or out of the expected ranges, this results in a quality incident on which the involved stakeholders need to agree.

In our proposition, we implement the data quality measurement methods in a blockchain smart contract to ensure a secured sharing and agreement of all the stakeholders on the

correct application of the agreed measurement methods and the resulting data quality.

Table 4.2 summarizes the selected related works and how they meet the studied three criteria.

	Quality dimension (C1)	**Quality levels (C2)**	**Use of blockchain smart contracts for data quality management (C3)**
IoT			
Li et al. [82]	Currentness and others	Data	N/A
Sicari et al. [115]	Accuracy, Currentness, Completeness and others	Data and Stream window	N/A
Kuemper et al. [77]	Accuracy and Consistency	Data and Data Source	N/A
Kolomvatsos [74]	Accuracy	Data	N/A
Kara et al. [67]	Accuracy, Completeness and others	Data	N/A
Erazo-Garzon et al. [39]	Accuracy, Completeness, Consistency (lack of measurement method), Currentness and others.	Data and Data source	N/A
IoT Data Quality models			
Karkouch et al. [68]	Accuracy and Completeness (in the evaluation)	Data and Stream window	N/A
Fagúndez et al. [41]	Accuracy, Completeness, Freshness and Consistency	Data and Stream window	N/A
Blockchain and IoT Crowdsensing platforms			
Gu et al. [47], Nguyen and Ali [100], Wei et al. [127], Cheng et al. [33], Huang et al. [55], Zou et al. [136] and Javaid et al. [65]	N/A	N/A	Data quality ensured through reviews, reputations and rewards mechanisms implemented in blockchain smart contracts
Blockchain, IoT and data qualification			
Casado-Vara et al. [26]	Accuracy and Consistency	N/A	Accuracy qualified outside the blockchain smart contract and Consistency inside it
Hang et al. [48]	Accuracy (outliers filtering)	N/A	Outlier's filtering inside and outside the blockchain smart contract
Leal et al. [80]	Accuracy, Consistency (multi-source variability) and Currentness (Contemporaneous)	N/A	Data qualification outside and inside the blockchain smart contract
Our proposition	Accuracy, Completeness, Consistency and Currentness	Data, Data source, Shipment and Shipment data source relationship (equivalent to Stream window)	Data qualified using quality dimensions implemented in a Blockchain smart contract

Table 4.2: Related works comparison summary

4.4.3 Related Works Study Conclusion

In the above studied works related to the IoT data qualification, we did not find any work meeting all the fixed study criteria of quality dimension (C1), quality levels (C2) and blockchain smart contracts integration for data quality management (C3). The proposed approach to meet all these criteria is presented in the following section.

4.5 The Proposed IoT Data Qualification for Logistic Chain

In this thesis, the data qualification refers to the definition of data quality measurement methods and the application of these methods on every data received and handled by the smart contract.

We focus on the qualification of traceability IoT data. Because this data is automatically collected and used by the smart contract for incident detection, its qualification is essential for building reliable and automated traceability system.

Thanks to a data quality study adapted to the logistic chain domain, we have identified *(i)* relevant IoT data quality dimensions and their respective *(ii)* measurement methods.

The IoT data quality model purpose is to be implemented in the traceability smart contract, to assess the shipment data quality and consequently improve the incidents creation process. Among the quality models proposed in the state of the art, the one proposed by [68] was the closest to the logistic chain needs of IoT data quality dimensions and levels handling. Consequently, we decided to implement and extend this model for the logistic chain domain.

As depicted in Figure 4.1, we added into the $Shipment$ class some data quality attributes to measure and monitor the data quality at the shipment level using a specific list of data quality dimensions for the shipment. To help the users with an aggregated data quality vision, the $qualityConfidenceIndex$ gives an overview of the data quality at the shipment level based on its data history. For the same purpose at the dimensions level and for fine data quality monitoring, the $dimensionsQualityIndex$ gives the same data quality overview, but it is organized by data quality dimension. The incident threshold control gives the stakeholders the required tools to define the quality incident triggering threshold at the shipment level. This is exactly what the $globalDataQualityThreshold$ is used for. In the same manner, to define and control shipment quality incidents triggering at the data event level, we added the $dataQualityIndexThreshold$ attribute. For fine quality incident thresholds control, we provided the stakeholders with the $dataQualityIndexDimensionsThresholds$ in which they can define the quality incident triggering threshold by data quality dimension, based on the stakeholder's context requirements.

For capturing the data quality during the association of the $IoTDataSource$ and the $Shipment$, we enriched the $Assignment$ class which reflects their temporary relationship, with new data quality measurement attributes.

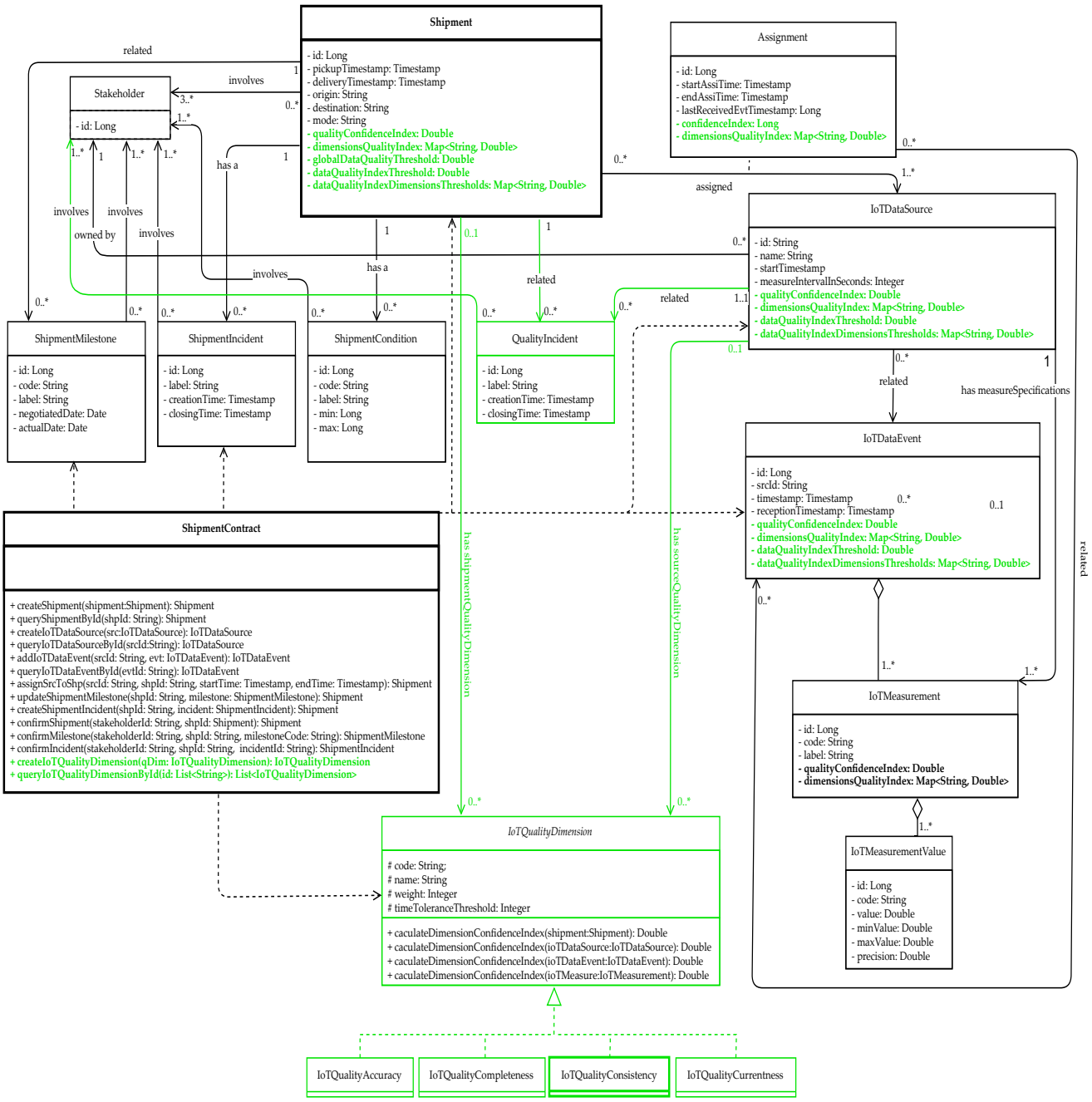


Figure 4.1: IoT Data Quality Class Diagram

Moreover, we highlighted in Figure 4.1 all the model entities and attributes added for the quality assessment purpose. The main class of this model is the *Shipment* which has its own *IoTQualityDimension* and its own *IoTDataSource* affected to it through the *Assignment* class. It is worth noting that the *IoTQualityDimension* has a weight attribute defining the importance of the dimension according to the stakeholders needs.

In the logistic chain context, we need to distinguish different application levels of each IoT quality dimension, for quality visibility at every object level. The quality index resulting from a quality dimension application is calculated differently for each dimension related class in the schema. In some cases (detailed in the next sections), a quality dimension is not defined for some entities of the schema. For example, the completeness dimension is not defined for the

entities $IoTDataEvent$ and $IoTMeasure$, it is used only for entities with an update time interval constraint such as $IoTDataSource$ and $Shipment$.

Furthermore, we introduce in this model a $qualityConfidenceIndex$, to provide users with an overview of the data quality for the main objects handled in the model, which are: the $Shipment$, the $IoTDataSource$, the $Assignment$ and the $IoTDataEvent$. We calculate this index for the $IoTDataSource$ and the $Assignment$ as an average of their events quality dimensions. For the $Shipment$ quality index calculation, we use the quality indexes of its related $Assignment$ objects. Regarding the $IoTDataEvent$, we use the average quality of its related $IoTMeasure$. The calculation of the quality index considers the quality dimensions' weights fixed by the users for $IoTDataSource$ and $Shipment$. The methods used to calculate the quality dimensions are detailed in the next sections.

The quality thresholds are set by the stakeholders to define the minimum accepted quality indexes. Values that do not respect this quality will be stored for traceability purpose but will not be used for dynamic incident detection purpose. To monitor the compliance of the received data according to both the quality threshold and the shipment transport conditions defined in the smart contract, we added into the model $QualityIncident$ class. It results from a non-compliance with the agreed quality thresholds. The $ShipmentIncident$ results from a non-compliance with the agreed transport conditions. If a newly received event generates a $DataQualityIncident$, this event is only saved by the smart contract, but not used to take any decision.

4.5.1 Accuracy

The accuracy measurement method is based on the data source specifications (precision, and minimum and maximum values). Using this method, we can ensure that the received measurement is a possible normal value that can be sent by the concerned data source; therefore, it could be used by the traceability smart contract, for example to create an incident if the received measurement is out of the ranges fixed by the shipper for this specific measurement. In case of inaccuracy, the received measurement could not be used in any business process, and it will just be saved in the blockchain for further audit purpose.

In the following subsection we detail the accuracy calculation method depending on the object level.

Accuracy levels

We identify five accuracy levels: the *measurement value* accuracy Acc_{MsrVal}, the *measure* accuracy Acc_{Msr}, the *event* accuracy Acc_{Evt}, the *IoT data source* accuracy Acc_{Src} and the *shipment* accuracy Acc_{Shp}.

The **measurement value accuracy** as indicated by its name is related to only one value

of the measurement. It is used to indicate if a value of the measurement is in the ranges of logic and acceptable values of this specific measurement value, based on the IoT data source specifications. For example, consider a measurement value m, with precision p and FTh_{min} and FTh_{max} are respectively the min and the max possible values given by the IoT data source fabricant.

We calculate the measurement value accuracy Acc_{MsrVal} using the following formula:

$$Acc_{MsrVal} = \begin{cases} 1 & \text{If } (m-p) \geq FTh_{min} \text{ and } (m+p) \leq FTh_{max} \\ \frac{m-FTh_{min}}{p} & \text{if } (m-p) < FTh_{min} \text{ and } m \geq FTh_{min} \\ \frac{FTh_{max}-m}{p} & \text{if } (m+p) > FTh_{max} \text{ and } m \leq FTh_{max} \\ 0 & otherwise \end{cases} \tag{4.1}$$

The IoT measurement is composed of n IoT measurement values, and consequently we calculate the **IoT measurement accuracy** Acc_{Msr} as an average of all its IoT measurement values' accuracies:

$$Acc_{Msr} = \frac{\sum_{i=1}^{n} Acc_{MsrVal_i}}{n} \tag{4.2}$$

The **IoT data event accuracy** Acc_{Evt} corresponds to an overview of the accuracies of all its related measurements. This is useful in the logistic chain context, where the event is considered as a coherent set of measurements. If this is not the case, the accuracy calculated at the measurement level can directly be used, and the event accuracy can be ignored. However, for an event with n related measurements, the event accuracy corresponds to the average of all the event related measurements' accuracies:

$$Acc_{Evt} = \frac{\sum_{i=1}^{n} Acc_{Msr_i}}{n} \tag{4.3}$$

The **IoT data source accuracy** Acc_{Src} gives an overview of all the data source related events accuracies, it is related to the history of events received from the IoT data source. In the logistic chain context, it is important to consider this history of events in the calculation of data source accuracy, because it indicates the reliability of the data source since it has been deployed and used in the traceability system. If the users are interested only in the data source measurement accuracies, the accuracy calculated at the measurement level could be reused at the data source level in order to give them a data source accuracy per measurement. The accuracy of an IoT data source corresponds to the accuracy average of all its related IoT data events:

$$Acc_{Src} = \frac{\sum_{i=1}^{n} Acc_{Evt_i}}{n} \tag{4.4}$$

Finally the **shipment level accuracy** emphasizes all the shipment related IoT data sources' accuracies for the specific time period in which a data source is assigned to a shipment. Every shipment is considered as an independent transport operation that should have its own accuracy value. For a shipment with n assignments to IoT data sources, the accuracy Acc_{Shp} corresponds to the average of all the shipment IoT data sources assignments. For each assignment accuracy Acc_{Assign_i}, the number of events $n_{EvtAssign}$ to be considered in the accuracy calculation, corresponds to the number of events sent by the data source for this specific shipment assignment relationship:

$$Acc_{Shp} = \frac{\sum_{i=1}^{n} Acc_{Assign_i}}{n} \quad \text{such as } Acc_{Assign_i} = \frac{\sum_{j=1}^{n_{EvtAssign}} Acc_{Evt_j}}{n_{EvtAssign}} \tag{4.5}$$

4.5.2 Completeness

The completeness measurement method calculates the gap in the data reception for a specific object. It concerns the levels of the data source, the assignment, and the shipment.

Completeness levels

At the IoT data source level, the completeness is calculated based on the source $startTimestamp$, the source measure interval I, the number of received IoT events n from the data sources and the reception timestamp of the last IoT data event $lastTimestamp$, related to the data source:

$$Com_{Src} = \begin{cases} 1 & \text{If } n \geq \frac{lastTimestamp - startTimestamp}{I} \\ \frac{n * I}{lastTimestamp - startTimestamp} & \text{otherwise} \end{cases} \tag{4.6}$$

The $Assignment$ completeness Com_{Assign} means that all the expected IoT data events of the assigned IoT data source Src, have been received by the shipment during the data source and shipment association time period enshrined in the smart contract. Consequently, for the shipment, the IoT data event frequency is at least one IoT data event per IoT update time interval I defined in the smart contract. The Com_{Assign} highlights for the stakeholders the capacity of each data source to send all the expected data during its association with a shipment. This helps the stakeholders to decide on the reusability of the data source for further shipments in case of a good Completeness value, otherwise, to take over the data source to identify the Completeness source problem. The Com_{Assign} evolves during the shipment and the data source association time period, and it is recalculated for every new IoT data event reception at the timestamp $evtTimestamp$, based on the current number of received IoT data events n, the shipment update interval I, the data source shipment assignment $startAssignTime$ and $endAssignTime$ timestamps.

$$Com_{Assign} = \begin{cases} 1 & \text{If } n \geq \frac{evtTimestamp - startAssignTime}{I} \\ & \text{and } evtTimestamp \in]startAssignTime, endAssignTime] \\ & \text{Or} \\ & n \geq \frac{endAssignTime - startAssignTime}{I} \\ & \text{and } evtTimestamp > endAssignTime \\ \frac{n*I}{endAssignTime - startAssignTime} & \text{If } evtTimestamp > endAssignTime \\ 0 & otherwise \end{cases} \tag{4.7}$$

At the shipment level, the completeness Com_{Shp} gives an idea of the Completeness trend of all the shipment related IoT data sources. It is calculated as a Com_{Assign} average of the n_{Assign} data sources assigned to the Shipment:

$$Com_{Shp} = \frac{\sum_{i=1}^{n_{Assign}} Com_{Assign_i}}{n_{Assign}} \tag{4.8}$$

Completeness incidents

The completeness problem reflects the missing IoT data events. Many reasons could be at the origin of missing IoT data events: network errors, synchronization problems or device malfunctions [90]. If it is not handled, missing data affects seriously the reliability of the data collected through the IoT data sources. We propose to generate a completeness incident, if the completeness index of the object fall below the completeness threshold fixed by the stakeholders.

4.5.3 Consistency

It is important to calculate the coherence degree between related different IoT data events and to alert the stakeholders in case of incoherence detection. The stakeholders should take a corrective action, such as identifying and removing failing data sources or adapting new threshold values etc.

The main IoT data source in this work is the shipment connected object. However, others IoT data sources could be added by any of the shipment transport stakeholders. When two or more IoT data sources assigned to the shipment monitor the same transport conditions, then we calculate the consistency of those data sources, by comparing their measurements. This comparison considers two tolerance thresholds: the time tolerance threshold Tt_{th} and the measurement tolerance threshold Mt_{th}. Those two thresholds should be defined at the shipment creation for every IoT data sources assigned to the shipment, of course through a mutual agreement between the stakeholders in charge of those IoT data sources.

Consistency levels

The consistency dimension concerns the levels of: *IoT data event* and *shipment*. When an IoT data event Evt_i is received from source Src_i at a timestamp Rt_i, and contains a list Msr_i of measurements, the smart contract check if there are other events related to the shipment and sent by other IoT data sources, verifying that for each event Evt_j, received from source Src_j at the timestamp Rt_j, and containing a list Msr_j of measurements:

$$\begin{cases} Src_i \neq Src_j \\ |Rt_i - Rt_j| \leq Tt_{th} \\ Msr_i \cap Msr_j \neq \emptyset \quad IoTMeasurements \text{ compared using their codes. See Figure 4.1} \end{cases} \tag{4.9}$$

If there is only one IoT data source for the shipment, or there is no IoT data events verifying the above conditions, then there is no consistency calculation to do. Otherwise, the **event consistency** is calculated using the following method:

$$Con_{Evt_i} = \begin{cases} 1 \quad \forall m \in Msr_i \cap Msr_j, |Val_{m_i} - Val_{m_j}| \leq Mt_{th} & Val_{m_i} \text{ is the value of m in } Msr_i, \text{ and } Val_{m_j} \text{ is the value of m in } Msr_j \\ \frac{NbCon_{Evt_i}}{NbEvt} & NbCon_{Evt_i} \text{ is the number of events concordant with } Evt_i, \text{ and } NbEvt \text{ is the total number of events verifying the above consistency conditions} \end{cases} \tag{4.10}$$

The **shipment consistency** Con_{Shp} gives an overview of the shipment data consistency between all the data sources related to the shipment and monitoring the same transport conditions. It is calculated as an average of the shipment related $Assignment$ consistency: Con_{Assign}.

$$Con_{Shp} = \frac{\sum_{i=1}^{n} Con_{Assign_i}}{n} \quad \text{such as } Con_{Assign_i} = \frac{\sum_{j=1}^{n_{EvtAssign}} Con_{Evt_j}}{n_{EvtAssign}} \tag{4.11}$$

4.5.4 Currentness

In the logistic traceability context, the Currentness dimension may not be critical. Indeed, the most important is to detect incidents, even though the data is received late. However, Currentness may reveal incidents concerning data acquisition. Thus, in the proposed traceability

smart contract, the stakeholders have the ability to define the shipment Currentness threshold accordingly to the use case.

Currentness levels

We consider the following Currentness levels: *IoT data event*, *IoT data source* and *shipment*.

For an **IoT data event** e_i, the currentness Cur_{Evt_i} is calculated based on the previous event reception timestamp t_{i-1}, the update interval defined in the smart contract I, the expected next event timestamp t_{i+1} which is equal to $t_{i-1} + 2 * I$ and the current event reception timestamp t_i. The t_0 corresponds to the start time of the shipment and the IoT data source assignment relationship. For the shipments, the interval I is a shipper requirement that should be met through the sending of an IoT measurement to the smart contract, every time that this interval has elapsed. Consequently, the Currentness indicates not only the quality of the data but also the meet degree of one of the more important shipper requirements defined in the smart contract, the shipment update interval I. The event Currentness at the shipment level is calculated using the following formula:

$$Cur_{Evt_i} = \begin{cases} 1 - \frac{|(t_{i-1}+I)-t_i|}{I} & \text{If } t_i \in]t_{i-1}, t_{i+1}[\\ 0 & otherwise \end{cases} \tag{4.12}$$

Furthermore, the Cur_{Evt_i} at the **IoT data source** level is calculated using the same above method, but it is worth noting that the IoT data source has its own update interval that could be different from the shipment update interval. Regarding the IoT data source, the Currentness corresponds to the degree in which the data source has met the update interval time requirement, in the history of all its related events, including the last received event. This dimension helps the users in the choice of the data sources to be assigned to the shipment, users will always choose the data source with the highest Currentness among the available data sources. The IoT data source Currentness Cur_{Src} is calculated as the average of all the data source related IoT events:

$$Cur_{Src} = \frac{\sum_{i=1}^{n} Cur_{Evt_i}}{n} \tag{4.13}$$

From the **shipment** perspective, the Currentness indicates the degree in which the shipper update time interval requirement has been met for the shipment by all its related data sources, during the shipment data sources association time period. To measure the Currentness performance of the shipment data source association, the Currentness calculated for this association Cur_{Assign} is saved in the $Assignment$ object. This Cur_{Assign} is useful when the shipment stakeholders need to investigate a low shipment Currentness, it helps to identify the shipment related data source(s) responsible(s) of the low Currentness value. The shipment Currentness Cur_{Shp} corresponds to the average Cur_{Assign_i} of all its n related $Assignment$ ob-

jects. The Cur_{Assign} is calculated as a Cur_{Evt_j} average of the $n_{EvtAssign}$ events received from the data source for the shipment, during their $Assignment$ association:

$$Cur_{Shp} = \frac{\sum_{i=1}^{n} Cur_{Assign_i}}{n} \quad \text{such as } Cur_{Assign_i} = \frac{\sum_{j=1}^{n_{EvtAssign}} Cur_{Evt_j}}{n_{EvtAssign}} \tag{4.14}$$

Currentness incidents

There are two currentness control points, the reception of the IoT measurement by the stakeholder Information System (shipper IS, carrier IS or consignee IS) and the reception of the measurement by the smart contract. In case of non-reception of the IoT measurement by the stakeholder IS, this leads to a missing update on the smart contract side. The IoT data source is configured to send a measurement every n seconds. If this interval has elapsed and no new measurement has been received from the IoT data source, the situation is considered as a missing update problem.

The missing update is not critical if the IoT update interval I_{sc} of the smart contract is larger than n seconds, because the smart contract generally does not wait for a new measurement as long as this update interval does not expire. In contrast, if the update interval is equal to n seconds, the stakeholder IS notifies the smart contract in case of missing data. Once notified, the smart contract assigns a missing update related incident to the data source owner. The update-missing incident created by the smart contract for every non received expected event, traces the history of missing data incidents related to an IoT data source. The origins of this kind of incidents are multiple, for example: the IoT data source is not able to connect to the IoT network, the IoT data source has internal problem or an IoT cloud data platform problem.

4.5.5 Conclusion of the IoT Data Qualification Proposition

In this section, we detailed the IoT data qualification proposition. This proposition includes an IoT data quality model adapted to the logistic chain context, and handling the accuracy, completeness, consistency and currentness dimensions. For each dimension, we presented the object levels concerned by the dimension, the dimension calculation methods at each defined level and the incidents related to the dimension.

The following section presents the implementation of the IoT data qualification proposition into a smart contract and the evaluation of this implementation.

4.6 Evaluation of the Proposed IoT Data Qualification

The objectives of this section are: *(i)* to evaluate the proposed quality measures; *(ii)* to evaluate the impact of the IoT data quality module on the number of created incidents; and *(iii)* to

evaluate the impact of the IoT data quality module on the IoT data event insertion time in the blockchain.

We evaluated the proposed quality measures to measure their pertinence and performance. We also monitored the number of quality incidents created to highlight the impact of the quality module. The number of *shipment* incidents was also monitored to emphasize the impact of the quality module on the business decisions.

The IoT data event insertion time in the blockchain was also measured in the evaluation tests. Firstly with the quality module activated and then with the quality module inactivated. This is to evaluate the impact of the proposed quality module on the data event insertion time and ensure the final users that this time is acceptable while ensuring the data quality.

4.6.1 Smart Contract Architecture

For the implementation purpose, we used the same architecture presented in Chapter 3. It is based on the use of Hyperledger Fabric as the blockchain implementation, with three peers (stakeholders): a *shipper*, a *carrier* and a *consignee*.

In the existing traceability smart contract, we updated the $addIoTEvent$ smart contract method with the following new functionalities: *(i)* calculate the event quality measures; and *(ii)* update the IoT data source and the quality measures of the *shipments* related to this IoT data source. Additionally, we enriched all the smart contract data model classes with the attributes presented in Figure 4.1 to handle the data quality.

4.6.2 Evaluation Experimental Choices

Due to a lack of real data to evaluate the proposed architecture in the selected use case, we chose to simulate IoT logistic data with a well-known dataset in the IoT domain.

The Intel Berkeley dataset is a collection of sensors' data, collected by Intel research team in the Intel Berkeley Research lab, between 28 February and 5 April 2004 [88]. An example of the dataset content is depicted in Table 4.3.

Date	**Time**	**Event ID**	**Sensor ID**	**Temperature**	**Humidity**	**Light**	**Voltage**
12 March 2004	16:29:04.084098	39302	1	21.8308	43.5855	165.6	2.53812
14 March 2004	15:45:11.669786	44974	2	26.9464	41.814	264.96	2.54901
19 March 2004	19:01:21.094445	59766	3	21.9092	45.1103	39.56	2.44412
...	...	...	...	...	...	...	...

Table 4.3: Samples of the Intel Berkeley dataset

To adapt this dataset to logistic chain context, we considered every sensor as an IoT data source. This gives us 54 data sources to be handled. For the *shipments*, we used every 24 h of

sensor data collection as a *shipment*, which results in 2052 *shipments* (54 sensors multiplied by 38, the number of data collection days), for the whole dataset.

Furthermore, we considered only the temperature measurements in this evaluation because it is the main measurement of the thesis medical equipment use case, but the module could be used to handle any other measurement type.

We began the evaluation phase by defining the user's quality thresholds requirements for all the data sources and *shipments*. We used the same threshold for the data sources, the *shipments* and the above-presented data quality dimensions. We made a series of tests by varying the defined threshold, going from 0 (no quality constraints) to 1 (strict quality), to show the impact of these thresholds on the number of created quality and *shipments* incidents.

In this evaluation, we established a weight measure from 1 for low importance to 4 for high importance. For the quality dimensions based on their importance for the thesis use case, we chose the following weights: a weight of 4 for the accuracy, the completeness, and the consistency, which are the most important for our users, and a weight of 1 for the currentness, which is not as critical as the other dimensions, as explained in Section 4.5.

For the *shipment* incidents, we chose an accepted temperature interval of 20 to 25 °C based on the work of Hui et al. [56]. This corresponds to an office ambient temperature interval comparable to the Intel dataset collection context. Beyond this temperature interval, if the received event quality is compliant with the *shipment* quality threshold, this event results in a *shipment* incident created for all the *shipments* that have an active assignment relationship with the event data source.

There was no information in the dataset about the sensor's precision value. Consequently, we chose to set this value to 0.5 °C, which is a recurrent precision value of temperature sensors.

In the following evaluation results, we did not consider the sensor 5 from which we did not see any event. We also ignored some other events with the sensor IDs 55, 56 and 58, because in the dataset reference the number of sensors was only 54, and events coming from the same sensor with the same event number (113,474 events in the dataset).

There were also 355 events in the dataset that we could not parse correctly due to their data presentation errors and 526 incomplete lines, from which we could not get all the event required data. This results in a total of 2,199,327 events integrated correctly in this evaluation, from a total of 2,313,682 events present in the dataset.

We used the event timestamp in the dataset as an event reception timestamp in this evaluation. Moreover, we used this timestamp to order and identify the events, for *shipment* incident creation and closing purpose. The results of this choice were 10,299 duplicated events, because they had the same timestamp as previously received events from the same sensor.

Furthermore, we use the quality threshold to define stakeholders' requirements of events quality indexes to be integrated in the data source or sent to *shipments*. All the events with

a quality index below the defined quality threshold value result in a quality incident and are not used to create *shipment* incidents in case of non-compliance with the agreed transport conditions. If the quality incident is detected by the data source, it will not send the event to its related *shipments*.

4.6.3 Results Concerning the Accuracy, Completeness and Currentness Dimensions

Firstly, regarding the accuracy, the sensors used to collect the Intel Berkeley dataset, a valid temperature value should be in the range of 0–50 °C according to [1], otherwise we consider this temperature as inaccurate.

Regarding the completeness, we used the following parameters: the update interval of 31 s, the maximum timestamp among the already integrated events timestamps, the start IoT data source and the *shipment* start timestamp. We set the IoT data source start timestamp at 28 February 2004 at 00:00:00 am, and, for the *shipment*, the start timestamp is the *shipment* date and the start time set at 00:00:00 am and the end at 11:59:59 pm.

Concerning the currentness, we used the measure interval of 31 s given for the dataset. We used this same update interval for the data sources and the *shipments*. In our tests, we did not consider the difference that could exist between the event reception timestamp and the event production timestamp. This difference could affect the test and need to be addressed in future works.

Table 4.4 shows the classification of quality results obtained for the sensors (data sources), regarding the different quality dimensions defined in this work and using multiple quality threshold values. Those results show that in the 53 retained sensors: 43 have a good accuracy, 29 have a poor completeness and 29 have a lower currentness.

Quality Threshold	Accuracy	Completeness	Currentness	Quality index
0, 0.5, 0.7, 0.9 and 1	0P 1L 43G 9H	29P 22L 2G 0H	1P 29L 20G 3H	0P 38L 15G 0H

Table 4.4: Sources quality evaluation results

Regarding the global sensor quality index, most sensors (38) have a low-quality index. If the quality threshold is set to a good quality value (e.g., 0.7), only 15 sensors are usable, and, in the case of threshold of high quality (e.g., 0.9), there is no usable sensor in this dataset.

Thanks to the quality module, all the events with a quality incident problem are not integrated into the *shipments* assigned to the event data source, and this keeps the *shipment* events quality at the level fixed and agreed by all the stakeholders. For example, in the case of Sensor 45, when we set the quality threshold at 1, 9% of the events received from this sensor have not been integrated into the source related *shipments*, due to their quality problems.

In Table 4.5, we can clearly see the impact of the threshold choice on the percentage of quality incidents. This percentage represents the events that do not respect the agreed quality thresholds. The events are filtered at the data source level according to the selected quality threshold value.

Shipments Quality Threshold	**Percentage of Quality Incidents**	**Percentage of Shipments Incidents**
0	0	0.21
0.5	25	0.4
0.7, 0.9 and 1	21	0.3

Table 4.5: Quality and *shipments* incidents results according to the quality threshold

Consequently, the percentage of quality incidents drops from around 25% of the total received events for a threshold at 0.5 to around 21% when the quality threshold was greater or equal to 0.7. The percentage of *shipment* incidents evolution is not linear due to the *shipments* number evolution depending on the selected quality threshold, as depicted in Table 4.6.

Shipments Quality Threshold	**Number of Shipments Without Any Event**	**Number of Shipments with at Least One Event**
0	421	1631
0.5, 0.7, 0.9 and 1	821	1231

Table 4.6: Shipments events number evolution

Regarding the *shipments* quality results, it is important to note that there were 421 *shipments* for which we did not receive any event, no matter what the quality threshold value was. This number increases to 821 *shipments*, when we set the quality threshold at 0.5, 0.7, 0.9 or 1, as depicted in Table 4.6. Consequently, we did not consider those *shipments* in the following *shipment* quality results, because all our quality dimension calculations are based on the events values and timestamps.

Table 4.7 shows that the percentage of *shipments* with a high accuracy level increase as the *shipments* quality thresholds increases, and this is the same for the currentness. The percentage of events with a poor completeness index increases due to events blocked by the quality threshold at the data source level.

Quality Threshold	**Accuracy (in %)**	**Completeness (in %)**	**Currentness (in %)**	**Quality index (in %)**
0	26P 1L 1G 72H	48P 29L 19G 4H	18P 30L 40G 13H	27P 16L 44G 13H
0.5, 0.7, 0.9 and 1	0P 0L 0G 100H	64P 19L 17G 1H	12P 32L 41G 15H	2P 47L 42G 10H

Table 4.7: Shipments quality evaluation results

The *shipment* quality index is also improved by the quality threshold increase; for example, we went from 27% of poor data quality *shipments* when the quality threshold was at 0 to only 2%, when the quality threshold was up to 0.5.

4.6.4 Results Concerning the Consistency Dimension

For the consistency evaluation, we selected four groups of sensors placed in proximity zones, as depicted in Figure 4.2: {1, 2, 3}, {11, 12, 13}, {15, 16, 17} and {49, 50, 51}. For each group, we linked each sensor to all its related sensors *shipments* in the same sensors group. The total

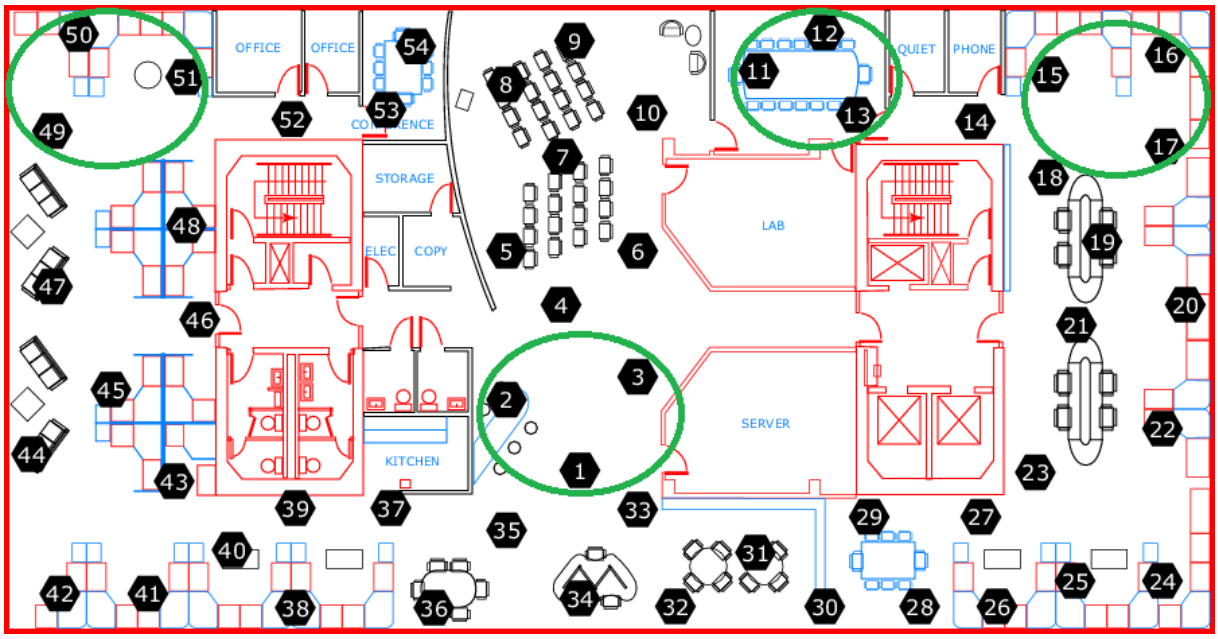


Figure 4.2: Intel Berkeley sensors arrangement diagram.

number of *shipments* related to the selected groups was 456 (12 sensors multiplied by 38 data collection days). Whatever the quality threshold value was, there were 84 *shipments* related to the selected sensors groups, for which we did not receive any event from the sensors. This number increases to 171 *shipments* when we set the quality threshold at 0.5, 0.7, 0.9 or 1, due to the events quality filtering at the data source level.

Furthermore, we set in this evaluation the tolerance time interval to 31 s and the consistency tolerance temperature to 0.5 °C. This means that two events are considered as eligible to the consistency test only when their timestamps difference is lower than 31 s, and they are considered as concordant if their reported temperatures difference is lower than 0.5 °C.

Table 4.8 summarizes the consistency evaluation results for the selected sensors groups. The group {1, 2, 3} has at least 76% of its *shipments* with a high consistency index. Those results show that the events reported by the group {1, 2, 3} were more concordant than those reported by the other groups.

Quality Threshold	Sensors group	Consistency (in %)
0	{1, 2, 3}	0P 3L 21G 76H
	{11, 12, 13}	0P 0L 73G 27H
	{15, 16, 17}	0P 0L 74G 26H
	{49, 50, 51}	0P 0L 68G 32H
0.5, 0.7, 0.9 and 1	{1, 2, 3}	0P 0L 15G 85H
	{11, 12, 13}	0P 0L 62G 38H
	{15, 16, 17}	0P 0L 88G 12H
	{49, 50, 51}	0P 0L 81G 19H

Table 4.8: Shipments consistency evaluation results

The consistency results for the selected groups were generally good to high, except for 3% of *shipments* related to the group {1, 2, 3}, when the quality threshold was at 0. This shows the impact of the quality threshold on the consistency quality results.

4.6.5 Impact of the IoT Data Quality Module on the IoT Data Event Insertion

For the smart contract IoT data quality evaluation, and due to our blockchain architecture response time (around 1 s per operation), we selected a sample of 3000 events from the dataset. This sample corresponds to the first 1000 events received from the Sensors 1–3 on 28 February 2004.

The average response time of the $addIoTEvent$ using the 3000 events data sample was around 1.7 s, with an average standard deviation of 0.174 s. When we disabled the quality module, with the same data sample, the average response time of this method drops to around 1.6 s, with an average standard deviation of 0.158 s.

This result shows that the proposed IoT data quality module adds only around 0.1 s to the event integration time. The additional quality module cost is acceptable regarding the data quality improvement brought by this module.

4.6.6 Related Works Discussion

As shown in Section 4.4, the works of Casado-Vara et al. [26], Hang et al. [48] and Leal et al. [80] are the closest to our work.

Casado-Vara et al. [26] proposed a vote method to address the accuracy and the consistency problems. Their vote method is based on the game theory to find a cooperative temperature among all the used temperature sensors. It is not applicable in the logistic chain context,

because we have different data sources owned by different stakeholders, and we need to report all the data sent by those data sources for audit purpose.

In case of discrepancy between the stakeholder's data sources related to the same *shipment*, we need to trace this discrepancy, and, if it goes below the fixed quality threshold, a corresponding quality incident is created by the smart contract. However, the vote method in [26] could be used in the very specific case of many *shipments* with similar data sources, the same *shipper*, the same *carrier* and from which we want to have a global measure trend.

Hang et al. [48] proposed a Hyperledger Fabric based architecture. This blockchain implementation choice is perfectly adapted to the logistic chain use case, and we used the same in the architecture presented in Chapter 3. However, they did provide any details in their article about the method used to handle the outlier filtering inside their proposed smart contract. Outside the blockchain, they addressed the accuracy problem (outlier filtering) using the Kalman filter, the consistency and the currentness.

It is worth noting that the standard version of Kalman filter did not meet the logistic chain transport conditions needs, because the outlier interval limits are not fixed and evolve according to the received data. This could be problematic when the Kalman filter goes in fail mode, as stated by Berman [18]. The usage of an assisted version of the Kalman filter needs to be explored in future work.

Leal et al. [80] proposed an Ethereum traceability-based architecture. Their Ethereum choice is justified by the solution monetization goal. However, we chose to work with Hyperledger Fabric which does not need any cryptocurrency management and has an organization architecture more adapted to the B2B logistic chain context, in terms of data access levels management.

In addition, authors in [80] addressed the accuracy, consistency and currentness problems using probability distribution methods, but they did not provide further details about their application and evaluation of those methods.

Furthermore, the authors of [80] proposed to filter the data inside and outside the blockchain, which is a good idea, and we already have in our architecture the inside blockchain data filtering. Besides, we need to explore the adding of a data filtering first level outside the blockchain, in future works.

The outside blockchain filtering needs to be done carefully, because it should not prevent the blockchain from getting the required traceability data; although, in some cases these data will be outliers, they need to be traced for further audit purposes.

4.6.7 Conclusions on the Evaluation

This evaluation section demonstrates the pertinence of the proposed IoT data quality module and the impact of this module on the data to be used in the traceability smart contract. The entire data qualification process is executed in a secured and distributed application on which users agree on every datum to be included, on its qualification process and decisions to be taken based on this datum.

It is worth noting that quality thresholds choice has a huge impact on data filtering process set at data source level. The events with a quality index below the defined quality threshold will never be sent to the *shipment*. This leads directly to data loss at the *shipment* level. For this reason, stakeholders may prefer selecting a good quality threshold ([0.7,0.9]), rather than a high one ([0.9,1]).

Although the proposed architecture evaluation shows encouraging results, this architecture still needs to be tested in a real-life scenario with real logistic data and additional stakeholders to get more information about its real performances.

4.7 Conclusion

In this chapter, we proposed a distributed architecture and a smart contract to enhance the IoT data quality in the context of logistic traceability. The proposed architecture uses a model of IoT data quality with four main data quality dimensions: accuracy, currentness, completeness and consistency.

We also proposed an approach for the calculation of the selected data quality dimensions. The dimensions calculation results are used in our traceability smart contract to set and control the data quality of events, data sources, *shipments* and *shipments* data sources associations.

The proposed architecture ensures the stakeholders agreement on the data quality calculation and application rules, and consequently their trust in the decisions taken automatically by the traceability smart contract. We evaluated our proposed IoT data quality assessment architecture based on an online available dataset, and the results show the relevancy of this architecture.

This work could be extended by evaluating the scalability of the proposition when adding more stakeholders and real logistic chain data. The approach used to calculate the quality dimensions could be combined with algorithms, such as DBSCAN [40] or an assisted version of the Kalman filter [66], to improve the quality index calculation.

The blockchain data charge could be alleviated by adding in this architecture a first level of data filtering on each stakeholder side. The IoT data sources' security and interoperability also

need to be addressed. Also, the architecture evaluation needs to be done in a real-life scenario to ensure its performance in the context of logistic chain traceability.

Finally, the volume of data collected from IoT objects and qualified by the proposed architecture could not be handled neither by human operators nor by classic smart contracts. This issue is addressed in the next chapter.

Chapter 5

Deep Learning Integration in Blockchain: A Traceability Incidents Prediction Use Case

5.1 Introduction

The advent of new traceability systems based on the Internet of Things (IoT) and blockchain improves data collection, security, and transparency of traceability systems.

However, the volume of data collected by this new generation of traceability systems could not be handled neither by human operators nor by classic blockchain smart contracts. Consequently, enhancing smart contracts with self-learning capabilities may ensure an efficient processing of all the available data while taking advantage from this data for logistic chain improvement decisions.

DL could help in this matter, as one of the most promising recent advances in AI domain. According to [134], the DL could be defined as "a process not only to learn the relation among two or more variables but also the knowledge that governs the relation as well as the knowledge that makes sense of the relation". The DL is a subdomain of Machine Learning (ML) which is a subdomain of the AI, as depicted in Figure 5.1.

However, to integrate DL in blockchain based systems, we should ensure that the same learning and prediction processes have been applied by all the blockchain stakeholders. In addition to the model accuracy, the response time of the DL model to be selected is a determinant criterion for the DL model evaluation and selection.

The DL-blockchain integration ensures the stakeholders' agreement on the model train, update, and prediction processes. The incident detection process gains in trust, transparency, and the incidents learning, and prediction processes are totally automated.

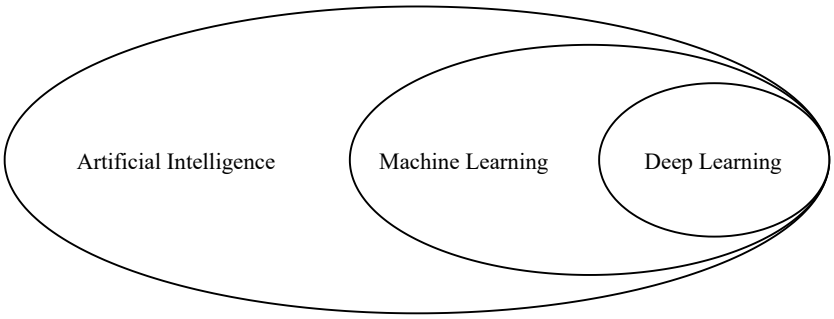


Figure 5.1: Artificial Intelligence, Machine Learning and Deep Learning Relationship

In the literature, the AI and blockchain combination has been studied by many of recent works, such as [112], [38] and [32] . However, there are few works on the AI integration inside the blockchain as in [124], [49] and [125]. In these works, some technical limitations to this integration have been reported by the authors, related essentially to the absence of support of some ML required mathematical functions by the blockchain used implementations: Quorum in [124] and Ethereum in [49] and [125].

To overcome the above limitations and answer the thesis **RQ3**, we propose in this chapter to implement a DL model using TensorFlow which is a framework implemented by Google Research Team for ML algorithm expression, implantation and execution [9].

Then the implemented DL model is integrated in Hyperledger Sawtooth which is a framework for enterprise-grade blockchain building, putting the focus on security, scalability, and modularity, with a support of Python and many other languages for smart contracts development, according to [71].

The TensorFlow and Hyperledger Sawtooth combination allows to take advantage of adapted technologies for Self-Learning systems development, with Tensorflow for Deep Learning, Hyperledger Sawtooth for the blockchain and Python for Self-Learning Smart Contracts development.

The proposed architecture for DL integration in blockchain is implemented and evaluated using a real logistic shipment dataset, to show the viability and the effectiveness of this architecture.

To explore the DL integration in blockchain based systems, the main contributions of this chapter are:

- the selection of a Deep Learning (DL) model for traceability incident prediction, to handle proactively these incidents;
- the integration of the selected DL model, its related training, prediction, and update processes into a traceability blockchain smart contract;
- the performance evaluation of the proposed architecture using a real logistic chain shipment dataset

The rest of this chapter is organized as follows. Section 5.2 discusses the ML and blockchain integration literature. The DL model selection method is detailed in Section 5.3. In Section 5.4, we present the proposed architecture for DL model integration in the blockchain. The proposed DL blockchain integration implementation and evaluation results are highlighted in Section 5.5. Section 5.6 concludes the chapter.

5.2 Machine Learning and Blockchain Integration Related Works

In this section, we study the research works that integrated ML approaches into Blockchain based architectures, using the following criteria:
(C1) ML model blockchain integration: where is deployed the ML model in the architecture? Inside or outside the blockchain? In the literature, both solutions have been proposed. In our context, data and learning process security and transparency among stakeholders is a key issue. Therefore, we need to deploy the ML model and its related processes inside the blockchain.
(C2) ML model training mode: Is the model training done offline, online or both? This question is crucial, and the answer depends on several factors among them, the training data availability, and the stability of the environment. Online training is appropriate when real world data are provided continually for training. In this case, the system continues to learn new concepts while preserving already learned information. On the contrary, offline training uses pre-acquired and stored data. This more classical learning mode guaranties stability for ground concepts. In our use case, these two conflicting requirements are needed to both initialize the process and deal with a dynamic environment.
(C3) Learning approach: which learning approach is used by the authors, supervised, semi-supervised or unsupervised? Supervised learning uses labeled data whereas unsupervised learning consists of working with unlabeled data. However, for anomaly detection, Semi-supervised learning is a much more popular option since anomalies are rare events and generally, only few data are labelled. Since the incidents are rare events in our context, in this chapter we are focusing on **semi-supervised learning methods**.
(C4) Blockchain implementation: What blockchain implementation is used? Is it permissioned or permissionless? Permissioned blockchains ensure for each one of the stakeholders the identity of the other. Since in this thesis we focus on B2B logistic chain in which the stakeholders' number is limited and they are known, **permissioned blockchain** are more suitable in this context.
(C5) Application domain: what is the domain of application of the work? This chapter targets a prediction process for **logistic traceability incident**.

A Systematic Literature Review (SLR) on Artificial Intelligence and Blockchain integration is proposed by [38]. In this SLR, the authors highlighted the key benefits of this integration and its main studies: classification or clustering, distributed management, security improvement

and prediction or decision making.

Since blockchain and AI technology are data driven, Chen et al. [32] proposed a research review on their combination. Their review shows some applications of this combination for Bitcoin transaction entity classification, Bitcoin price prediction without deep methods, privacy preserving, computing power allocation using deep methods, IoT, and Bitcoin mining using Reinforcement Learning.

Blockchain and AI integration challenges are reviewed by [112]. According to these authors, the Blockchain and AI integration provides application with enhanced data security, improved trust on Robotic decisions, collective decision making and high efficiency.

The works to be compared with our work could be divided into two main categories according to the ML model integration type (outside or inside the blockchain).

5.2.1 ML model outside the blockchain

An LSTM (Long Short-Term Memory) model trained outside the blockchain and integrated in a smart contract is proposed in [89]. It implements a credit evaluation system that adopts blockchain technology to strengthen the supervision and management of traders in food supply chain.

Authors in [46] trained a Neural Network to classify incident reports submitted by users using the report content analysis (high, middle priority). The result of the analyze is submitted to an Ethereum smart contract that handles the report lifecycle, passing by acquisition, use, archival and disposal steps.

The integration of AI in the blockchain and the limitations related to this integration were explored in [12]. The authors identified the following elements as the main factors influencing this integration: the mining mechanism, the choice of data storage on or off-chain, the blockchain types (public or private) and the blockchain data integrity. Due to the cost of transaction processing in public blockchain, the authors proposed to train the AI model on off-chain data, then the smart contract could use the model through APIs (Application Program Interfaces) implemented specifically for this purpose. The authors gave some application examples, essentially on a recommendation engine for supply chain contract negotiation. However, they did not cover supply chain incidents management.

To meet the food industry traceability requirement, [72] proposed an optimized architecture based on the combination of blockchain, IoT and Advanced Deep Learning (ADL). The IoT collects data throughout the entire supply chain, and the blockchain ensures a secure and trusted environment for the collected data sharing. The ADL is used to handle the supply and demand forecasting. The authors proposed to combine LSTM and GRU (Gated Recurrent Units) models, with a Genetic Algorithm (GA). The GA is used for the LSTM and GRU models hyperparameters optimization.

A literature review of Reinforcement Learning (RL) integration in Blockchain-Industrial Internet of Things (IIoT) enabled Networks, is proposed by [63]. According to the authors, among the main opportunities of this integration application there are: the forking events minimization, energy efficiency improvement, minimization of the time to finality, transaction throughput enhancing, link security improvement and average block Time reduction. In addition, the authors presented a case study on forking events minimization using a Q-learning approach. Finally, the authors identified some challenges and open research questions, among them, the integration of the RL into the blockchain for performance improvement network higher gains, and the development of new methods for appropriate RL technique selection.

For drug recommendation and Supply Chain Management, [10] proposed an architecture combining DL and blockchain. On one hand, a DL approach is used to analyze users' drugs reviews and develop a drug recommendation module for patients. On the other hand, a Hyperledger Fabric blockchain is used for the supply chain management and to ensure drugs traceability among all their stakeholders: suppliers, manufacturer, distributors, pharmacies, hospitals, doctors, and patients.

For Perishable Food Supply Chain Tracing, [113] proposed a blockchain based architecture to secure the data sharing and improve the data quality. The blockchain data is used by an IBM Watson Machine Learning platform for perishable food expiration date prediction. Furthermore, a Fuzzy Logic approach is used to evaluate the food quality based on humidity, temperature sensors and the transit time.

This first category of works does not integrate the ML model inside the blockchain and consequently there is no guarantees for the blockchain stakeholders on the correct application of the agreed model training, update, and prediction processes

5.2.2 ML model inside the blockchain

The trust and automation problems of machine learning were addressed in [124]. The author proposed a blockchain smart contract-based framework, in which submission and rewards methods for machine learning models were implemented using a smart contract. He used Association Rule Mining (ARM) as demonstration example for his proposed blockchain machine learning integration framework. However, as stated by the author, his work was at a preliminary stage and the evaluation was limited due to his limited dataset and his blockchain implementation choice.

For incremental learning on blockchain, [49] proposed a smart contract implementation for training and updating machine learning models inside the blockchain. The authors proposed an incentive mechanism to encourage users to submit data that improve the model's accuracy. However, users can use freely the smart contract model to predict data. In addition, the authors highlighted some limitations related to the implementation of machine learning in Ethereum, such as the gas cost and the absence of support of Floating-Point Numbers.

While the trust and automation are handled by the blockchain in blockchain machine learning based architecture, the data efficiency still needs to be addressed to bring this kind of architecture in the edge. To address this issue, [125] proposed an approach in three steps: model training in off-chain, model saving in a secured and immutable ASTORE data format to be compatible with the blockchain, using the saved model to score new data online.

Authors in [49] and [125] worked with Ethereum which is a permissionless blockchain, with a cryptocurrency gas cost, an energy-intensive consensus algorithm (Proof of Work), and consequently is not adapted to the enterprise context which needs more a permissioned blockchain. Additionally, the Ethereum Virtual Machine (EVM) lacks of native support of many representations and functions, such as Floating Point Numbers and $exp()$ function, required by many of machine learning techniques, as stated by [78].

5.2.3 Related works conclusion

After the above study of the related works, we did not find any work meeting all the criteria of model integration inside the blockchain (C1), with offline and online training support (C2), using a semi-supervised learning approach (C3), and a permissioned blockchain (C4), for logistic traceability incident prediction (C5). Consequently, we decided to select a deep learning model adapted to the traceability incident prediction needs, and we integrated it inside a permissioned blockchain. The blockchain smart contract ensures the model training, updating, and predicting processes. This provides the logistic chain stakeholders with a self-learning smart contract, an automated, trusted, and transparent learning and prediction process, and guarantees the stakeholders agreement on the model management and its resulting incident predictions.

Table 5.2.3 summarizes the related works comparison.

Ref.	ML model blockchain integration (C1)	ML model training mode (C2)	Learning approach (C3)	Blockchain implementation (C4)	Application domain (C5)
[89]	Outside	Offline	Supervised	Hyperledger Fabric (Permissioned)	Trader's credibility evaluation in food supply chain
[46]	Outside	Offline	Supervised	Ethereum (Permissionless)	Cyber incident reports classification and incident management
[12]	Outside	Offline	Not specified	Not specified	Supply chain contract negotiation
[72]	Outside	Offline	Supervised	Hyperledger Fabric (Permissioned)	Food Provenance, and the supply and demand forecasting
[63]	Outside	Online	Reinforcement	Not specified	Blockchain-Enabled IIoT Networks
[10]	Outside	Offline and Online	Supervised	Hyperledger Fabric (Permissioned)	Drug Recommendation and Supply Chain Management
[113]	Outside	Not specified	Not specified	Ethereum (Permissionless)	Perishable Food Supply Chain Traceability
[124]	Inside	Offline	Unsupervised (in the evaluation)	Quorum (Permissioned)	Pharmaceutical domain (in the evaluation)
[73]	Inside	Online	Supervised	Hyperledger Fabric (Permissioned)	Character Recognition and Healthcare domains (in the evaluation)
[49]	Inside	Online and Offline	Supervised	Ethereum (Permissionless)	Movie reviews (in the evaluation)
[125]	Inside	Offline	Supervised	Ethereum (Permissionless)	Autonomous driving
This work	Inside	Offline and Online	Semi-supervised	Hyperledger Sawtooth (Supports Permissioned and Permissionless)	Logistic traceability incidents

Table 5.1: ML and Blockchain Integration Related Works Comparison

5.3 DL Model Selection

In this work, we focus on DL approaches because they outperform the classical ML approaches, in terms of accuracy and properties relationships discovery, according to many recent works [75] and [94].

Traceability incidents are considered as anomalies in the logistic chain normal functioning. In DL, autoencoders are widely used to handle the anomaly detection problem, as stated by [30]. Consequently, we decide to implement autoencoders for the logistic chain incident prediction.

An autoencoder is "a neural network that is trained to attempt to copy its input to its output" [45]. It is composed of an encoder and a decoder that are trained together. The encoder learns to compress the data and the decoder tries to minimize its reconstruction error.

The number of shipments generating incidents in the studied logistic chain dataset represents less than 10%. This is a real learning challenge. Therefore, as stated before, we decided to use a semi-supervised learning approach, and to train the model to be selected only on shipments with incidents to increase the model sensibility to this kind of shipments.

In the literature, there are many approaches to evaluate and select neural network models. In this work, we used the model evaluation, optimization and algorithm selection method proposed by [107]. It mainly consists of three steps: firstly, the performance estimation, secondly the hyperparameter optimization and finally the model and algorithm comparison.

To implement the traceability incident detection autoencoder, we study in this chapter the three main types of DL networks:

- Dense Neural Networks (DNN): the basic architecture in DL neural networks. They consist of fully connected layers. This means that every neuron in a layer receives connections from all the neurons in the previous neighboring layer.
- Recurrent Neural Networks (RNN): proposed by [110] in 1986. They adjust repeatedly the neurons connection weights in the network and introduce hidden layers for features extraction. These kinds of neural networks are suitable for sequential data prediction where current state is influenced by historical data. Classical RNN suffer from a short historical data memory issue and instability related to network weights as stated by [105]. To overcome these issues, some new types of RNN were proposed in the literature such as Long Short-Term Memory (LSTM) [54].
- Convolutional Neural Networks (CNN): proposed by [79] in 1990 for Handwritten Character Recognition. They preserve input data spatial configuration and ensure translation invariance property. Nowadays, they are widely used in computer vision and recommendation systems.

This study purpose is to measure the performance of these neural networks in term of

accuracy over the shipment dataset and select the more efficient network to be adopted for the traceability incident prediction autoencoder.

5.4 DL Model Integration In The Blockchain

Nowadays, to develop trusted and self-learning information systems based on DL, the key issue is to ensure data reliability and security, and this is exactly what the blockchain provides.

The blockchain offers natural complementarity with DL and opens new opportunities for the development of Self-Learning Smart Contracts (SLSC) [12], through the integration of DL model training and prediction processes into the smart contract logic. Therefore, we believe that this type of hybridization is essential for the cross-fertilization between DL and smart contracts. The DL model provides smart contract with self-learning capabilities, and the DL process gains automation, trust, and transparency from the smart contract.

In the logistic context, the SLSC integration is done on the top of the traceability architecture presented in Chapter 3, as depicted in Figure 5.2. The blockchain stakeholders are the shipper, the carrier, and the consignee. Each stakeholder has its own blockchain nodes and is responsible of the data integrated in the blockchain through its Information System (IS). These data are collected from many sources, for example RFID and Barcodes' readers, IoT Sensors, Connected Warehouses, Factories etc.

This architecture is designed to handle traceability data related to the transport operations. For every new transport operation, the DL model indicates its incident predictions based on the transport data provided by the shipper. This raises two possible cases:

- The model predicts an incident for this transport operation, then it is tagged with a predicted incident for stakeholders' information and action.
- No incident is predicted by the model, the transport operation is tagged as predicted incident free.

Whatever the real results of the transport operation in terms of incident, and the exactitude of the model prediction, the model is updated with this transport operation.

5.5 Evaluation

The main objective of this section is to show the feasibility and evaluate the relevancy of the proposed DL-blockchain integration architecture, in terms of model accuracy and per operation response time.

It is worth noting that we used another blockchain implementation in this evaluation. How-

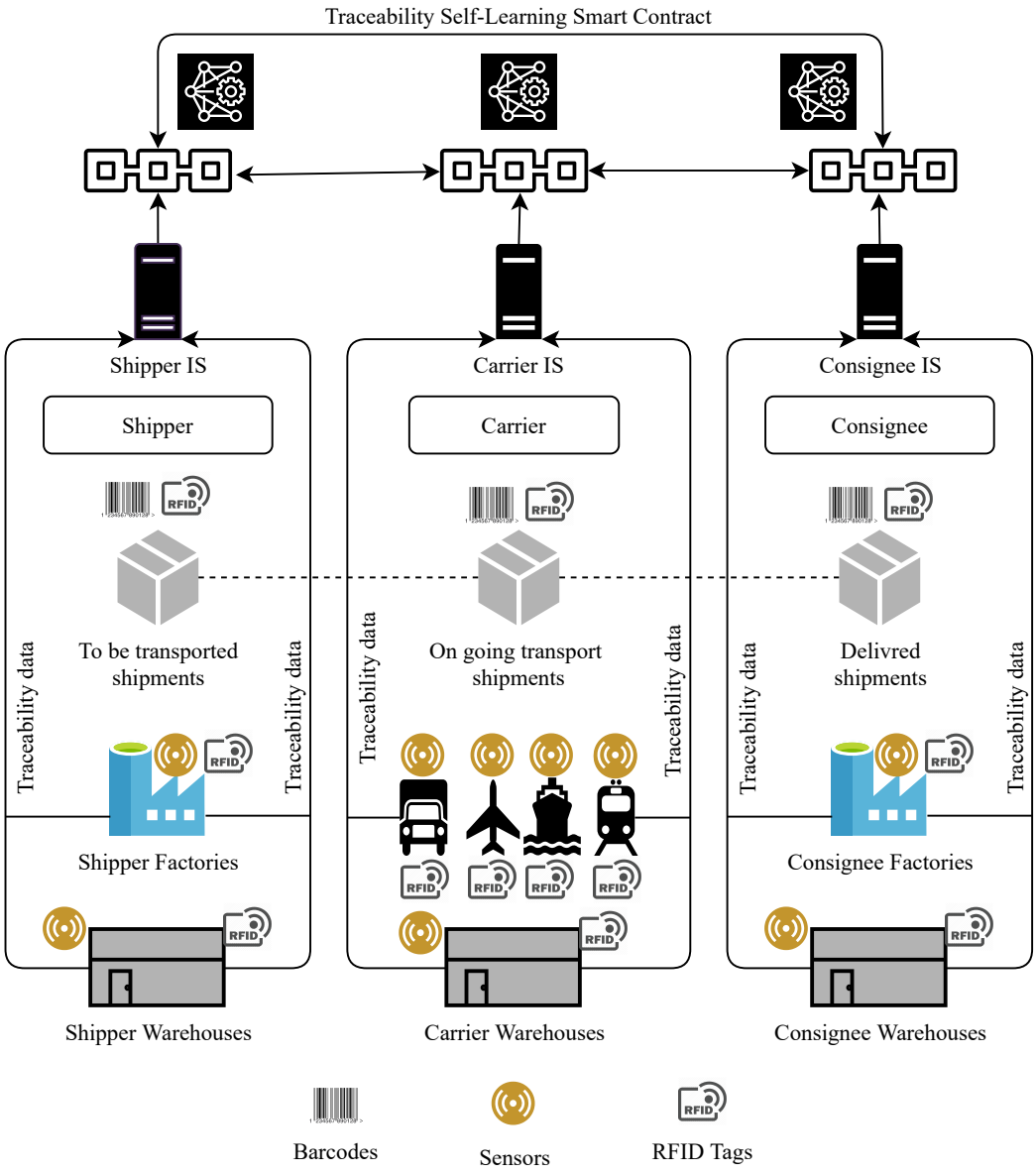


Figure 5.2: DL in Blockchain Based Traceability Architecture

ever, the core traceability architecture proposition is still the same as evaluated in Chapter 3. Therefore, we must, test the performance criterion to verify that it has not been impacted by the blockchain implementation modification.

The business requirements of this evaluation are the followings: a model accuracy of at least 50% for incidents prediction and a response time less or equal to 3 seconds for any operation, according to the performance criterion presented in Chapter 3.

In the following sub-sections, we detail a Proof of Concept of the proposed architecture. Firstly, we start by the dataset preparation and the initial DL model production using TensorFlow. Secondly, we detail the DL model integration in the Hyperledger Sawtooth based architecture. Finally, we conclude the evaluation section by a discussion of the evaluation results.

5.5.1 Dataset Preparation

The dataset preparation was divided into six steps. The first one was the work with MyTower business team on the identification of the relevant shipment data listed in Table 5.2.

Data column	**Description**
Pick slip	Shipment item id
Number of parcels	Number of parcels in the shipment.
Origin	Shipment origin country.
Carrier code	Carrier ID in the shipper system.
Number of destination	Number of destinations in the shipping request.
Number of pick slip	Number of items in the shipment.
Creation date	Date and time of the transport request creation in the shipper system.
Date of availability	The date of shipment availability indicated by the shipper.
Pickup negotiated date	The shipment pickup agreed execution date between the shipper and the carrier.
Delivery negotiated date	The shipment delivery agreed execution date between the shipper and the carrier.
Shipment volume and weight	The shipment volume and weight indicated by the shipper.
Consignee code	Consignee ID in the shipper system.
Consignee name	Consignee company name.
Consignee address	Number, street, and complements.
Consignee zip code and city name	
Consignee Country	The consignee country code. It is a simple number identifying each country in the dataset. We do not consider the origin country because there is one origin country in the shipment dataset.
Lead time	Number of days between the carrier shipment pickup and delivery dates.
Transport mode	Air, Road or Sea
Truck	Pickup truck number.
PO Ref.	Purchase Order Reference.
Service Level	The code of the transport service level agreed with the carrier.
Deadline	Delivery deadline hour.
Multi-provider	A value of 0 (No) or 1(Yes) indicating if the shipping operation involves multiples service providers.
Equipment	A value of 0 (No) or 1(Yes) indicating if the transported product is a medical equipment.
Dry ice	A value of 0 (No) or 1(Yes) indicating if the shipment transportation requires dry ice.
Dangerous	A value of 0 (No) or 1(Yes) indicating if the shipment contains dangerous goods.
Incident	A value of 0 (No) or 1(Yes) indicating if the shipment generated an incident

Table 5.2: Shipment Dataset Columns

The second step was the work on a data base dumping provided by the MyTower technical team to extract all the relevant data identified with the business team. The third step was the data transformation. For example, we extracted from the shipment creation date: the day of the week, the month, the day of the month. All these extracted data could be eventually

a source of shipment incident. The fourth step was the labels encoding using the Pandas framework (Version 1.2.4) [93] factorize function. This function helps in obtaining a numeric representation of an array values. The fifth step was the data normalization using Sklearn framework (Version 0.24.2)[103]. This step is important for the test of many neural network models, such as RNN, due to their sensibility to the input data.

The final step in the data preparation was the division of the dataset into training, validation, and test datasets. The extracted dataset is composed of 377,971 shipments, in the ascending order of their shipment creation date time. These shipments were collected between December 2016 and February 2021, by the shipper TMS. We divided this dataset into a training dataset including the first 264,579 shipments (70%), a validation dataset of 75,594 shipments (20%), and the remaining 37,798 shipments (10%) were used as the test dataset. It is worth noting that for the autoencoder training, we used only 16702 shipments (4.4%) which correspond to the number of shipments with incidents in the training dataset.

5.5.2 Model Selection Training and Tests

The objective of this section is to compare the performance of the studied neural networks over the shipment dataset, in terms of (1) validation accuracy results and (2) generated model file size.

For the generated models, we choose to store them in HDF5 format, because this format generates a single file easy to handle and store.

The model used in this evaluation is an autoencoder composed of an encoder with 4 layers (36, 16, 8 and 4 nodes respectively) and a decoder with 3 layers (8, 16 and 36 nodes respectively).

We used a common model configuration in anomaly detection [122, 133], with a ReLU (Rectified Linear Unit) activation function for all the hidden layers, a sigmoid one for the output layer and an ADAM optimizer. Also, to adapt the input data to the input structure of RNN(LSTM) and the One-dimensional CNN(Conv-1D), input data was reshaped by adding a time unit of 1. For the DNN, the input data was used directly without any reshape need.

In this work, we are interested in the incident detection, and we chose to train the model only on the training dataset shipments related to an incident to increase the model sensibility to this kind of shipments. However, the validation is done using the whole validation dataset including shipments with and without incidents.

We implemented the models using the *TensorFlow* framework (Version 2.4.1)[9], and we modified only the hidden layers by setting them to the neural network type to be tested.

In the training process, we set the epochs number to 20 and the batch size to 32, to reduce the model training time in the target blockchain architecture.

As depicted in Figure 5.3, the validation loss results of the LSTM model is low (around 0.12 for its lower value).

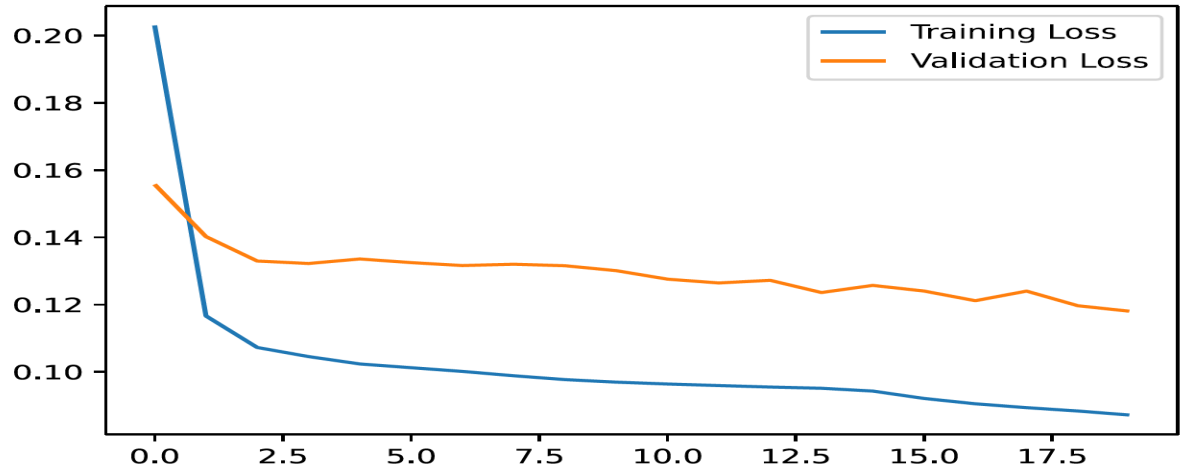


Figure 5.3: RNN (LSTM) Training and validation loss results

The confusion matrix of the LSTM model is depicted in Figure 5.4. It shows how the model handle the normal and incident shipment classification, by comparing the real shipment class with the one predicted by the LSTM model.

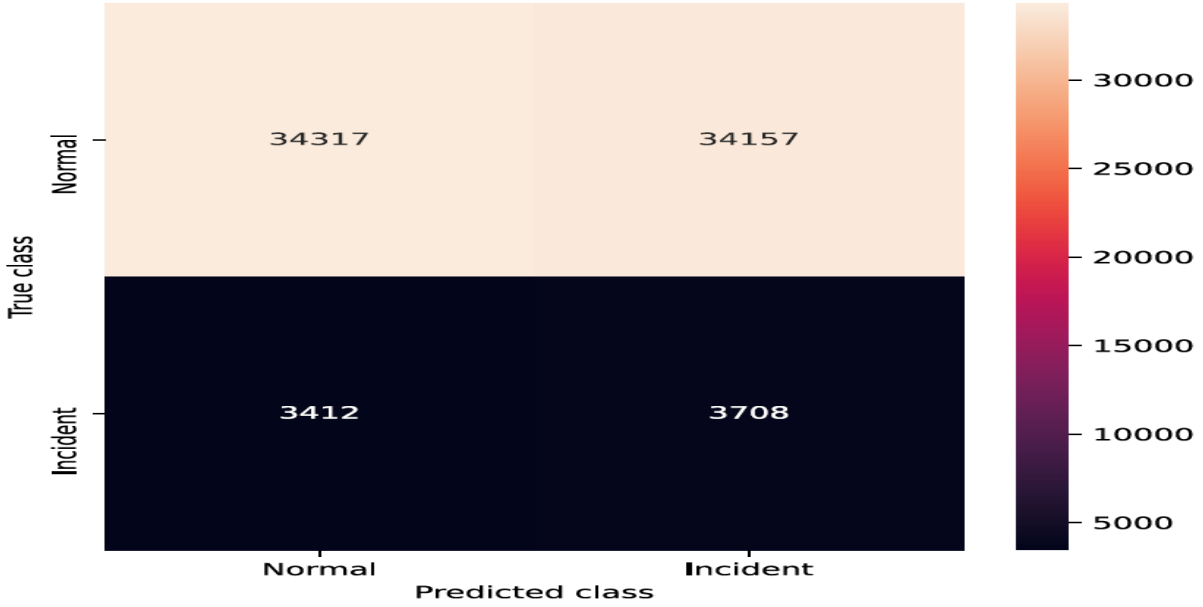


Figure 5.4: RNN (LSTM) Confusion Matrix

The models' test results are depicted in Table 5.3.

	DNN	**RNN (LSTM)**	**CNN (Conv-1D)**
Global validation shipments accuracy	50.17%	**50.30%**	50.01%
Normal validation shipments accuracy	**50.18%**	50.11%	50.07%
Incident validation shipments accuracy	50.08%	**52.07%**	49.52%
Generated model (HDF5) size in KB	**73.2**	161.9	73.5

Table 5.3: Models' validation and training results

The models' accuracy results on the validation dataset were close, however the RNN (LSTM) performed slightly better on the incident detection, with a big size file model in comparison to the DNN and the CNN (Conv-1D).

The model generated file size is an important criterion to consider in the model choice because a new version of the model is generated and saved on the Traceability Transaction Processor (TTP) node file system after each training operation. However, in this work we focus more on the model accuracy on the incident detection which is the main goal of this study.

For the following model integration into the blockchain, and regarding the above presented results, we choose to work with the RNN (LSTM) model. However, it is worth noting that we tried many model configurations and hidden and output layers combinations, but we were unable to go beyond the above presented accuracy results. We think that the data augmentation is the last avenue to explore to improve the model accuracy. For this purpose, we will need more data and more discriminant columns in the dataset to be able to identify the shipments with incidents and have consequently a model accuracy eligible to production deployment in real life logistic chain traceability scenario.

5.5.3 Hyperledger Sawtooth Network Configuration

We used the 1.1.2 release of Hyperledger Sawtooth to implement a blockchain architecture including the three selected stakeholders: the shipper, the carrier, and the consignee. All the architecture nodes are implemented using Docker to facilitate the lifecycle management of each node.

Each one of the stakeholders has the following list of nodes:

- Client: in charge of preparing and submitting the transaction batches used to group transactions. The client submits these batches to the REST API.
- REST API: the main role of this node is to adapt the communication with the validator to HTTP/JSON Standards. This communication could also be achieved using custom client-validator communication API, without need of the REST API node.
- Validator: as indicated by its name, this node validates the transaction batches submitted

by the client. It is also responsible of blocks creation, consensus and communication maintaining with the other network node. It is the main node of the architecture.

- PoET Engine: provides the Proof of Elapsed Time (PoET) consensus functionality to the architecture. It communicates with the validator through the consensus API
- PoET Validator Registry Transaction Processor: it handles the new validators addition into the network through the PoET consensus
- Setting Transaction Processor: stores all the on-chain configuration settings. For example, the validator block wait time, max transactions per block and the keys authorized to update settings.
- TTP: this is the business process node that handles the incident prediction model initialization method, the tracking events creation and the model training and update process. It is the Traceability Smart Contract node, TTP in Hyperledger Sawtooth terms.

Figure 5.5 shows the architecture view on the shipper side. The other stakeholders have their own nodes on their sides, with the same architecture.

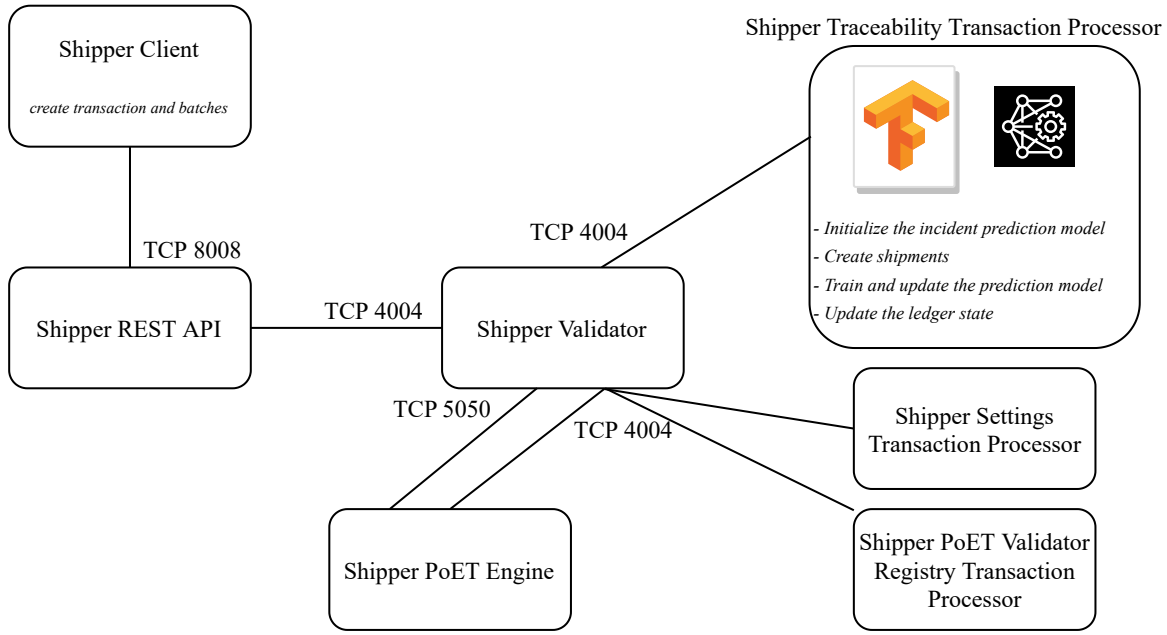


Figure 5.5: Hyperledger Sawtooth and Deep Learning Integration Architecture (Shipper view)

In the Hyperledger Sawtooth settings, we set the consensus algorithm to PoET which is a Crash Fault Tolerant (CFT) algorithm. It is adapted to companies' private collaboration networks context in which the risk of byzantine failure is limited. We also set the block initial and target generation waiting time to 10 seconds to let a sufficient time to the model training and update processes execution. The Sawtooth settings are depicted in Source Code 5.1.

```
sawtooth.consensus.algorithm.name=PoET
sawtooth.consensus.algorithm.version=0.1
sawtooth.poet.target_wait_time=10
sawtooth.poet.initial_wait_time=10
```

Source Code 5.1: Hyperledger Sawtooth Consensus Settings

For the blockchain performance monitoring we used a combination of InfluxDB (V 1.7.8) and Grafana (V 4.6.3). InfluxDB is high-performance data store built specifically for timeseries data [4]. Grafana is a powerful metrics visualization and query framework [2]. The combination of InfluxDB and Grafana allows the collection and the visualization of performance metrics in real-time. We set the shipper Rest API and the shipper validator components to send their performance metrics (transaction process duration, CPU, RAM, etc) to the InfluxDB component. The collected data is then consulted using a customized Sawtooth dashboard on the Grafana component, as depicted in Figure 5.6.

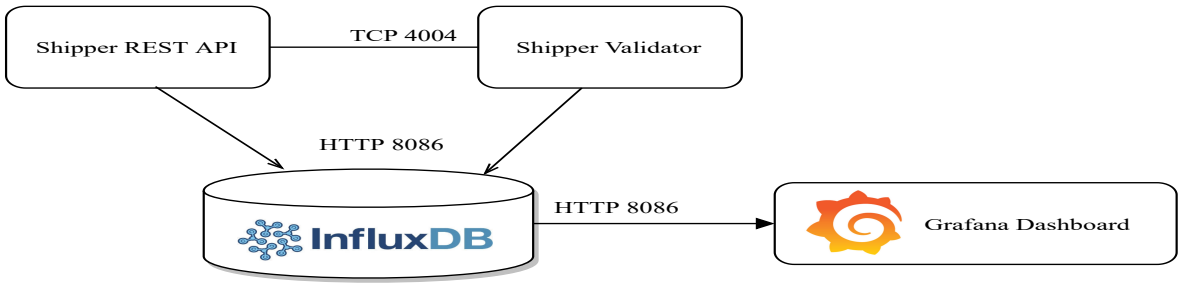


Figure 5.6: Hyperledger Sawtooth Performance Monitoring using InfluxDB and Grafana

5.5.4 Blockchain and Deep Learning Integration Settings

In this evaluation, the integration of DL models into the blockchain goes through the following steps.

Firstly, on the file system of the Traceability Transaction Processor (TTP) node, we deployed the model that will be used to initialize the blockchain architecture. In this chapter, we focus on the stakeholders' agreement on the model output rather than the model file itself. The agreement on model file needs to be addressed in future work.

Secondly, it is important to ensure the determinism of the model output results in the blockchain context. The model prediction and training operations are executed on each stakeholder TTP instance, and for blockchain coherence, these operations should give the same results on each TTP instance. For this purpose, we set the deterministic configuration in the used frameworks with a random fixed value for the seed used as a start value by random generators. We also activate the deterministic options in TensorFlow, as depicted in Source Code 5.2.

```
import os, random
import numpy as np
import tensorflow as tf

seed=42
random.seed(seed)
tf.random.set_seed(seed)
np.random.seed(seed)
os.environ['PYTHONHASHSEED'] = str(seed)
os.environ['TF_DETERMINISTIC_OPS'] = '1'
os.environ['TF_CUDNN_DETERMINISTIC'] = '1'
```

Source Code 5.2: TensorFlow Deterministic Settings

The TTP contains two main methods. The first one is used for the model training settings initialization, such as the batch size and the epochs. The second method is used for shipment creation, incident prediction and model training and update. For the sake of clarity, we call this method the shipment creation method. In the following discussion we will focus on this method and its related performance, because the first method is used only for the initialization.

We tested the performance of the shipment creation method using the test dataset. To go around the Sawtooth back pressure Deny of Service (DoS) protection mechanism, we set manually a waiting before submission interval of 30 seconds between the shipment batches, on the client side.

The shipment creation transactions are submitted by batch of 32 shipments. According to number of shipments in the dataset, the shipper handles an average of 412 shipments per day which is largely covered by the 32 shipment per 30 seconds configuration.

5.5.5 Evaluation Results And Discussion

In Figure 5.7, we can see the response time evolution of the shipment creation method with the LSTM model, during the test time. This result shows a mean transaction process duration up to 5 seconds. It does not meet the evaluation requirement of maximum 3 seconds per operation.

The LSTM Sawtooth performance test could not be completed with the current configuration because of the high transaction processing duration. Due to the long transaction processing duration and the Sawtooth backpressure protection mechanism, we got around 45 batch rejection, 80 minutes after the performance test start. To complete this test, we had to set a bigger wait time interval (at least 180 seconds) between the client batch submission which will take around 2.5 days for the whole test dataset.

We think that there are two main reasons of the poor performances related to the LSTM model training and updating time. Firstly, the complexity of the LSTM nodes and the cal-

Figure 5.7: Sawtooth-LSTM Transaction Processing duration (99th Percentile)

culation required by this model, and it is worth noting that we work in a deterministic and constrained model train and update environment. Secondly, the necessity of data reshaping before its usage by the LSTM model.

Consequently, we decide to do the tests with the DNN model which was the second-best model according to the models' accuracy results on the validation dataset. As depicted in Figure 5.8, this model shows a mean transaction duration processing less than 1 second which meets largely the evaluation per operation requirement. This test took around 9.5 hours to integrate the whole test dataset.

Figure 5.8: Sawtooth-DNN Transaction Processing duration (99th Percentile)

The above evaluation results show promising results for the integration of ML processes inside blockchain based architecture.

5.5.6 Evaluation conclusion

In this evaluation, we showed the feasibility and the effectiveness of a DL integration in blockchain architecture using TensorFlow and Hyperledger Sawtooth.

The evaluation results show that this architecture meets the above presented performance requirements of model precision and the maximum per operation time response, for the studied shipment incidents prediction use case. However, this is only a first step in the long road of DL integration in blockchain. There are still many issues to be addressed in future works on this topic:

Firstly, the prediction model accuracy could be improved by working on the model architecture and fine tuning. We also need to predict the incident type and go beyond the simple action of incident prediction by recommending the stakeholders some actions to avoid the predicted incident. However, to be able to do so, we need more data than what we have. This should be addressed in future work in collaboration with the customer.

Secondly, in this evaluation, we directly used the normalized data to train, validate and test the proposed architecture. The correspondence between the normalized and the original data needs to be addressed in future work.

Finally, to secure the model used for incident prediction, we need to ensure that the same model version has been used on each stakeholder side. This issue could be addressed by working on unique signature for the generated model that could be stored on the blockchain to ensure the security and the uniqueness of the prediction model, and the stakeholders' agreement on the model file.

5.6 Conclusion

In this chapter, we proposed an architecture for DL integration in Blockchain using a traceability incident prediction use case. Firstly, we studied the state of the art of ML and blockchain integration and defined five criteria to check. Secondly, we selected a DL model adapted to the logistic chain traceability incident prediction and we integrated this model in an enterprise adapted blockchain, to secure and ensure the stakeholders' agreement on the model training, updating and its output prediction results. The obtained solution fulfills the pre-defined criteria. Finally, we evaluated the proposed architecture using a real logistic chain shipment dataset, and the evaluation results show the viability and the relevancy of the proposed architecture.

It is worth noting that this work is only a first step in the DL integration in blockchain. However, it opens new opportunities for the development of trusted DL and self-learning blockchain based systems.

To improve the proposed architecture, future works will focus on improving the model accuracy, handling the data normalization, securing the generated model by integrating a hash of this model in the blockchain and exploring model-less learning approaches for DL integration in blockchain architectures.

Finally, in the logistic chain, the traceability is not the unique use case for this integration. It could be used for example in transport booking and price predictions. Beyond the logistic domain, this integration could be used in any application involving multiple stakeholders and requiring transparency and stakeholders' agreement on the ML processes. It would merely be necessary to adapt the blockchain settings to the stakeholders' numbers and to use an ML model suitable for the context dataset. However, the blockchain-ML integration remains the same as in the architecture proposed in this chapter.

Chapter 6

Conclusion and Perspectives

In this thesis we have addressed the following logistic chain traceability enhancement issues. Firstly, the decentralized, secured, and transparent sharing of the traceability data and its related processing rules among all the stakeholders. Secondly, The IoT data qualification and the stakeholder's agreement on the data qualification rules. Finally, the efficient and intelligent management of the huge data volume generated by the IoT data source.

6.1 Contributions

In this objective, we focused on the three contributions presented below.

1. We have proposed a traceability architecture combining blockchain and IoT. The proposed architecture uses blockchain and smart contracts to implement a secure and trusted traceability process shared among all the logistic chain stakeholders. Through a generic traceability smart contract, the proposed architecture handles the logistic chain milestones as well as incident management based on data collected automatically using the IoT.

2. This architecture allows to push forward the traceability automation for detection of incidents and decision making. In this context, it is essential to ensure user trust in the traceability process. For this purpose, we have included in the architecture an IoT data qualification module. The proposed module uses blockchain smart contract to qualify IoT data. It deals with the following data quality dimensions: Accuracy, Completeness, Consistency and Currentness. The stakeholders may define quality thresholds for data quality control. The use of the smart contract ensures the stakeholders' agreement on the IoT data qualification rules. They agree on the definition and the implementation of each data quality dimension using the smart contract. The IoT data quality is calculated and aggregated at four levels: (1) IoT data event, (2) IoT data source, (3) shipment and (4) IoT data source-shipment relationship. This provides the stakeholders with a fine con-

trol and monitoring of the IoT data quality and allows the detection of defective IoT data sources.

3. Moreover, to take advantage of the high volume of data collected by the IoT and be able to detect proactively traceability incidents and improvement actions, we have proposed to empower the traceability smart contract with learning capabilities using DL. This proposition secures the DL model results and its related data and guarantees the stakeholders agreement on the DL model training and update processes using smart contract. Furthermore, the blockchain and DL combination was evaluated on a logistic chain incident prediction use case. This evaluation shows the feasibility and the relevancy of this combination.

The above propositions were evaluated using autogenerated, online available and real logistic chains datasets. The evaluation shows promising results for the proposed architecture, the IoT data qualification and the blockchain DL integration proposition.

Based on the evaluations results, the traceability architecture proposed in this thesis is ready to be tested in real life logistic chain scenario. The deployment of the proposed architecture will help the logistic chain stakeholders in their daily struggle for visibility, transparency of the traceability data and processes, through the secure sharing of shipments traceability data, and the automation of the whole traceability data collection and process.

The work achieved in this thesis and its evaluation results open already new opportunities for the development of decentralized, secured, transparent, automated, and intelligent traceability systems. This kind of systems will be a game changer in the logistic chain business collaboration and will greatly improve the traceability process security and transparency among all the logistic chain stakeholders. However, the architecture performance needs to be validated in real life scenario with more stakeholders, more traceability data and real network communication constraints.

It is worth noting that blockchain based traceability architectures adoption in the enterprise context is not merely a simple technology evolution. It is a paradigm change that will generate many challenges related to the user experience and the responsibilities in the blockchain network usage and maintenance. These challenges study and exploration will facilitate the adoption of blockchain based architectures, not only in the logistic domain, but in all the domains in which the blockchain is applicable.

6.2 Perspectives

Although the proposed Blockchain-IoT based traceability architecture is showing promising results on the evaluation, we have identified many areas toward the stakeholder's perfect traceability architecture.

We believe that the data collected in this architecture could be used directly in logistic chain accounting systems for bill generation and inter stakeholders' payment execution. This raises questions about the blockchain based billing systems and the money to be used for payment execution: classical money, standard cryptocurrencies, or a specific logistic chain cryptocurrency. However, this will be a big step forward in the whole logistic chain automation.

Moreover, the real-time data flows management has not been covered in this thesis. It is a very interesting research area, especially for blockchain based architectures, and the growing number of IoT data sources to be handled. To consider the real-time management constraints, the traceability architecture needs to be adapted on each stakeholder side.

Furthermore, the IoT data qualification approach proposed in this thesis covers only some main data quality dimensions, namely: Accuracy, Completeness, Consistency and Currentness. It could be enhanced by including other data quality dimensions such as Credibility, Accessibility, Compliance, Confidentiality and Efficiency. Also, the methods proposed for the IoT data quality dimensions were static. They can be improved using dynamic algorithms such as DBSCAN [40], an assisted version of the Kalman filter [66] or quality learning methods using DL.

Finally, the blockchain-DL integration is a very recent approach in the literature, and the work proposed in this thesis is only a first step in this integration. Therefore, to push forward this integration, model-less learning approaches, such as Reinforcement Learning need to be explored. These approaches do not need to handle a learning model file, and consequently they could present a better blockchain integration option compared to the model-based ones.

Bibliography

[1] Mpr/mib user's manual. URL http://www-db.ics.uci.edu/pages/research/quasar/MPR-MIB%20Series%20User%20Manual%207430-0021-06_A.pdf.

[2] Grafana. URL https://grafana.com/.

[3] Tradelens solution brief. Technical report, IBM. URL https://www.ibm.com/account/reg/signup?formid=urx-42717. Accessed on 17-August-2021.

[4] Influxdb. URL https://www.influxdata.com/products/influxdb/.

[5] *A Study on the Construction of Logistics Information System in Flexible Manufacturing Enterprises*, volume 1, 2010. doi: 10.1109/ISDEA.2010.92.

[6] Blockchain in logistics (dhl and accenture report). Technical report, 2018. URL https://www.logistics.dhl/content/dam/dhl/global/core/documents/pdf/glo-core-blockchain-trend-report.pdf. Accessed: 2021-02-05.

[7] Sigfox and louis vuitton luggage tracking solution, Apr. 2018. URL https://www.sigfox.com/en/news/sigfox-and-louis-vuitton-partner-innovative-luggage-tracker.

[8] *Autonomous services selection system and distributed transportation database(s)*, Aug. 2018. URL https://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&s1=%2220180232693%22.PGNR.&OS=DN/20180232693&RS=DN/20180232693.

[9] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org; accessed on 14-June-2021.

[10] K. Abbas, M. Afaq, T. Ahmed Khan, and W.-C. Song. A blockchain and machine learning-based drug supply chain management and recommendation system for smart pharmaceutical industry. *Electronics*, 9(5), 2020. ISSN 2079-9292. doi: 10.3390/electronics9050852. URL https://www.mdpi.com/2079-9292/9/5/852.

[11] A. Ali, G. A. Shah, M. O. Farooq, and U. Ghani. Technologies and challenges in developing machine-to-machine applications: A survey. *Journal of Network and Computer Applications*, 83:124 – 139, 2017. ISSN 1084-8045. doi: https://doi.org/10.1016/j.jnca.2017.02.002. URL http://www.sciencedirect.com/science/article/pii/S1084804517300620.

[12] A. S. Almasoud, M. M. Eljazzar, and F. Hussain. Toward a self-learned smart contracts. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, pages 269–273, 2018. doi: 10.1109/ICEBE.2018.00051.

[13] A. Andrea. The silk road in world history: A review essay. *The Asian review of World Histories*, 2, 01 2014. doi: 10.12773/arwh.2014.2.1.105.

[14] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. pages 1–15, 2018. doi: 10.1145/3190508.3190538. URL http://arxiv.org/abs/1801.10228. arXiv: 1801.10228.

[15] K. Ashton. That 'internet of things' thing, 2009. URL https://www.rfidjournal.com/that-internet-of-things-thing. Accessed: 2021-02-04.

[16] L. Atzori, A. Iera, and G. Morabito. Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140, 2017. ISSN 1570-8705. doi: https://doi.org/10.1016/j.adhoc.2016.12.004. URL https://www.sciencedirect.com/science/article/pii/S1570870516303316.

[17] R. Azzi, R. K. Chamoun, and M. Sokhn. The power of a blockchain-based supply chain. *Computers & Industrial Engineering*, 135:582–592, 2019. ISSN 0360-8352.

[18] Z. Berman. Outliers rejection in kalman filtering — some new observations. In *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pages 1008–1013, 2014. doi: 10.1109/PLANS.2014.6851466.

[19] T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller. Blockchains everywhere - a use-case of blockchains in the pharma supply-chain. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 772–777, May 2017.

[20] M. L. Brodie. Data quality in information systems. *Information & Management*, 3(6):245–258, 1980. ISSN 0378-7206. doi: https://doi.org/10.1016/0378-7206(80)90035-X. URL https://www.sciencedirect.com/science/article/pii/037872068090035X.

[21] D. Bumblauskas, A. Mann, B. Dugan, and J. Rittmer. A blockchain use case in food distribution: Do you know where your food has been? *International Journal of Information Management*, 52:102008, 2020. ISSN 0268-4012. doi: https://doi.org/10.1016/j.ijinfomgt.2019.09.004. URL http://www.sciencedirect.com/science/article/pii/S026840121930461X.

[22] V. Buterin. Ethereum: A next-generation smart contract and decentralized application platform, 2014. URL https://github.com/ethereum/wiki/wiki/White-Paper. Accessed: 2020-04-28.

[23] V. Buterin. On public and private blockchains, 2014. URL https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/. Accessed: 2021-02-02.

[24] J. Byabazaire, G. O'Hare, and D. Delaney. Data quality and trust: Review of challenges and opportunities for data sharing in iot. *Electronics*, 9(12):2083, Dec 2020. ISSN 2079-9292. doi: 10.3390/electronics9122083. URL http://dx.doi.org/10.3390/electronics9122083.

[25] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda. Blockchain-based traceability in agri-food supply chain management: A practical implementation. In *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*, pages 1–4, May 2018. doi: 10.1109/IOT-TUSCANY.2018.8373021.

[26] R. Casado-Vara, F. de la Prieta, J. Prieto, and J. M. Corchado. Blockchain framework for iot data quality via edge computing. In *Proceedings of the 1st Workshop on Blockchain-Enabled Networked Sensor Systems*, BlockSys'18, page 19–24, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360500. doi: 10.1145/3282278.3282282. URL https://doi.org/10.1145/3282278.3282282.

[27] F. Casino, T. K. Dasaklis, and C. Patsakis. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36:55 – 81, 2019. ISSN 0736-5853. doi: https://doi.org/10.1016/j.tele.2018.11.006. URL http://www.sciencedirect.com/science/article/pii/S0736585318306324.

[28] F. Casino, V. Kanakaris, T. K. Dasaklis, S. Moschuris, and N. P. Rachaniotis. Modeling food supply chain traceability based on blockchain technology. *IFAC-PapersOnLine*, 52 (13):2728 – 2733, 2019. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2019.11.620. URL http://www.sciencedirect.com/science/article/pii/S2405896319316088. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.

[29] M. Castro and B. Liskov. Practical Byzantine fault tolerance. In *USENIX ASSOCIATION PROCEEDINGS OF THE THIRD SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (OSDI '99)*, pages 173–186. USENIX Assoc; IEEE TCOS; ACM SIGOPS, 1999. ISBN 1-880446-39-1. 3rd Symposium on Operating Systems Design and Implementation (OSDI 99), NEW ORLEANS, LA, FEB 22-25, 1999.

[30] R. Chalapathy and S. Chawla. Deep Learning for Anomaly Detection: A Survey. *arXiv e-prints*, art. arXiv:1901.03407, Jan. 2019.

[31] S. E. Chang, Y.-C. Chen, and M.-F. Lu. Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process. *Technological Forecasting and Social Change*, 144:1 – 11, 2019. ISSN 0040-1625. doi: https://doi.org/10.1016/j.techfore.2019.03.015. URL http://www.sciencedirect.com/science/article/pii/S0040162518305547.

[32] F. Chen, H. Wan, H. Cai, and G. Cheng. Machine learning in/for blockchain: Future and challenges. *arXiv*, 2020. URL https://arxiv.org/abs/1909.06189.

[33] J. Cheng, H. Long, X. Tang, J. Li, M. Chen, and N. Xiong. A reputation incentive mechanism of crowd sensing system based on blockchain. *Communications in Computer and Information Science*, 1253 CCIS:695–706, 2020. doi: 10.1007/978-981-15-8086-4_65.

[34] CSCMP. Cscmp supply chain management definitions and glossary, 2013. URL https://cscmp.org/CSCMP/Academia/SCM_Definitions_and_Glossary_of_Terms/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms.aspx?hkey=60879588-f65f-4ab5-8c4b-6878815ef921.

[35] P. Cui, J. Dixon, U. Guin, and D. Dimase. A blockchain-based framework for supply chain provenance. *IEEE Access*, 7:157113–157125, 2019.

[36] I. B. Damgård. A design principle for hash functions. In G. Brassard, editor, *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 416–427, New York, NY, 1990. Springer New York. ISBN 978-0-387-34805-6.

[37] P. Dimitrov and S. Wandel. An international analysis of differences in logistics performance. Iiasa working paper, IIASA, Laxenburg, Austria, April 1988. URL http://pure.iiasa.ac.at/id/eprint/3174/.

[38] A. Ekramifard, H. Amintoosi, A. H. Seno, A. Dehghantanha, and R. M. Parizi. *A Systematic Literature Review of Integration of Blockchain and Artificial Intelligence*, pages 147–160. Springer International Publishing, Cham, 2020. ISBN 978-3-030-38181-3. doi: 10.1007/978-3-030-38181-3_8. URL https://doi.org/10.1007/978-3-030-38181-3_8.

[39] L. Erazo-Garzon, J. Erraez, L. Illescas-Peña, and P. Cedillo. A data quality model for aal systems. In E. Fosenca C, G. Rodríguez Morales, M. Orellana Cordero, M. Botto-Tobar, E. Crespo Martínez, and A. Patiño León, editors, *Information and Communication Technologies of Ecuador (TIC.EC)*, pages 137–152, Cham, 2020. Springer International Publishing. ISBN 978-3-030-35740-5.

[40] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[41] S. Fagúndez, J. Fleitas, and A. Marotta. Data stream quality evaluation for the generation of alarms in the health domain. *Journal of Intelligent Systems*, 24(3):361 – 369, 01 Aug. 2015. doi: https://doi.org/10.1515/jisys-2014-0166. URL https://www.degruyter.com/view/journals/jisys/24/3/article-p361.xml.

[42] J. Fouad, A. Jabir, A. Khaoula, and M. Youssef. Design on improvement of traceability process in the outsourcing of logistics' activities using the internet of things (iot) applications. *International Journal of Advanced Science and Technology*, 29(1):1093 – 1108, Jan. 2020. URL http://sersc.org/journals/index.php/IJAST/article/view/3602.

[43] A. Gharehgozli, H. de Vries, and S. Decrauw. The role of standardisation in european intermodal transportation. *Maritime Business Review*, 4(2):151–168, 2019. doi: 10.1108/MABR-09-2018-0038.

[44] J. Gialelis, G. Theodorou, and C. Paparizos. A low-cost internet of things (iot) node to support traceability: Logistics use case. In *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, GoodTechs '19, page 72–77,

New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362610. doi: 10.1145/3342428.3342661. URL https://doi.org/10.1145/3342428.3342661.

[45] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[46] R. Graf and R. King. Neural network and blockchain based technique for cyber threat intelligence and situational awareness. In *2018 10th International Conference on Cyber Conflict (CyCon)*, pages 409–426, 2018. doi: 10.23919/CYCON.2018.8405028.

[47] X. Gu, J. Peng, W. Yu, Y. Cheng, F. Jiang, X. Zhang, Z. Huang, and L. Cai. Using blockchain to enhance the security of fog-assisted crowdsensing systems. In *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pages 1859–1864, 2019. doi: 10.1109/ISIE.2019.8781332.

[48] L. Hang, I. Ullah, and D.-H. Kim. A secure fish farm platform based on blockchain for agriculture data integrity. *Computers and Electronics in Agriculture*, 170:105251, 2020. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2020.105251. URL http://www.sciencedirect.com/science/article/pii/S016816991932006X.

[49] J. D. Harris and B. Waggoner. Decentralized and collaborative ai on blockchain. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 368–375, 2019. doi: 10.1109/Blockchain.2019.00057.

[50] H. Hasan, E. AlHadhrami, A. AlDhaheri, K. Salah, and R. Jayaraman. Smart contract-based approach for efficient shipment management. *Computers & Industrial Engineering*, 136:149 – 159, 2019. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2019.07.022. URL http://www.sciencedirect.com/science/article/pii/S0360835219304140.

[51] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar. A survey on iot security: Application areas, security threats, and solution architectures. *IEEE Access*, 7:82721–82743, 2019. doi: 10.1109/ACCESS.2019.2924045.

[52] P. Helo and Y. Hao. Blockchains in operations and supply chains: A model and reference implementation. *Computers & Industrial Engineering*, 136:242–251, 2019. ISSN 0360-8352.

[53] J. Hinckeldeyn and K. Jochen. (short paper) developing a smart storage container for a blockchain-based supply chain application. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 97–100, June 2018. doi: 10.1109/CVCBT.2018.00017.

[54] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[55] J. Huang, L. Kong, H. Dai, W. Ding, L. Cheng, G. Chen, X. Jin, and P. Zeng. Blockchain-based mobile crowd sensing in industrial systems. *IEEE Transactions on Industrial Informatics*, 16(10):6553–6563, 2020. doi: 10.1109/TII.2019.2963728.

[56] Z. Hui et al. Reducing building over-cooling by adjusting hvac supply airflow setpoints and providing personal comfort systems. In *The 15th Conference of the International Society of Indoor Air Quality & Climate (ISIAQ), Philadelphia, USA*, 2018. URL https://escholarship.org/uc/item/3164g329.

[57] A. Imeri and D. Khadraoui. The security and traceability of shared information in the process of transportation of dangerous goods. pages 1–5, 02 2018. doi: 10.1109/NTMS.2018.8328751.

[58] ISO. Iso 9000:2005(en) quality management systems — fundamentals and vocabulary, 2005. URL https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-3:v1:en. Accessed: 2021-03-10.

[59] ISO. Iso 28000:2007(en) specification for security management systems for the supply chain, 2007. URL https://www.iso.org/obp/ui/#iso:std:iso:28000:ed-1:v1:en. Accessed: 2021-03-10.

[60] ISO. Iso/ts 17451-2:2017(en) packaging — codification of contents for inventories for shipments of household goods and personal effects — part 2: Xml messaging structure for electronic transmission of inventory data, 2017. URL https://www.iso.org/obp/ui/#iso:std:iso:ts:17451:-2:ed-1:v1:en. Accessed: 2021-03-10.

[61] ISO-25012. Iso 25012 : Quality of data product, 2008. URL https://iso25000.com/index.php/en/iso-25000-standards/iso-25012. Accessed: 2021-02-04.

[62] Y. Issaoui, A. Khiat, A. Bahnasse, and H. Ouajji. Smart logistics: Study of the application of blockchain technology. *Procedia Computer Science*, 160:266–271, 2019. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2019.09.467. URL https://www.sciencedirect.com/science/article/pii/S1877050919316825. The 10th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2019) / The 9th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2019) / Affiliated Workshops.

[63] F. Jameel, U. Javaid, W. U. Khan, M. N. Aman, H. Pervaiz, and R. Jäntti. Reinforcement learning in blockchain-enabled iiot networks: A survey of recent advances and open challenges. *Sustainability*, 12(12), 2020. ISSN 2071-1050. doi: 10.3390/su12125161. URL https://www.mdpi.com/2071-1050/12/12/5161.

[64] G. C. James Macaulay, Lauren Buckalew. Internet of things in logistics. Technical report, DHL and CISCO, 2015. URL https://www.dhl.com/content/dam/Local_Images/g0/New_aboutus/innovation/DHLTrendReport_Internet_of_things.pdf.

[65] A. Javaid, M. Zahid, I. Ali, R. Khan, Z. Noshad, and N. Javaid. Reputation system for iot data monetization using blockchain. 08 2019.

[66] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. ISSN 0021-9223. doi: 10.1115/1.3662552. URL https://doi.org/10.1115/1.3662552.

[67] M. Kara, O. Lamouchi, and A. Ramdane-Cherif. A quality model for the evaluation aal systems. *Procedia Computer Science*, 113:392 – 399, 2017. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2017.08.354. URL http://www.sciencedirect.com/science/article/pii/S1877050917317647. The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017) / The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2017) / Affiliated Workshops.

[68] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel. A model-driven architecture-based data quality management framework for the internet of things. In *2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech)*, pages 252–259. IEEE, 2016. ISBN 9781467388948.

[69] A. Karkouch, H. Mousannif, H. Al-Moatassime, and T. Noël. Data quality in internet of things : A state-of-the-art survey. *Journal of Network and Computer Applications*, 2016.

[70] Y. Kayikci. Sustainability impact of digitization in logistics. *Procedia Manufacturing*, 21:782–789, 2018. ISSN 2351-9789. doi: https://doi.org/10.1016/j.promfg.2018.02.184. URL https://www.sciencedirect.com/science/article/pii/S2351978918302245. 15th Global Conference on Sustainable Manufacturing.

[71] O. Kelly, B. Mic, M. James, A. Shawn, M. Dan, and M. Cian. Sawtooth: An introduction, 01 2018. URL https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger_Sawtooth_WhitePaper.pdf. [Online; accessed on 09-May-2021].

[72] P. W. Khan, Y.-C. Byun, and N. Park. Iot-blockchain enabled optimized provenance system for food industry 4.0 using advanced deep learning. *Sensors*, 20(10), 2020. ISSN 1424-8220. doi: 10.3390/s20102990. URL https://www.mdpi.com/1424-8220/20/10/2990.

[73] H. Kim, S.-H. Kim, J. Y. Hwang, and C. Seo. Efficient privacy-preserving machine learning for blockchain network. *IEEE Access*, 7:136481–136495, 2019. doi: 10.1109/ACCESS.2019.2940052.

[74] K. Kolomvatsos. A distributed, proactive intelligent scheme for securing quality in large scale data processing. *Computing*, 101(11):1687–1710, 2019. ISSN 0010-485X.

[75] W. Kratsch, J. Manderscheid, M. Röglinger, and J. Seyfried. Machine learning in business process monitoring: A comparison of deep learning and classical approaches used for outcome prediction. *Business and Information Systems Engineering*, 2020. ISSN 23637005. doi: 10.1007/s12599-020-00645-0. URL https://link.springer.com/article/10.1007/s12599-020-00645-0.

[76] N. Kshetri. 1 blockchain's roles in meeting key supply chain management objectives. *International Journal of Information Management*, 39:80, 2018. ISSN 02684012. URL http://search.proquest.com/docview/2059164679/.

[77] D. Kuemper, T. Iggena, R. Toenjes, and E. Pulvermueller. Valid.iot: A framework for sensor data quality analysis and interpolation. In *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, page 294–303, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450351928. doi: 10.1145/3204949.3204972. URL https://doi.org/10.1145/3204949.3204972.

[78] A. B. Kurtulmus and K. Daniel. Trustless machine learning contracts; evaluating and exchanging machine learning models on the ethereum blockchain. *CoRR*, abs/1802.10185, 2018. URL http://arxiv.org/abs/1802.10185.

[79] Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. *Handwritten Digit Recognition with a Back-Propagation Network*, page 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1558601007.

[80] F. Leal, A. E. Chis, S. Caton, H. González–Vélez, J. M. García–Gómez, M. Durá, A. Sánchez–García, C. Sáez, A. Karageorgos, V. C. Gerogiannis, A. Xenakis, E. Lallas, T. Ntounas, E. Vasileiou, G. Mountzouris, B. Otti, P. Pucci, R. Papini, D. Cerrai, and M. Mier. Smart pharmaceutical manufacturing: Ensuring end-to-end traceability and data integrity in medicine production. *Big Data Research*, 24:100172, 2021. ISSN 2214-5796. doi: https://doi.org/10.1016/j.bdr.2020.100172. URL https://www.sciencedirect.com/science/article/pii/S221457962030040X.

[81] Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang. Aimq: a methodology for information quality assessment. *Information & Management*, 40(2):133–146, 2002. ISSN 0378-7206.

[82] F. Li, S. Nastic, and S. Dustdar. Data quality observation in pervasive environments. In *2012 IEEE 15th International Conference on Computational Science and Engineering*, pages 602–609, 2012.

[83] X. Li, L. Yang, Y. Duan, Z. Wu, and X. Zhang. Developing a real-time monitoring traceability system for cold chain of tricholoma matsutake. *Electronics*, 8(4):423, Apr 2019. ISSN 2079-9292. doi: 10.3390/electronics8040423. URL http://dx.doi.org/10.3390/electronics8040423.

[84] Q. Lin, H. Wang, X. Pei, and J. Wang. Food safety traceability system based on blockchain and epcis. *IEEE Access*, 7:20698–20707, 2019.

[85] C. Liu, P. Nitschke, S. Williams, and D. Zowghi. Data quality and the internet of things. *Computing*, 07 2019. doi: 10.1007/s00607-019-00746-z.

[86] G.-X. Liu and F. Liu. Iot-based tpl whole supply chain logistics information system model. In *2013 International Conference on Machine Learning and Cybernetics*, volume 04, pages 1758–1762, July 2013. doi: 10.1109/ICMLC.2013.6890882.

[87] C. Loebbecke and P. Powell. Competitive advantage from it in logistics: The integrated transport tracking system. *International Journal of Information Management*, 18(1):17–27, 1998. ISSN 0268-4012. doi: https://doi.org/10.1016/S0268-4012(97)00037-6. URL https://www.sciencedirect.com/science/article/pii/S0268401297000376.

[88] S. Madden. Intel lab data. URL http://db.csail.mit.edu/labdata/labdata.html.

[89] D. Mao, F. Wang, Z. Hao, and H. Li. Credit evaluation system based on blockchain for multiple stakeholders in the food supply chain. *International journal of environmental research and public health*, 15(8):1627, Aug 2018. ISSN 1660-4601. doi: 10.3390/ijerph15081627. URL https://pubmed.ncbi.nlm.nih.gov/30071695.

[90] I. P. S. Mary and L. Arockiam. Imputing the missing data in iot based on the spatial and temporal correlation. In *2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, pages 1–4, 2017.

[91] F. Masood and A. R. Faridi. Consensus algorithms in distributed ledger technology for open environment. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–6, 2018. doi: 10.1109/CCAA.2018.8777695.

[92] A. J. Mayo and N. Nohria. The truck driver who reinvented shipping, 2005. URL https://hbswk.hbs.edu/item/5026.html.

[93] W. McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[94] V. Menger, F. Scheepers, and M. Spruit. Comparing deep learning and classical machine learning approaches for predicting inpatient violence incidents from clinical text. *Applied Sciences*, 8(6), 2018. ISSN 2076-3417. doi: 10.3390/app8060981. URL https://www.mdpi.com/2076-3417/8/6/981.

[95] L. Mika. Smartlog: Piloting blockchain for logistics, 12 2017. URL https://wpassets.porttechnology.org/wp-content/uploads/2019/05/25183554/036-038_3.pdf.

[96] A. Milne. The rise and success of the barcode: Some lessons for financial services. *Journal of Banking Regulation*, 14, 02 2013. doi: 10.2139/ssrn.2238134.

[97] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking. A review on internet of things (iot), internet of everything (ioe) and internet of nano things (iont). In *2015 Internet Technologies and Applications (ITA)*, pages 219–224, 2015. doi: 10.1109/ITechA.2015.7317398.

[98] W. Müller. *The elements of the science of war; containing the modern, established, and approved principles of the theory and practice of the military sciences ... Illustrated by seventy-five plates, on artillery, fortification, &c. and remarkable battles fought since the year 1675, for the use of military schools and self-instruction ... By William Müller.* Longman, Hurst, Rees, Orme and co. [etc.], London, 1811.

[99] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *BITCOIN*, 2008. URL https://bitcoin.org/bitcoin.pdf.

[100] D. Nguyen and M. I. Ali. Enabling on-demand decentralized iot collectability marketplace using blockchain and crowdsensing. In *2019 Global IoT Summit (GIoTS)*, pages 1–6, 2019. doi: 10.1109/GIOTS.2019.8766346.

[101] D. Ongaro and J. Ousterhout. In search of an understandable consensus algorithm. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pages 305–319, Philadelphia, PA, June 2014. USENIX Association. ISBN 978-1-931971-10-2. URL https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro.

[102] E. D. Pascale, J. McMenamy, I. Macaluso, and L. Doyle. Smart contract slas for dense small-cell-as-a-service. *CoRR*, abs/1703.04502, 2017. URL http://arxiv.org/abs/1703.04502.

[103] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[104] W. Peter and L. Felicity. Fsa orders tests of processed beef after horsemeat is found in findus lasagne. *The Guardian*. URL https://www.theguardian.com/world/2013/feb/08/fsa-tests-horsemeat-lasagne.

[105] F. Piccialli, V. D. Somma, F. Giampaolo, S. Cuomo, and G. Fortino. A survey on deep learning in medicine: Why, how and when? *Information Fusion*, 66:111–137, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2020.09.006. URL https://www.sciencedirect.com/science/article/pii/S1566253520303651.

[106] M. Pournader, Y. Shi, S. Seuring, and S. L. Koh. Blockchain applications in supply chains, transport and logistics: a systematic review of the literature. *International Journal of Production Research*, 58(7):2063–2081, 2020. doi: 10.1080/00207543.2019.1650976. URL https://doi.org/10.1080/00207543.2019.1650976.

[107] S. Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv*, 2020. URL https://arxiv.org/abs/1811.12808.

[108] G. Rempel. Defining standards for web page performance in business applications. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, ICPE '15, page 245–252, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450332484. doi: 10.1145/2668930.2688056. URL https://doi.org/10.1145/2668930.2688056.

[109] D. M. S. Richard Y. Wang. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12, 1996. doi: Vol.12,No.4(Spring, 1996),pp.5-33.

[110] D. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[111] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar. Blockchain-based soybean traceability in agricultural supply chain. *IEEE Access*, 7:73295–73305, 2019. ISSN 2169-3536.

[112] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha. Blockchain for ai: Review and open research challenges. *IEEE Access*, 7:10127–10149, 2019. doi: 10.1109/ACCESS.2018.2890507.

[113] Z. Shahbazi and Y.-C. Byun. A procedure for tracing supply chains for perishable food based on blockchain, machine learning and fuzzy logic. *Electronics*, 10(1), 2021. ISSN 2079-9292. doi: 10.3390/electronics10010041. URL https://www.mdpi.com/2079-9292/10/1/41.

[114] A. Shanley. Could blockchain improve pharmaceutical supply chain security? *Pharmaceutical Technology*, pages S34–S39, 2017. ISSN 15432521. URL http://search.proquest.com/docview/1934901755/.

[115] S. Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini. A secure and quality-aware prototypical architecture for the internet of things. *Information Systems*, 58:43 – 55, 2016. ISSN 0306-4379. doi: https://doi.org/10.1016/j.is.2016.02.003. URL http://www.sciencedirect.com/science/article/pii/S0306437916300072.

[116] G. J. Simmons. Symmetric and asymmetric encryption. *ACM Comput. Surv.*, 11(4): 305–330, Dec. 1979. ISSN 0360-0300. doi: 10.1145/356789.356793. URL https://doi.org/10.1145/356789.356793.

[117] M. Staples, S. Chen, S. Falamaki, A. Ponomarev, P. Rimba, A. B. Tran, I. Weber, X. Xu, and J. Zhu. Risks and opportunities for systems using blockchain and smart contracts. 06 2017. doi: 10.4225/08/596e5ab7917bc.

[118] G. Suciu, C. Nădrag, C. Istrate, A. Vulpe, M. Ditu, and O. Subea. Comparative analysis of distributed ledger technologies. In *2018 Global Wireless Summit (GWS)*, pages 370–373, 2018. doi: 10.1109/GWS.2018.8686563.

[119] K. Sunny and N. Scott. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Technical report, 2012.

[120] N. Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997. ISSN 13960466. doi: 10.5210/fm.v2i9.548. URL https://ojphi.org/ojs/index.php/fm/article/view/548.

[121] S. Underwood. Blockchain beyond bitcoin. *Commun. ACM*, 59(11):15–17, Oct. 2016. ISSN 0001-0782. doi: 10.1145/2994581. URL https://doi.org/10.1145/2994581.

[122] A. M. Vartouni, S. S. Kashi, and M. Teshnehlab. An anomaly detection method to detect web attacks using stacked auto-encoder. In *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pages 131–134, 2018. doi: 10.1109/CFIS.2018.8336654.

[123] B. Violino. The history of rfid technology, 2005. URL https://www.rfidjournal.com/the-history-of-rfid-technology.

[124] T. Wang. A unified analytical framework for trustable machine learning and automation running with blockchain. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4974–4983, 2018. doi: 10.1109/BigData.2018.8622262.

[125] T. Wang, M. Du, X. Wu, and T. He. An analytical framework for trusted machine learning and computer vision running with blockchain. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 32–38, 2020. doi: 10.1109/CVPRW50498.2020.00011.

[126] S. Wattanakul, S. Henry, M. L. Bentaha, N. Reeveerakul, and Y. Ouzrout. Improving risk management by using smart containers for real-time traceability. In *9th International Conference on Logistics and Transport (ICLT 2017)*, Bangkok, Thailand, Nov. 2017. URL https://hal.archives-ouvertes.fr/hal-01722866.

[127] L. Wei, J. Wu, and C. Long. A blockchain-based hybrid incentive model for crowdsensing. *Electronics*, 9(2), 2020. ISSN 2079-9292. doi: 10.3390/electronics9020215. URL https://www.mdpi.com/2079-9292/9/2/215.

[128] Q. Wen, Y. Gao, Z. Chen, and D. Wu. A blockchain-based data sharing scheme in the supply chain by iiot. In *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, pages 695–700, May 2019. doi: 10.1109/ICPHYS.2019.8780161.

[129] M. Westerkamp, F. Victor, and A. Küpper. Blockchain-based supply chain traceability: Token recipes model manufacturing processes. In *2018 IEEE 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)*, pages 1595–1602, 2018.

[130] A. Whitmore, A. Agarwal, and L. Da Xu. The internet of things—a survey of topics and trends. *Information Systems Frontiers*, 17, Apr. 2014. doi: 10.1007/s10796-014-9489-2.

[131] B. Yong, J. Shen, X. Liu, F. Li, H. Chen, and Q. Zhou. An intelligent blockchain-based system for safe vaccine supply and supervision. *International Journal of Information Management*, 52, 2020. ISSN 0268-4012.

[132] P. Yuan, K. Zheng, X. Xiong, K. Zhang, and L. Lei. Performance modeling and analysis of a hyperledger-based system using gspn. *Computer Communications*, 153:117 – 124, 2020. ISSN 0140-3664. doi: https://doi.org/10.1016/j.comcom.2020.01.073. URL http://www.sciencedirect.com/science/article/pii/S0140366419306474.

[133] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar. Efficient gan-based anomaly detection. *CoRR*, abs/1802.06222, 2018. URL http://arxiv.org/abs/1802.06222.

[134] W. Zhang, G. Yang, Y. Lin, C. Ji, and M. M. Gupta. On definition of deep learning. In *2018 World Automation Congress (WAC)*, pages 1–5, 2018. doi: 10.23919/WAC.2018.8430387.

[135] R. Zhou, S. Shao, W. Li, and L. Zhou. How to define the user's tolerance of response time in using mobile applications. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 281–285, 2016. doi: 10.1109/IEEM.2016.7797881.

[136] S. Zou, J. Xi, H. Wang, and G. Xu. Crowdblps: A blockchain-based location-privacy-preserving mobile crowdsensing system. *IEEE Transactions on Industrial Informatics*, 16 (6):4206–4218, 2020. doi: 10.1109/TII.2019.2957791.

ECOLE DOCTORALE

Titre : Amélioration de la traçabilité des chaînes logistiques B2B à l'aide de la Blockchain, l'IoT et le Deep Learning

Mots clés : Chaîne logistique, Traçabilité, Blockchain, Internet des objets, Deep Learning

Résumé : Les systèmes d'information des entreprises connaissent aujourd'hui une évolution rapide. Dans le contexte de la chaine logistique, cette évolution est marquée par l'introduction des nouvelles technologies comme l'Internet des Objets. Puisque la chaine logistique implique plusieurs intervenants, elle exige le partage des données entre les intervenants pour assurer la traçabilité des produits tout au long de la chaine logistique. Les systèmes de traçabilité traditionnels sont centralisés et ne garantissent pas le partage sécurisé des données et l'accord des intervenants sur les données partagées et leurs règles de traitement. Plusieurs travaux ont été proposés dans la littérature en utilisant la blockchain pour surmonter les problèmes susmentionnés. L'objectif principal de cette thèse est d'aller au-delà de l'état de l'art actuel et de proposer une architecture de traçabilité basée sur la blockchain et l'Internet des Objets et adaptée aux chaines logistiques B2B. Par ailleurs, la qualité des données de l'Internet des Objets est un frein au développement de ce type d'architecture de traçabilité. Pour surmonter ce problème, et assurer la confiance des intervenants dans les données collectées et faciliter l'automatisation du processus de collection des données de traçabilité, l'architecture proposée inclut un module de qualification des données de l'Internet des Objets. Ce module fournit aux intervenants des données de haute qualité et un contrôle et suivi fins de la qualité des données basés sur les exigences qualité des intervenants. De plus, l'Internet des Objets génère un volume important de données et pour assurer un traitement efficace et intelligent de cet important volume de données, l'architecture proposée est renforcée avec des capacités d'apprentissage en utilisant l'apprentissage renforcé. En outre, toutes les propositions de la thèse ont été évaluées et leurs évaluations montrent des résultats prometteurs pour le déploiement de l'architecture de traçabilité proposée dans la chaine logistique pour aider les intervenants dans leur lutte quotidienne pour la traçabilité.

Title : Enhancing the traceability of B2B logistic chains using blockchain, IoT and Deep Learning

Keywords : Logistic chain, Traceability, Blockchain, Internet of Things, Deep Learning

Abstract : Nowadays, company information systems are witnessing a very fast evolution. In the logistic chain context, this fast evolution is characterized by the introduction of new technologies such as the Internet of Things. Since the logistic chain involves multiples stakeholders, it requires data sharing among all these stakeholders to ensure products traceability in the whole logistic chain. Traditional traceability systems are used by the stakeholders for traceability data sharing. However, these traditional systems are centralized and do not guarantee the secure sharing of data and the stakeholders agreement on the shared data and its processing rules. Many works have been proposed in the literature using blockchain to overcome the above issues. The main objective of this thesis is to go beyond the current state of the art and propose a blockchain-IoT based traceability architecture adapted to the B2B logistic chain context. In addition, the IoT data quality is a hindrance to the development of this kind of traceability architectures. To overcome this issue and ensure the stakeholders trust in the collected data and facilitate the automation of the traceability data collection process, the proposed architecture includes an IoT data qualification module providing the stakeholders with high data quality and fine data quality control and monitoring based on the stakeholders quality requirements. Moreover, the IoT generates a huge data volume and to ensure an efficient and intelligent data management of this huge data volume, the proposed architecture is boosted with learning capabilities using Deep Learning. Furthermore, all the thesis propositions have been evaluated and their evaluation shows promising results for the deployment of the proposed traceability architecture in the logistic chain to help the stakeholders in their traceability daily life struggle.

Institut Polytechnique de Paris
91120 Palaiseau, France