Trust mechanisms for connectivity service assurance on multi- actor infrastructures

Vincent Messié

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE MINES-TÉLÉCOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Informatique

Par

Vincent MESSIÉ

Trust mechanisms for connectivity service assurance on multi-actor infrastructures

Thèse présentée et soutenue à Brest, le 18/11/2022
Unité de recherche : Lab-STICC, IMT Atlantique et Orange Innovation
Thèse N° : 2022IMTA0316

Rapporteurs avant soutenance :

Pr. Stefano SECCI	Professeur, Conservatoire National des Arts et Métiers
Dr. Hassnaa MOUSTAFA	Principal Engineer, Intel, USA – HDR

Composition du Jury :

Président :	Pr. Jean-Paul DELAHAYE	Professeur émérite, Université de Lille
Examinateurs :	Pr. Stefano SECCI	Professeur, Conservatoire National des Arts et Métiers
	Dr. Hassnaa MOUSTAFA	Principal Engineer, Intel, USA – HDR
	Dr. Quentin BRAMAS	Maître de conférences, Université de Strasbourg
	Dr. Isabel AMIGO	Maître de conférences, IMT Atlantique
Dir. de thèse :	Pr. Sandrine VATON	Professeure, IMT Atlantique

Invité(s) :

Dr. Benoit RADIER	Ingénieur de Recherche, Orange	*Co-encadrant de thèse*
M. Jérôme PONS	Ingénieur de Recherche, Orange	
M. Gaël FROMENTOUX	Ingénieur de Recherche, Orange	*Co-dir. de thèse*

Acknowledgement

First and foremost, I wish to thank Orange Innovation for giving me the opportunity of doing this thesis work and funding it. I wish to thank more particularly Benoit Radier and Gaël Fromentoux for supervising my work on Orange side. This work hasn't always been easy, yet thank your for being supportive! On Orange side, I'd also like to express my gratitude toward Xavier Marjou and Tangui le Gléau, whose work on collaborative models for telco-related use-cases supported and complemented mine. Many other individuals helped me during my thesis, including but not limited to Nathalie Labidurie, Luc le Beller, Veronica Quintuna Rodriguez, Pierre-Yves le Lann, Yvon Gourhant, Stéphane Tuffin... I also extend my thanks to the ARC and OSONS teams, whose members where always happy to help when I needed it.

I also want to thank Sandrine Vaton and Isabel Amigo, on IMT Atlantique side, for supervising my thesis work, and for the help. I also express my gratitude to Alexandre Reiffers-Masson for his good advices on DAG-based ledgers.

Finally, on a more personal note, I wish to thank all the people, friends whom I met as part of extra-professional activities, who were supportive toward me during these three years.

Résumé en français

Contexte

Le réseau Internet est aujourd'hui devenu un élément indispensable de notre quotidien, et son accès est maintenant considéré comme un droit de l'homme fondamental par les Nations Unis [1]. Il est également à noter que dans le futur, de nouveaux usages des réseaux vont arriver, à l'instar des drones, voitures connectées, de l'Internet des Objets, etc. Les réseaux devront ainsi être développés pour accompagner le développement de tous ces nouveaux besoins. De manière plus spécifique, les opérateurs devront déployer de nouvelles infrastructures pour répondre à la demande, et doivent innover pour réduire non seulement les coûts de déploiement, mais également la consommation de ressources de ces nouvelles infrastructures.

Dans ce contexte, le partage des infrastructures réseaux entre plusieurs acteurs de la connectivité est pertinent afin de pouvoir non seulement construire de nouvelles infrastructures réseaux plus efficaces, mais également pour permettre de réduire les coûts pour tous. Ainsi, par exemple un opérateur ayant besoin d'utiliser une infrastructure ne lui appartenant pas pourrait, grâce à des mécanismes de collaboration, utiliser une infrastructure fournie par un autre opérateur et sous-utilisée par ce dernier [2]. De plus, la virtualisation des infrastructures réseaux permet maintenant de dynamiser les échanges de ressources entre acteurs de la connectivité. En effet, une infrastructure réseau virtuelle ne devient composée que de briques logicielles déployées sur des infrastructures physiques banalisées (par exemple de type Cloud), et son déploiement et cycle de vie peut être complètement automatisé, notamment grâce à des initiatives comme l'Open Network Automation Platform (ONAP) [3]. Ainsi, une telle infrastructure peut être facilement partagée entre plusieurs acteurs, certains fournissant des infrastructures physiques (antennes mobiles, infrastructures Cloud, réseaux optiques, etc.) et d'autres déployant des services réseaux virtualisés grâce à ces infrastructures partagées.

Problématique & Objectifs

Cependant, pour que de tels services de connectivité puissent voir le jour, il est notamment nécessaire de pouvoir assurer une Qualité de Service (QoS) pour les services réseaux virtualisés déployés sur une telle infrastructure. Dans le cas d'une infrastructure fournie par plusieurs acteurs, il est indispensable de garantir le bon fonctionnement de l'intégralité de l'infrastructure réseau, pour que des services de connectivité de bout en bout puissent être déployés dessus. Pour cela, il est nécessaire de pouvoir collecter des données de performance fiables depuis l'infrastructure de chaque acteur impliqué dans la collaboration, et que chaque acteur puisse y avoir confiance.

Ce travail de thèse apporte des éléments de réponses aux questions suivantes:

* Qu'est-ce qui peut être fait sur l'infrastructure réseau pour faciliter la collaboration?
* Quels sont les nouveaux scénarios de collaboration possibles maintenant?
* Comment une source de données fiable pourrait-elle être implémentée de manière décentralisée? Quelle serait son architecture haut niveau, indépendemment du cas d'usage associé? Est-ce qu'un registre distribué pourrait être utilisé dans ce contexte?
* En considérant l'utilisation d'un registre distribué dans ce contexte, quelle technologie serait la plus intéressante dans ce cas? Est-ce que la technologie elle-même pourrait être améliorée pour mieux supporter un tel cas d'usage?
* Quels mécanismes pourraient être implémentés par des opérateurs impliqués dans un service de connectivité multi-acteurs, afin de sécuriser et fiabiliser des données opérationnelles?

Contributions de la thèse

Nouvelles opportunités de collaboration dans les réseaux

Pour répondre à ces questions, plusieurs travaux ont été effectués lors de ce travail de thèse. Tout d'abord, dans le Chapitre 2 de ce manuscrit sont présentés des travaux autour des opportunités de collaboration autour du monde des télécommunications. Dans ce chapitre, des solutions pour améliorer l'infrastructure afin de mieux supporter la collaboration sont explorées. Cette thèse m'a déjà permis de participer à l'élaboration d'un brevet portant sur un procédé permettant l'optimisation d'une infrastructure réseau supportant un environnement collaboratif [4]. L'architecture de collaboration considérée est celle définie par l'Industrial DataSpace Association (IDSA) [5]. J'ai également pu proposer un autre brevet proposant un protocole de temps décentralisé [6]. Le procédé proposé permet notamment à des acteurs impliqués dans une collaboration d'avoir confiance dans l'horodatage des données échangées, et ce sans avoir à recourir à un tiers de confiance unique. Dans ce même chapitre sont également présentés des travaux auxquels j'ai participé, définissant des scénarios de collaboration entre acteurs du marché de la

connectivité. D'abord, un scénario de partage d'infrastructure entre opérateurs mobiles a été évoqué, et modélisé grâce à des algorithmes d'apprentissage par renforcement multi-acteurs [2]. Ces travaux ont notamment pu montrer que des opérateurs mobiles peuvent réussir à collaborer et partager leurs infrastructures pour économiser de l'énergie, à la condition qu'ils soient aidés par un environnement de collaboration pilotant le partage. Ce même environnement a besoin de données fiables sur la consommation d'énergie des infrastructures des opérateurs mobiles, afin d'optimiser au mieux le partage de ressources. J'ai également pu participer à une autre contribution dans le cadre d'une coopération internationale, dans un Catalyst TMF [7]. Ces travaux ont permis de décrire l'architecture générale de ce que serait une "place de marché de la connectivité" permettant à des opérateurs de déployer des services réseaux virtualisés sur les infrastructures de plusieurs acteurs différents. Là encore, ces travaux ont montré la nécessité d'être en mesure de fournir des données fiables depuis une infrastructure de réseau. Ces contributions ont donc surtout permis de souligner ce besoin d'informations de performance fiables pour qu'il puisse exister une collaboration entre différents acteurs de la connectivité.

De la technologie des registres distribués

Le troisième chapitre de ce manuscrit se concentre ensuite sur la technologie des registres distribués ("Distributed Ledger Technology", DLT). L'étude de cette technologie dans le contexte de cette thèse est motivée par la capacité des registres distribués à pouvoir stocker des informations irréfutables dans des bases de données complètement distribuées, sans avoir à recourir à un tiers de confiance. La pertinence de cette technologie pour stocker des informations de performance irréfutables a été évaluée, notamment au regard des performances de la technologie. Il faut savoir que la Blockchain, la technologie historique de registre distribué [8] est souvent décriée pour ses performances médiocres et sa consommation de ressources jugée excessive. Ainsi, de nombreuses innovations ont vu le jour pour améliorer les performances et l'efficacité des registres distribués [9, 10]. Parmi ces innovations, une nouvelle génération de registres distribués à base de graphes acycliques orientés a vu le jour. Ces registres ne sont plus limités à une chaîne de blocs ne croissant que dans une seule dimension, mais stockent les données dans des structures à base de graphe. De telles structures permettent ainsi de meilleures performances. Cela rend ce type de registre intéressant pour des cas d'usage nécessitant le stockage d'un volume important de données, et avec une consommation de ressources minimisée. Les travaux de cette thèse ont notamment permis de valider le choix du Tangle, une technologie de registre distribué à base de graphes acycliques [11] pour stocker des informations de performance depuis une infrastructure réseau multi-acteur [12]. Pour cela, des simulations du Tangle ont été effectuées durant ces travaux de thèse, afin de valider ce choix, et le résultat de ces travaux a pu être publié dans une conférence [12]. Durant ces travaux de thèse, un brevet a également pu être proposé, permettant de réduire l'impact d'un registre distribué sur les capacités de stockage des

machines qui le supporte. En effet, la capacité de stockage requise par un registre distribué ne fait que croître dans le temps de par le fonctionnement des registres distribués où les données ne peuvent être que rajoutées. Cela peut par exemple poser problème pour des nœuds participant à un registre distribué, ayant une capacité de stockage limitée tout en manquant de confiance pour déléguer le stockage du registre distribué à d'autre acteurs. Le procédé proposé permet alors à un ensemble d'acteurs utilisant un registre distribué de choisir eux-même un ou plusieurs "archivistes" de confiance pour stocker le contenu du registre distribué utilisé. Le procédé proposé permet plus particulièrement la sécurisation des données stockées par ces archivistes grâce à des preuves cryptographiques stockées sur le registre distribué. Le procédé permet ainsi à des noeuds ayant des capacités de stockage limitées de participer à n'importe quel registre distribué, sans perte de confiance liée au stockage des données historiques. Ces travaux ont pu faire l'objet d'un brevet [13].

Utilisation des registres distribués pour permettre la collecte de données de performance fiables

Le quatrième chapitre de ce manuscrit décrit les contributions de ce travail de thèse sur des architectures permettant la collecte de données de performance fiables. Tout d'abord, une solution permettant la collecte d'informations d'usage d'un chemin réseau composé de plusieurs acteurs a été imaginé en début de thèse. Ce procédé consiste en l'échange, à intervalle de temps régulier, d'une trame contenant des informations d'usage du réseau. Cette trame effectue un aller-retour sur le chemin réseau, passant alors par toutes les infrastructures des différents acteurs le composant. Elle est signée par tous les acteurs dans les deux sens. Cette dernière est ensuite stockée sur un registre distribué, afin de la rendre irréfutable. Cet enregistrement contient alors une preuve irréfutable de l'usage qui a été fait du chemin réseau, ainsi que de son approbation par l'ensemble des acteurs impliqués.

Cette contribution a déjà pu faire l'objet d'un brevet [14], et d'une publication dans une conférence [15]. Par la suite, une étude de l'implémentation d'une telle solution sur un registre distribué a été réalisée dans ce travail de thèse. L'étude a surtout porté sur les performances de la technologie des registres distribués pour traiter un tel cas d'usage. En effet, en considérant 45000 chemins réseaux simultanément actifs (chiffres du Wi-Fi public [16]), chacun émettant un rapport d'usage toutes les 8 secondes, le taux de transactions (données à traiter, ici des rapports d'usage) par seconde atteint environ 5600, alors que des solutions historiques de registres distribués à base de Blockchain comme le Bitcoin [8] n'atteignent que 5 transactions par seconde. Des simulations du cas d'usage présenté ont également montré qu'un point de congestion pouvait être atteint, empêchant un registre distribué de fonctionner correctement. Ces travaux ont ensuite permis de définir une architecture où plusieurs instances de registres distribués sont déployées pour mitiger l'impact d'un volume important de données. Cette approche, appelée "*Sharding*", est également

utilisée par d'autres technologies de registres distribués [17]. Ces travaux complémentaires ont pu être présentés dans un article de revue [18].

Finalement, une architecture dite de "*data layer*" ou "couche de données" a été présentée durant ce travail de thèse. Cette proposition est la contribution principale de cette thèse. Cette solution considère le cas d'usage d'une place de marché de la connectivité telle que décrite dans le chapitre 2 de cette thèse [7], et vise à fournir à la place de marché des indicateurs de performance fiables, à partir des infrastructures des acteurs participant aux services réseaux virtualisés de bout en bout. Pour cela, cette architecture est basée autour d'un registre distribué stockant les données de performance des services de connectivité créés grâce à la place de marché, et est implémentée de la façon suivante:

* tout d'abord, des agents virtualisés sont déployés sur les infrastructures des acteurs composants un service de bout en bout donné, à son instanciation. Ces agents ont ensuite la charge de collecter des informations de télémétrie à partir d'agents de supervision pré-existants (comme des sondes Simple Network Management Protocol (SNMP), Netflow, etc.).
* À la collecte de ces métriques, ces agents vont contrôler ces métriques à partir de règles qu'ils auront pré-établies entre eux à l'établissement du service. Ces contrôles peuvent impliquer non seulement l'authentification des agents de supervision eux-mêmes s'ils implémentent des logiciels certifiés, mais également des règles de vérification propres aux métriques elle-mêmes. Toutes ces vérifications sont implémentées à l'aide d'un contrat intelligent déployé au préalable sur le registre distribué, de façon à les rendre transparentes et irréfutables pour l'ensemble des acteurs impliqués dans le service réseau.
* Si les vérifications sont concluantes, les métriques contrôlées sont alors stockées sur le registre distribué, ce qui a pour effet de les rendre irréfutables, ainsi que leur traitement. À ce stade, les métriques peuvent être considérées comme étant fiables par l'intégralité des acteurs participant au service de connectivité associé, car elles sont validées avec des règles approuvées par tous, et leur traitement devient irréfutable grâce au registre distribué.
* Ainsi, ces métriques de performance stockées sur le registre sont ensuite interceptées par des agents déployés par les acteurs sur un environnement garantissant de meilleures performances de calcul. À ce stade, les métriques sont ensuite agrégées en Indicateurs de Performance Clés (*Key Performance Indicators*, KPIs), c'est à dire des données directement exploitables par la place de marché comme informations fiables provenant de l'infrastructure réseau. Le calcul de ces KPIs est également à convenir entre les acteurs composant le service de connectivité, et est également implémenté comme contrat intelligent sur le registre distribué de la data layer.
* Afin de mitiger l'impact du registre distribué sur les agents déployés sur l'infrastructure réseau, les acteurs impliqués dans ce processus peuvent également implémenter dans un contrat intelligent le procédé d'archivage présenté dans le Chapitre 3 de ce document. En effet, les

données de performances stockées sur le registre n'ont pour ces noeuds aucune valeur ajoutée une fois ces dernières traitées et agrégées en KPIs. Ils peuvent alors déléguer leur archivage à des nœuds archivistes de confiance.

Dans ce travail, il a également été proposé d'utiliser un registre distribué à base de DAG. En effet, ce type de registre distribué est connu pour présenter de meilleures performances comparées aux Blockchains [9, 19]. Pour cela, une étude de l'implémentation de la data layer sur le Tangle [11], un registre de ce type, a été effectuée. Cette étude a également permis les simulations présentées dans le Chapitre 3 de ce manuscrit.

Ce travail aura permis de conclure que l'utilisation d'un registre distribué à base de DAG est pertinente dans ce cas-là. Il est à noter que cette approche permet également d'éviter du Sharding avec le déploiement de plusieurs instances de registres distribués, ce qui simplifie le déploiement de l'architecture.

Conclusions et perspectives de la thèse

Contributions de la thèse

Ce travail de thèse a principalement permis la définition d'une architecture dite de "*data layer*" permettant la fiabilisation de rapports de performance produits par une architecture multi-acteur. Il a en effet été identifié lors de ce travail de thèse le besoin de confiance dans les indicateur de performances Cloud/réseau afin de garantir une collaboration fructueuse. La technologie des registres distribués est considérée pour déployer l'architecture proposée. En effet, cette technologie permet la création d'une base de données décentralisée sur un réseau de pair à pair, rendue irréfutable par des mécanismes cryptographiques, ce qui permet l'implémentation de la *data layer* de manière décentralisée, permettant ainsi de renforcer la confiance entre les acteurs et de réduire les coûts.

Cette thèse a également permis d'identifier les registres distribués à base de DAG comme technologie de registre distribué adaptée pour ce cas d'usage. Ce type de technologie est en effet réputé par ses performances offertes, et sa consommation de ressources relativement faible. Des simulations du Tangle, une technologie de registre distribué à base de DAG ont permis de confirmer ce choix.

D'autres contributions ont également pu être proposées autour de cette architecture de *"data layer"*, décrites dans les chapitres 2 et 3. Ces contributions incluent notamment un procédé permettant l'optimisation d'une infrastructure réseau pour optimiser la collaboration, un procédé permettant l'élaboration d'une source de temps décentralisée, ainsi qu'un procédé permettant de réduire l'impact d'un registre distribué sur le stockage de certains nœuds y participant.

Perspectives de recherche

Ce travail de thèse a permis d'établir les bases d'une architecture permettant la sécurisation d'indicateurs de confiance depuis une infrastructure multi-acteurs. Ce travail ouvre plusieurs perspectives de recherche, allant de pair avec l'évolution des réseaux :

* Tout d'abord, de nouvelles architectures de collaboration dans le monde des télécommunications peuvent être définies. Ces évolutions vont de pair non seulement avec l'évolution des besoins de connectivité, mais également avec les évolutions techniques dans les réseaux.
* Ensuite, les indicateurs de performance à partager pour assurer des services réseaux multi-acteurs pourront être mieux définis. Pour cela, il sera néanmoins nécessaire de mieux définir les cas d'usage de partage d'infrastructures, et les contraintes associées. Sur ce sujet, il est également à noter qu'un travail est d'ores et déjà effectué pour standardiser les indicateurs de performance à remonter.
* Aussi, des mécanismes de fiabilisation de données de performance, avant leur stockage sur un registre distribué par exemple, gagneraient à être définis. Ces mécanismes permettraient en effet d'améliorer la confiance dans de telles infrastructures multi-acteurs. Il est cependant à noter qu'il est nécessaire de mieux définir la nature des données de performance à collecter pour pouvoir définir des mécanismes permettant leur fiabilisation.
* Finalement, l'évolution de la technologie des registres distribués est également à suivre, cette technologie étant récente et sujette à de nombreuses innovations. De nouvelles technologies pourraient ainsi devenir plus adaptées pour l'architecture proposée dans ce travail de thèse.

Table of Contents

List of acronyms

ANFR Agence Nationale des FRéquences. 127

API Application Programming Interface. 14, 54, 55, 71, 111, 112

ARCEP Autorité de Régulation des Communications Électroniques, des postes et de la distribution de la Presse. 48

B2B Business to Business. 79

BALAdIN BAndwidth Ledger AccountIng Network. 15, 22, 30, 31, 97, 98, 101–103, 105–113, 115–119, 131, 132, 138, 139

BFT Byzantine Fault Tolerant. 44, 79, 80

BGP Border Gateway Protocol. 35

C-RAN Centralised-RAN. 120, 131, 136, 138

CAPEX CAPital EXpenditure. 37, 51

CM Cognitive Module. 122

COCOS COnnectors COllaborative Synchronisation. 13, 21, 42, 43, 122, 131

CPU Central Processing Unit. 108, 113, 117

CSP Communication Service Provider. 40, 51, 52, 56, 119, 120

CU Central Unit. 120

DAG Directed Acyclic Graph. 10, 14, 15, 29, 31, 59, 75–77, 79–81, 83, 85, 87–89, 96, 119, 128, 129, 131–133, 137–139

DDoS Distributed Denial of Service. 42, 130

DHT Distributed HashTable. 62

DLT Distributed Ledger Technology. 7, 14–16, 22, 27–31, 33–35, 38, 39, 42, 44–57, 59, 61, 63, 65, 67, 69, 71, 73–75, 77, 79, 89–102, 104, 106, 108–110, 112, 114, 116, 118–122, 124–128, 130–133, 135, 137–140, 159

DSO Data Space Optimisation. 28, 40, 41

DU Distributed Unit. 120

E2E End-to-End. 26, 31, 51, 55, 98, 119–123, 125–129, 131, 132, 135

EEA European Economic Area. 35

eMBB enhanced Mobile BroadBand. 127

ERC Ethereum Request for Comments. 71

ETSI European Telecommunications Standards Institute. 47

HNT Helium Network Token. 46, 101

IaaS Infrastructure as a Service. 21, 37

ICO Initial Coin Offering. 47

IDSA Industrial DataSpace Association. 6, 13, 21, 26, 28, 34, 38–42, 44, 45, 50, 51, 55, 98, 99, 136

IoT Internet of Things. 25, 38, 77

IP Internet Protocol. 75

KPI Key Performance Indicator. 9, 10, 26, 31, 98, 118, 119, 121, 123, 125, 127–132, 135, 137, 138, 140

LTE Long Term Evolution. 106

MARL Multi-Agent Reinforcement Learning. 16, 29, 31, 34, 38, 45, 48, 56, 136, 138, 155, 156

MCMC Markov Chain Monte-Carlo. 89

MEC Multi-access Edge Computing. 28, 33, 34, 40, 46, 47, 120, 136

MEF Metro Ethernet Forum. 47

MNO Mobile Network Operator. 16, 34, 35, 45, 47–51, 56, 98, 120, 136, 139, 155, 156

NFT Non Fungible Token. 71

NFV Network Function Virtualisation. 37, 120

NTP Network Time Protocol. 28, 42, 113

OAI Open Air Interface. 26

ONAP Open Network Automation Platform. 5, 26

OS Operating System. 36

P2P Peer-to-Peer. 60, 62, 110

PaaS Platform as a Service. 21, 37

PKI Public Key Infrastructure. 103, 104, 116

PoB Proof of Bandwidth. 15, 22, 30, 72, 99–119, 131, 138, 140

PoET Proof of Elapsed Time. 110, 111, 117

PoS Proof of Stake. 14, 72–74, 76, 100, 136

PoW Proof of Work. 14, 28, 42, 44, 62, 64–66, 68, 70, 72, 73, 76, 78, 80, 130, 136

QoE Quality of Experience. 118

QoS Quality of Service. 6, 25, 26, 40, 41, 51, 53, 55, 98, 118, 120, 127, 131, 135–137

RAM Random Access Memory. 108, 113

RAN Radio Access Network. 15, 35, 47, 48, 119, 120, 139

REST REpresentational State Transfer. 111, 112

RSA Rivest–Shamir–Adleman. 108

RTT Round Trip Time. 40, 43

RU Radio Unit. 120

SaaS Software as a Service. 21, 37

SAM Software Asset Management. 122

SCRATT SeCuRe Archiving of Transactions for distributed ledger Technology. 92, 94, 95, 125, 126, 131

SGX Software Guard eXtensions. 110, 117

SHA Secure Hash Algorithm. 64, 108

SIM Subscriber Identity Module. 52

SLA Service Level Agreement. 26, 52–55, 120–123, 125

SNMP Simple Network Management Protocol. 9

SNRI Signal-to-Noise Ratio Interference. 127

SWIFT Society for Worldwide Interbank Financial Telecommunication. 59, 60

TOR TOR Onion Router. 19, 99–101

TPS Transactions per Second. 72, 80, 91, 109, 110, 115, 127, 129, 131–133

TSA Tip Selection Algorithm. 77, 78, 80, 82–84, 89

UE User Equipment. 111

UN United Nation. 25, 48, 122

uRLLC Ultra-Reliable Low-Latency Communications. 121, 127, 128, 136

UTXO Unspent Transaction Output. 21, 60–62, 90

VM Virtual Machine. 36, 115

VNF Virtualised Network Function. 120, 128, 135

VRF Verifiable Random Function. 73, 74

WDM Wavelength Division Multiplexing. 34, 35, 53, 120, 159

Wi-Fi Wireless Fidelity. 110

ZKP Zero Knowledge Proof. 140

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Context of the thesis

The Internet is now a whole part of our daily lives. The United Nation (UN) has indeed declared access to the Internet as one of the human rights through a non-binding resolution [1]. Moreover, the Covid-19 outbreak and its consequences has further fostered the use of new technologies, thus making telecommunication networks essential as per the generalisation of remote working/studying.

Furthermore, multiple new network-related use-cases are spreading, including but not limited to smart cars, drones, Internet of Things (IoT) devices, etc. This particularly leads to an increase of general data consumption to sustain the fast development of this growing ecosystem, while also requiring resilient network infrastructures to meet the necessary Quality of Service (QoS) (including but not limited to latency, availability, throughput, etc.).

On telco side, new infrastructures will be needed to sustain this development. In the meantime, deploying and maintaining new network infrastructures is a burden for telcos as per the high cost that comes which such operations. Furthermore, the energy consumption and global footprints of digital infrastructures will thus increase with the size of the new networks.

In this context, sharing the infrastructure between multiple providers is relevant as this allows to maximise its usage, thus allowing telcos to not only build more efficient networks, but also lower costs. Indeed, a connectivity provider needing a temporary increase of capacity to

accommodate customer's needs could outsource some resources to another telco under-using his infrastructure [2].

Such exchanges can be made highly dynamic thanks to virtualisation technologies. Virtualisation allows the deployment of virtual, software-based network functions running on commodity hardware. This further allows to fully automate the life-cycle of a virtual network infrastructure, and deploy/withdraw virtual resources on physical infrastructures automatically as no more physical interventions are then needed.

As a result, a global End-to-End (E2E) network infrastructure could then be completely self-managed with software units, with resources automatically ordered, managed and withdrawn when needed. Various initiatives have been led to achieve such automation, such as the Open Network Automation Platform (ONAP) [3], providing a full set of utilities to fully orchestrate and manage the lifecycle of a softwarised network infrastructure, or the Open Air Interface (OAI) project [20] providing various softwares to help creating softwarised mobile network infrastructures.

Such a virtualised network infrastructure could then further be easily shared among multiple *providers* and *consumers* (*prosumers*), the firsts providing physical premises (antennas, optical networks, cloud infrastructures, etc.) to deploy specific network elements, and the seconds building full E2E network infrastructures.

1.2 Problem statement

However, performance requirements must be met for the successful operation of a virtualised network infrastructure to ensure the delivered QoS. Such requirements are of various nature, either on the networking side (maximum latency/jitter, available bandwidth, etc.), or on the computing side (available CPU, RAM, storage, etc.). In a multi-actor context, the performance must then be guaranteed to prove that the infrastructure meets the required QoS, negotiated in Service Level Agreements (SLAs). This condition is necessary to allow deployment of End-to-End (E2E) virtualised network services through the infrastructure of multiple stakeholders.

As a result, some performance indicators need to be collected on the network infrastructure, secured, cleaned and aggregated into trusted, reliable Key Performance Indicators (KPIs) in a trustworthy way. The produced data can then assure that the network infrastructure complies with the negotiated and derived SLAs. The produced indicators can further be used to facilitate billing or for conflict resolution.

The problem of having access to trusted, reliable operational data isn't bound to shared network infrastructures, as most multi-actor collaboration use-cases might need such data, produced by trusted interfaces with the outside world. As an example, the Industrial DataSpace Association (IDSA) architecture allows multiple stakeholders to share data in a secure way

within a federated cloud infrastructure. The whole journey of the data is certified from its production to its usage, thus making it trustworthy in a collaborative environment. It however needs a "Data Clearing House" as a trusted third party to validate and clean the produced data [21]. Similar problems arise with smart contracts [22] deployed on Distributed Ledgers, as they may need in many cases data provided by *oracles* to work properly. Oracles are trusted third parties required for the successful operation of smart-contracts requiring such an interface with the outside world [23]. Any failure of the Oracle can compromise the entire smart-contract. Furthermore, a corrupt smart-contract may lead to huge consequences [24].

To summarise, a trusted interface between a multi-actor network infrastructure and its management entity is needed for a successful operation. Yet, nowadays the question of the implementation of such an interface remains complex, for the telecommunication ecosystem is rapidly mutating and evolving, with novel techniques and use-cases still being studied nowadays. Truthfulness technologies like the DLT are rapidly evolving as well. As a result, this thesis work attempts to answer to the following questions:

* Which work can be achieved on the infrastructure itself to help it support collaboration?
* Which new collaboration scenarii are now possible to further enhance communication networks?
* How can a trusted source of reliable data could be implemented in a decentralised way? What would be its high-level architecture, regardless of the underlying connectivity use-cases and infrastructure? Can the DLT be used to achieve decentralisation in this context?
* Considering the usage of the DLT for producing trusted operational data, which current technology can be the better fit? Can any steps be made to further improve the DLT for this use-case?
* Which mechanisms telcos involved in multi-actor collaboration can implement to secure operational data?

1.3 Manuscript structure & contributions

To address the issues presented above, many contributions have been explored on this thesis work. All of these contributions are detailed below :

* On Chapter 2 collaboration between telcos is explored. This chapter covers more in depth some possible infrastructure sharing scenarii between telcos, as well as the benefits and challenges to solve for such scenarii. For that purpose, at first some state of the art technologies allowing infrastructure sharing are explored. This thesis work has then been the opportunity to improve the infrastructure to sustain collaboration. A method has indeed been proposed during this work, to allow the optimisation of the telecommunication architecture for improved

data exchange, taking the Industrial DataSpace Association (IDSA) architecture as a use-case. Notably, a Data Space Optimisation (DSO) function has been proposed to facilitate communication between actors needing infrastructure optimisations (e.g. IDSA connectors), and the telcos managing the underlying infrastructure. This function then receives optimisation requests from the actors, process them and request to telcos specific infrastructures changes to meet actors' requests. The proposed method does not only allow telcos to improve their infrastructure to meet actors' needs, but also allows the optimal placement of applications to deploy on the infrastructure (e.g. let the possibility to deploy an application close to end user connection points), thanks to Multi-access Edge Computing (MEC) capabilities. Such optimisations can be fully automated thanks to the virtualisation of network infrastructures. This work has led to the following patent:

> B. Radier, G. Fromentoux, A. Braud, and V. Messié, "Procédé de traitement d'un service de transport de données", pat. WO2022034273A1, Feb. 17, 2022. [Online]. Available: `https://patents.google.com/patent/WO2022034273A1/en?inventor=messi%C3%A9&oq=inventor:messi%C3%A9` (visited on 05/17/2022)

Then, another contribution has been made during this work, as to allow multiple partners exchanging data to synchronise their clocks in a decentralised way. The proposed method further aims at allowing partners involved in a collaboration to trust the timestamping of the data they exchange. This contribution accounts for partners having each access to different time sources usually not precisely synchronised between them (such as navigation satellite, Network Time Protocol (NTP) servers, etc.). For that purpose, the method involves a process partners implement to select among them a "synchronisator", whose clock is then used as the reference in the collaboration environment. The process then repeats at a regular interval, to allow the selection of a new synchronisator at each iteration. The proposed process relies on a consensus mechanism like the Proof of Work (PoW) for a decentralised selection of the synchronisator. This contribution has led to the following patent:

> V. Messie, B. Radier, A. Braud, and G. Fromentoux, "Procédé de synchronisation d'une pluralité de serveurs de communications, dispositifs et programmes d'ordinateurs correspondants", French pat. 3114712A1, Apr. 1, 2022. [Online]. Available: `https://patents.google.com/patent/FR3114712A1/fr?inventor=messi%C3%A9&oq=inventor:messi%C3%A9` (visited on 05/17/2022)

Then novel collaboration scenarii have been studied in this thesis work. It is worth noting that new technologies like the Distributed Ledger Technology (DLT) allow multiple novel collaboration scenarii to arise, thanks to the level of trust provided. The technology further allows decentralisation, and minimises the impact of trusted third parties on multi-actor

collaboration scenarii. State-of-the-art novel collaboration scenarii based on the DLT are first presented. Novel collaboration scenarii are also proposed as contributions of this thesis work, taking advantages of novel technologies such as Multi-Agent Reinforcement Learning (MARL) or the DLT to foster collaboration between telcos. Along with game theory, MARL can indeed be used to model interactions between multiple actors acting in their personal interest, as well as testing which mechanisms are efficient to foster collaboration. The DLT can be further used to help implementing the proposed collaborative models in a decentralised way.

The proposed contributions then involve a mobile access network sharing scenario to save energy, and then a federated telecommunication marketplace architecture. In both cases, the goal is to give every single actor incentives to collaborate, thanks to a trusted environment fostering collaboration. These two contributions have led to the following publications:

* X. Marjou, T. Le Gléau, V. Messié, B. Radier, T. Lemlouma, and G. Fromentoux, "Evaluating Inter-Operator Cooperation Scenarios to Save Radio Access Network Energy", in *2022 1st International Conference on 6G Networking (6GNet)*, Jul. 2022, pp. 1–5. DOI: 10.1109/6GNet54646.2022.9830283

* A. Adhiappan, A. Chernetsov, M. Fenomenov, U. Karabudak, A. Korabanova, S. Kislyakov, L. Le Beller, M. Nati, B. Radier, A. Sushkov, A. Ustimenko, A. Vedin, O. Yurlov, T. Ben Meriem, V. Messié, and N. Omnes, "Federated CSPs Marketplace : A DLT-based Data Trust enabling Business Assurance for CSPs Platforms Federation", TM Forum, White Paper 1.0, Nov. 13, 2020. [Online]. Available: `https://www.tmforum.org/vertical-industry-telcos-federated-dlt-based-marketplace/` (visited on 09/20/2022)

* Then in Chapter 3 focus on the DLT is made. This technology allows the creation of a decentralised, trusted database, that has the ability to store certified information. While first devoted to crypto-currencies like Bitcoin, the technology has been further expanded to take into account various applications, thanks to technologies allowing the creation of customised smart-contracts. The technology can be implemented either within a *permissioned* infrastructure (private, with access control) or within a *permissionless* infrastructure (public, without access control). In this chapter, first a literature review is made on current existing technologies, as well as their key characteristics. Then the Tangle technology [11, 25], a DLT relying on a Directed Acyclic Graph (DAG) structure to store transactions, is further explored. This study of the Tangle has led to the creation of a simulator of the Tangle's DAG in this thesis work [1]. This simulator models the Tangle as a stochastic process where time is discretised to samples, considering two previous studies existing in the literature [11, 26].

1. Simulator is in open-source and available at `https://gitlab.com/vmessie/dag-simulator`

The simulations undertaken then showed consistence of the simulations with state-of-the-art results. This contribution then helped to validate this thesis' proposal of a trusted "data layer", presented later in Chapter 4.

Another aspect of the DLT investigated on this thesis work is the question of the storage used by the DLT. Indeed, as by-design a DLT is an "add-only" database where all transactions must be kept for auditing purposes, the required amount of storage to cache the ledger is perpetually increasing. Mechanisms are already put in place to mitigate the impact of this perpetually-growing storage requirement on the DLT nodes with low storage capabilities. However, such mechanisms usually imply nodes to rely on fixed trusted third parties with higher storage capabilities to archive old transactions. To address this drawback, a method allowing DLT nodes to select "archivist" nodes of their choice for trust delegation has been explored in this thesis work. Such a process should then mitigate the trust issue, as the nodes are then able to choose which node to trust, instead of relying on imposed ones to store the ledger content. This method can be implemented as a smart-contract, and thus be overlayed on any DLT, either public or private. This method then led to the following patent:

> V. Messié, B. Radier, G. Fromentoux, and A. Braud, "Procédé de gestion d'un registre local d'un noeud appartenant à un ensemble de noeuds contribuant à un registre distribué", French pat. 2 105 671, Filed, 2022

* Finally in Chapter 4, various DLT-based solutions providing sources of trusted, reliable network operational data are explored. The work on collaborative network scenarii presented in Chapter 2 has indeed shown the need for such a trusted operational data source. The Distributed Ledger Technology (DLT) is further considered to help providing such a trusted data source, in a decentralised way and then foster its adoption by multiple distinct actors. Prior to this thesis work, BAndwidth Ledger AccountIng Network (BALAdIN), a contribution allowing the creation of collaborative networks had been proposed. This contribution aims at giving incentives to a crowd of "local actors" like shops, railway stations, etc. to provide a network coverage with the help of their telcos. The system then allows the creation of network paths between a (travelling) customer, a local actor, the local actor's telco and the customer's telco. For that purpose, a Blockchain-based DLT is used to enhance trust, as well as a "Proof of Bandwidth (PoB)" mechanism. This mechanism allows actors involved in a network path to pro-actively monitor its usage in a decentralised way, thanks to a special frame exchanged at a regular interval, and then stored onto the Blockchain as a proof testifying on the used bandwidth of the path. This initial proposal has been further expanded during this thesis work, with an enhanced study of the initial proposal, as well as simulations of the proposed protocol. In the latter work, a deployment architecture of the DLT processing PoBs has

been proposed and implemented on a virtualised network testbed [2]. It is worth noting that it has been considered in this work to use a Blockchain-based DLT. The work then indicated that such a DLT might not be suitable to support the proposed use-case, for it lacks enough elasticity. The BALAdIN proposal as a whole has then led to the following publications:

* V. Messié, G. Fromentoux, and N. Omnes, "Method for preparing usage data for relays used during a communication between two devices and for searching for the data and associated devices", U.S. Patent 20210092110A1, Mar. 25, 2021. [Online]. Available: `https://patents.google.com/patent/US20210092110A1/en?inventor=messi%C3%A9&oq=inventor:messi%C3%A9` (visited on 05/17/2022)

* V. Messié, G. Fromentoux, X. Marjou, and N. Labidurie, "BALAdIN for blockchain-based 5G networks", in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, IEEE, 2019, pp. 201–205. DOI: `10.1109/ICIN.2019.8685867`

* V. Messié, G. Fromentoux, N. Labidurie, B. Radier, S. Vaton, and I. Amigo, "BALAdIN: truthfulness in collaborative access networks with distributed ledgers", *Annals of Telecommunications*, Jun. 6, 2021, ISSN: 1958-9395. DOI: `10.1007/s12243-021-00855-x`

Then, during this thesis work, a trusted "data layer" architecture has been defined, aiming at helping partners involved in a multi-actor disaggregated E2E connectivity service chain to produce trusted Key Performance Indicators (KPIs) out of shared performance metrics, in a decentralised way. The produced KPIs can then be used on the system managing the collaborative network as trusted sources of information. Example of such collaborative networks capable of implementing the proposed data layer are MARL-based mobile network sharing scenario analysed Subsection 2.4.2, and the federated marketplace proposal proposed Subsection 2.4.3. Based on the previous work on the BALAdIN proposal, DAG-based DLTs are rather considered to implement the proposed data layer. Then simulations of the Iota's Tangle using the simulator presented in the previous chapter allowed to validate this proposition. This work has led to the following publication:

V. Messié, B. Radier, V. K. Quintuna Rodriguez, G. Fromentoux, S. Vaton, and I. Amigo, "A decentralised data layer for collaborative End-to-End service assurance", in *2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, Mar. 2022, pp. 81–85. DOI: `10.1109/ICIN53892.2022.9758094`

2. Code is in open source at `https://gitlab.com/vmessie/baladin-transaction-processor` and `https://gitlab.com/vmessie/baladin-dummy-pob-generator`

Each chapter of this work is then about a specific functional brick allowing collaboration and trust between telcos. Figure 1.1 illustrates the relationships between chapters:

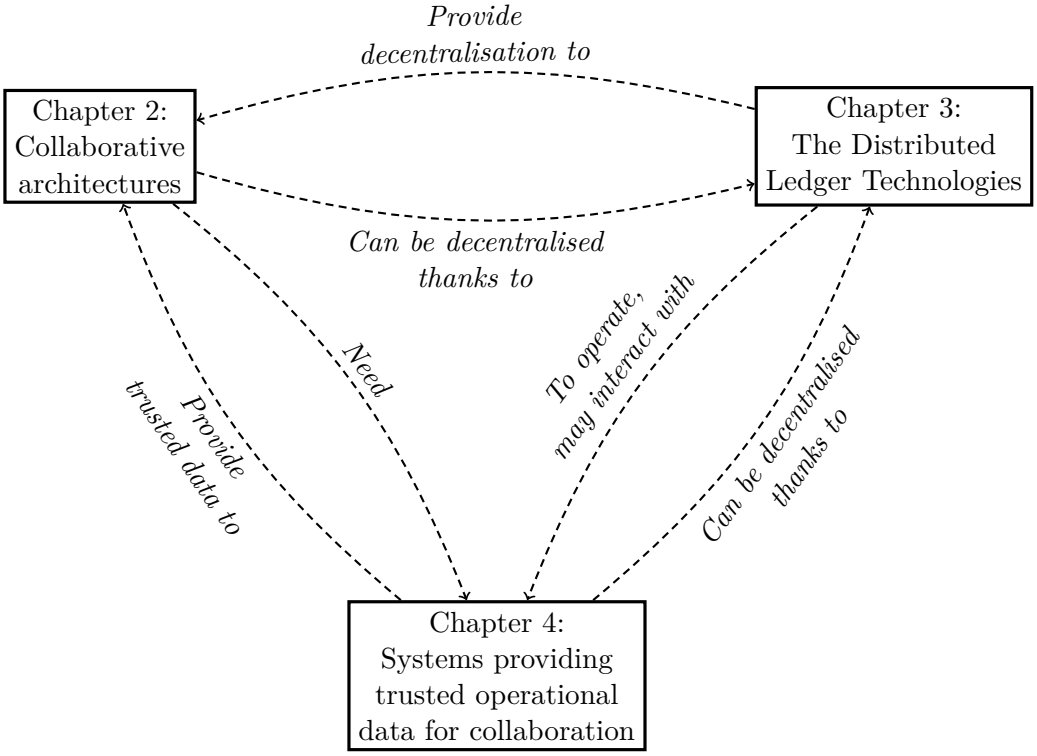


Figure 1.1 – The document structure

Chapter 2

Decentralised systems for collaborative networks

Contents

This chapter explores collaboration in telecommunication related environments. First, architectures fostering trust between multiple distinct actors are explored. Then, collaborative network architectures are presented, allowing and inciting multiple telcos and non telco-partners to share connectivity-related resources thanks to truthfulness technologies like the Distributed Ledger Technology (DLT).

2.1 Introduction

The evolution of telecommunication networks towards cloud-native environments in addition to the diversification of customer needs has given rise to a deep transformation in network ecosystems. Connectivity services are no more provided by a single network operator but may involve various telcos and other players, each providing specific network domains and sub-services deployed within virtual and physical infrastructures, cloud environments, edge sites, transport networks, etc. From a business perspective, network evolution has enabled the emergence of new markets involving various telecom assets providers, and a crowd of "consumers" having specific connectivity needs. These providers include towercos (the owners of towers), Multi-

access Edge Computing (MEC) providers offering a better quality of service by placing their cloud infrastructures as close as possible to end-users, cloud providers managing distributed computing and storage capacities, and connectivity providers taking advantage of technologies such as Wavelength Division Multiplexing (WDM) to share fiber links between multiple services. As a result, many architectures implying infrastructure sharing are imagined. As an example, Section 2.2 details nowadays's cloud sharing use-cases.

In this chapter, optimisations of the infrastructure to sustain collaboration are first explored. In Subsection 2.3.1, the Industrial DataSpace Association (IDSA) architecture is presented as a use-case aiming at facilitate secure data exchanges between multiple distinct partners. The infrastructure further allows data owners to keep control on which usage of their data is made [21].

Optimisations of the infrastructure sustaining such an architecture have then been proposed in this thesis work. First, a method aiming at optimising the network infrastructure supporting an IDSA "dataspace" has been explored in this thesis. This method, presented in Subsection 2.3.2, has led to a patent [4]. Then a method providing a decentralised clock synchronisation protocol has been proposed. This method accounts for the need to accurately timestamp data produced in an IDSA environment, and allows partners to solve this problem without having to rely on a single trusted time source, thus a single point of failure. This contribution, presented in Subsection 2.3.3, has also led to a patent [6]. While the above methods have been designed for the IDSA environment, they can be further extended to any similar collaborative infrastructure.

Then the next section deals with collaboration between telcos themselves. Further scenarii are evoked where multiple telcos not trusting each other share their infrastructure in order to provide enhanced connectivity services. For that purpose, the use of novel technologies fostering collaboration and truthfulness, such as Multi-Agent Reinforcement Learning (MARL) or the Distributed Ledger Technology (DLT) are considered. MARL algorithms can indeed help multiple actors to learn which rules and policies to implement in order to cooperate in the most efficient way, while the DLTs can help foster trust between actors in a decentralised way. A Distributed Ledger is a shared database, made immutable and non-repudiable thanks to cryptographic mechanisms. The created database is add-only, which means stored data can't be deleted. Data stored onto a Distributed Ledger is then foolproof and easily auditable, which makes the technology well-suited for storing contractual agreements. At first, Subsection 2.4.1 presents multiple DLT-based collaborative scenarii use-cases, aiming at providing better services for customers. Then Subsection 2.4.2 introduces a contribution of this thesis work on a solution allowing multiple Mobile Network Operators (MNOs) to share their infrastructure to save energy. For that purpose, they use a MARL framework to help them evaluating the optimal collaboration policy, and the usage of the DLT is considered to provide reliable energy consumption reports from the infrastructure. This contribution has led to a publication [2]. Finally, a "federated connectivity" marketplace has been introduced and explored during this thesis work. The

presented architecture aims at connecting multiple "asset providers" providing various network infrastructures such as tower infrastructures (mobile antennas), Cloud, or optical infrastructures and multiple "service providers", using multiple asset providers' infrastructure in order to deploy virtualised network functions to provide to their customers enhanced, customised connectivity services. The proposed architecture aims at managing every aspect of such a marketplace, and goes from providing a search engine to allow service providers to search specific assets, to automated asset onboarding (deployment), orchestration as well as billing and accounting. The DLT is considered to power this marketplace in a decentralised way, as the technology can be used to store any marketplace events in a trusted, easily auditable database. This research has been performed in collaboration with multiple external, both non-telco and telco partners in a TMForum Catalyst. It has led to a white paper [7]. this contribution is further extended in Subsection 2.4.3.

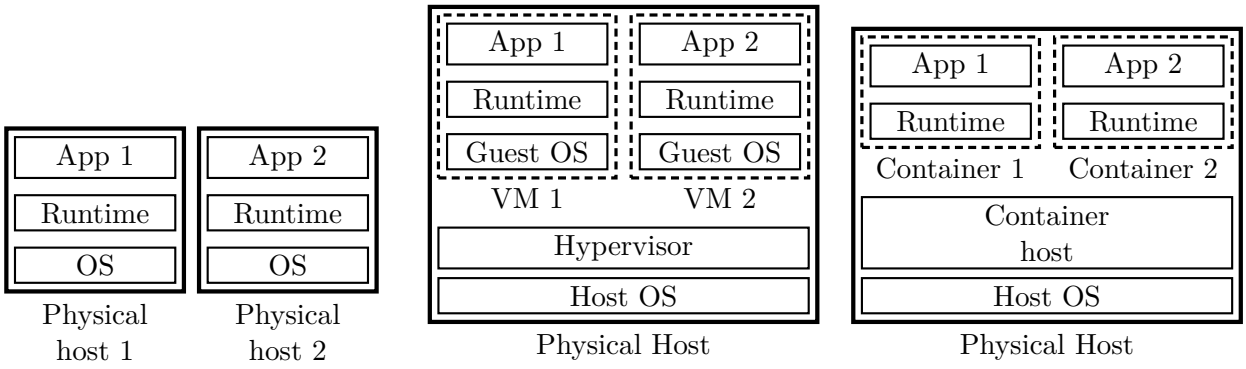
2.2 On digital resource sharing

Network infrastructure sharing isn't something new, for there already are many situations requiring the cooperation of multiple telcos. The prime example of infrastructure sharing is the Internet network, composed of multiple distinct "Autonomous Systems" managed by different actors. They are then interconnected using the Border Gateway Protocol (BGP), a protocol used to facilitate such multi-actor interconnections [27]. Another popular infrastructure sharing use-case is **international roaming**. Roaming allows a customer of a given telco (a *home operator*) to get a network coverage while travelling abroad, provided by a local telco (a *visited operator*), that has an agreement with the home telco. However, such agreements can be hard to meet, and are often costly for the final user due to the complexity for telcos to reach a consensus. In the European Economic Area (EEA), a "Roam Like At Home" policy has been established, in order to abolish roaming surcharges for every European Mobile Network Operator (MNO). Although work has shown the benefits of this agreement considering value creation, many policies and regulations had to be set up to prevent abusive usages of roaming, both from users and MNOs [28]. At nation scale, in both fixed and mobile networks, the physical infrastructure can often be shared between multiple operators to lower costs. An optical network may be shared between multiple operators thanks to technologies like Wavelength Division Multiplexing (WDM) [29]. The access network, either fixed or mobile, can also be shared between multiple operators, or be managed by a completely different stakeholder. As an example, the mobile Radio Access Network (RAN) can also be shared between multiple telcos, to further enhance connectivity. [30] lists multiple scenarii of RAN-sharing available for telcos. In all of these sharing scenarii, telcos share their infrastructure to provide a better service to their customer (e.g international network coverage, improved fixed network coverage by lending optical networks, etc.), while

avoiding to deploy costly physical infrastructure. Yet agreements must be established between implied stakeholders to successfully provide the connectivity services.

2.2.1 Cloud Sharing

Connectivity resource sharing can be further enhanced thanks to cloud capabilities. Cloud services are mainly designed for enabling the creation of dedicated IT services on agnostic, mutualised hardware, with the help of virtualisation technologies. Like illustrated on Figure 2.1, virtualised resources can be "*virtual machines*" mimicking the behaviour of a real computer, or "*containers*". As IT services now become softwarised, sharing principles can then further apply on such infrastructures. At first, the physical infrastructure can become now mutualised, as a single physical equipment may give host to multiple virtualised services. This then allows a more efficient use of physical equipments, thus lowering costs. Also, virtual services can be replicated on multiple infrastructures to make them resilient and avoid faults with redundancy, or for "load balancing" purposes i.e. distribute the load of a given application onto multiple equipments [31].



(a) Physical environment (b) Virtual machines (c) Containers

Figure 2.1 – Physical environment vs. Virtualised machines vs Containers

All these innovations have led to the emergence of "as a service" offers, allowing services providers to outsource resources to third-party infrastructure providers. Indeed, a commodity hardware supporting virtualised infrastructures can be then easily mutualised and shared. Furthermore, from customer side, "as a Service" offers allow customers to easily order, deploy and use virtualised infrastructures on physical premises they don't own, thus preventing them to make huge investments in physical infrastructures. Infrastructure providers may offer various levels of service depending on the customer's needs and constraints [32]. Figure 2.2 illustrates the difference between those models, while Table 2.1 lists some cloud infrastructure providers existing on the market. At first, it is possible for infrastructure providers to host and fully

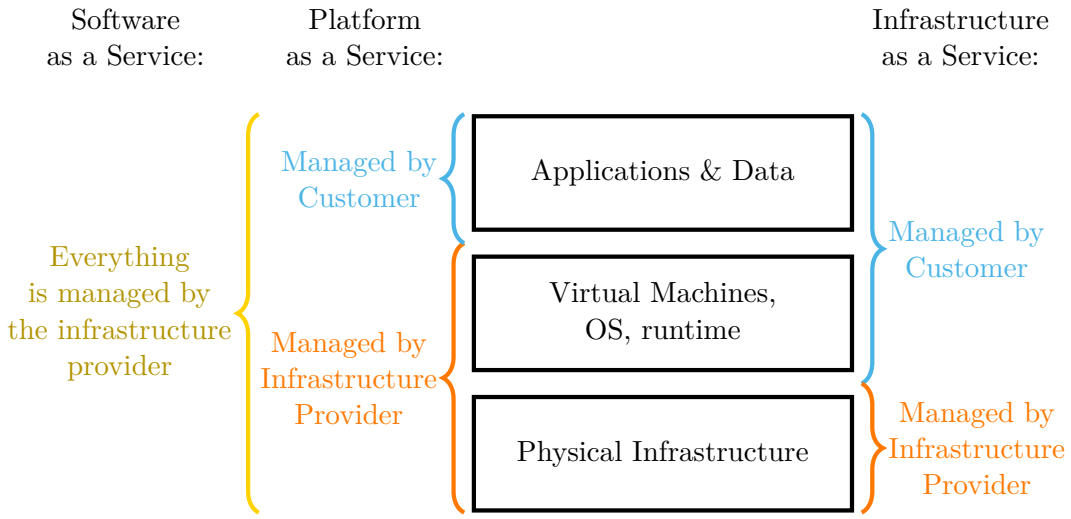


Figure 2.2 – IaaS vs PaaS vs SaaS

Name	SaaS offer	PaaS offer	IaaS offer
Orange Flexible Engine		X	X
Amazon Web Services	X	X	X
Microsoft Azure	X	X	X

Table 2.1 – Some types of Cloud offers

manage specific applications on their infrastructure. This type of service is named Software as a Service (SaaS). Another type of offer is Platform as a Service (PaaS): customers deploy their own applications within the infrastructure, while every other aspect (operating system, runtime, etc.) is managed by the provider. Then Infrastructure as a Service (IaaS) offers allow customers to fully manage their virtual appliances (applications, virtual machine, runtimes, etc.) while only the physical infrastructure is managed by the provider.

Thanks to Network Function Virtualisation (NFV), network functions can also be virtualised, to be deployed on commodity hardware on so-called "*Telco-Clouds*". As a result, cloud technologies enable networks to become more elastic and flexible. It further allows the then-banalised infrastructure to be shared.

Virtualisation technologies thus allow many more infrastructure scenarii to emerge, for the infrastructure itself is made more "elastic". Indeed, as the hardware now becomes banalised and designed to support various needs, a virtualised infrastructure can be instantiated, managed and decommissioned on-the-fly, as this only implies the management of pieces of softwares.

The new "as a Service" business opportunities can further be applied for telecommunication environments, thus allowing telcos to deploy a service on an infrastructure they don't necessarily own, preventing them to use their CAPital EXpenditure (CAPEX) on new infrastructures. Many

collaborative network architectures involving multiple telcos sharing their infrastructure can then be imagined, as to allow telcos to build enhanced connectivity services at lower costs. This market is now further open to non-telco actors [33]. In the following sections, novel collaborative architectures are explored, and telecom-related collaboration is emphasised. First, a framework to support global data exchange between multiple partners is considered, as a framework of trust is mandatory for a successful collaboration between distinct actors. Then thesis contributions aiming at enhancing the infrastructure's capability to support collaboration are explored [4, 6].

After, connectivity-related collaborative architectures are explored, with first a review of existing or past DLT-based infrastructure sharing use-cases. Then novel collaboration scenarii are presented as thesis contributions, taking advantages of technologies such as MARL and the DLT to achieve truthfulness and strengthen bonds between actors [2, 7].

2.3 Improving the infrastructure to sustain collaboration

2.3.1 The Industrial DataSpace Association architecture, a framework for data sharing

As the Internet of Things (IoT) is rapidly expanding, new solutions are emerging for efficiently handle data produced by various services accross the internet. All of these data may however need extra securing, to ensure the integrity of the data itself, its sources, and the applications processing it.

The Industrial DataSpace Association (IDSA) architecture provides a framework allowing multiple actors to safely exchange data between certified *"connectors"* [21]. The goal of this architecture is to properly secure the "*journey of the data*", from its production to its usage. For that purpose, different roles, presented on Figure 2.3, are discussed:

* *data providers* are actors capable of sharing data that they may not own with other partners (*data owner* is a separate role).
* *data consumers* are actors demanding data from providers. They are able to parse the data with the help of a *vocabulary provider*.
* The data is shared via a *service provider*. In a similar way, data consumers and providers can be put in relationship with a *broker service provider*, keeping track of data sources within the dataspace, and providing a search engine for data consumers
* Exchange of data can be automated by deploying applications onto the dataspace, with the help of *app providers* and *app store providers* that consumers and providers may use to browse for applications to use in a data processing workflow.
* Finally, a *data clearing house* is used to keep track on every data usage event happening into the dataspace, in the format of *transactions* stamped, proofed, secured and archived into

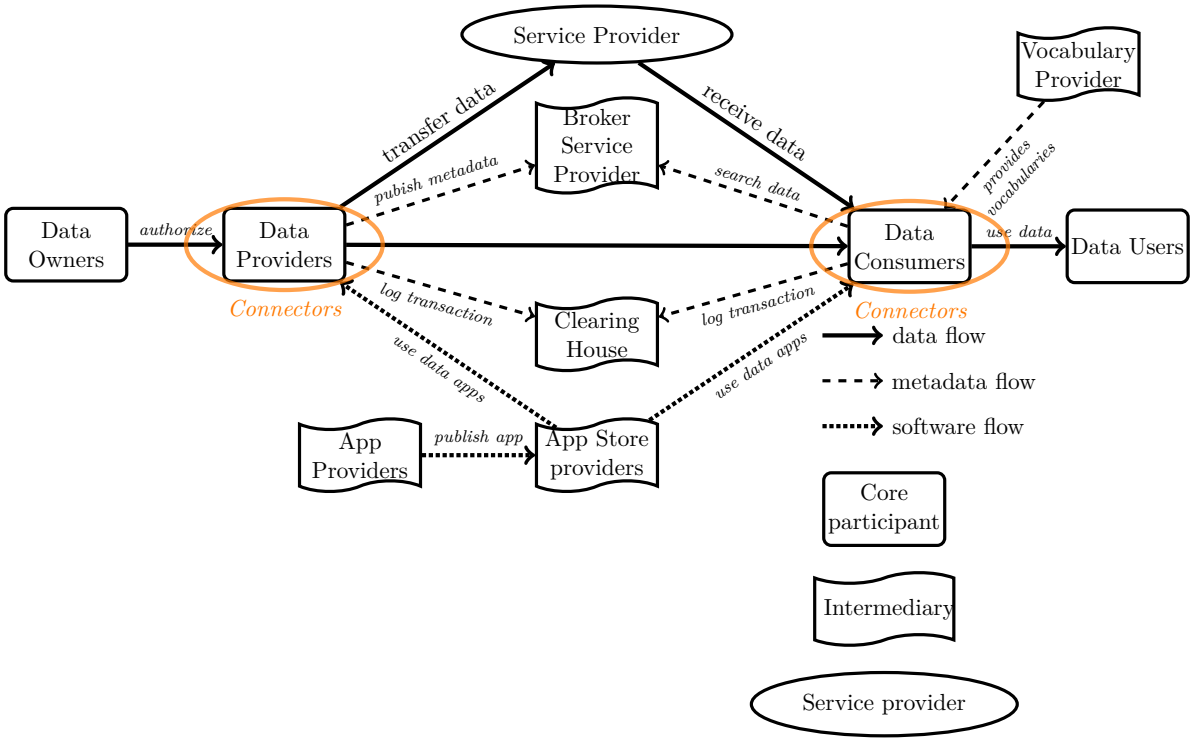


Figure 2.3 – The interactions between the different IDSA roles and components

an easily auditable database. The record of events can then be used afterwards for legal or reconciliation purposes. If any actor of the dataspace fails to meet his obligations (if a data provider fails to answer a request in time for example), the event will be logged as an evidence of this particular actor's failure.

Data consumers and providers are further implemented within *connectors*, trusted environments especially designed for operating IDSA-related functions. Although the IDSA remains agnostic about the data being exchanged, it can be used for the creation of enhanced network connectivity services implying multiple telcos not trusting each other, as the data workflow is then secured. Such use-cases are presented later on this chapter. It is also worth noting that intermediary roles can be implemented in a decentralised way to avoid single points of failure. For example, the DLT can support the data clearing house, thanks to its ability to validate the integrity of any transaction, and store them in an easily auditable database.

Next subsections then present contributions of this thesis aiming at improving the infrastructure to better operate an IDSA or IDSA-like architecture.

2.3.2 Optimising the network infrastructure to optimise data exchanges

In this section a contribution of this thesis work is presented, aiming at providing and guaranteeing a network infrastructure connecting multiple sites exchanging data. This contribution has led to a patent [4]. The architecture considered is the IDSA one, presented Subsection 2.3.1, and aims at allowing the actors involved in the framework to optimise the infrastructure used to support the dataspace. Indeed, the current architecture does not take into account the network infrastructure used to support data exchange. Yet work can be done on the infrastructure itself to make it handle efficiently the journey of data, from source to usage/storage. Indeed, depending of their nature, special needs can be associated with the data handled by the IDSA. As an example, applications might need to collect a huge volume of data at a high frequency, and/or might also require a low latency between data sources and data users/associated applications. These needs can be met by either optimising the network links between the servers ("*connectors*") implied in this specific data exchange, or by deploying the applications close to the data sources in Multi-access Edge Computing (MEC) infrastructures. Yet telcos need to be involved to fulfill such needs as they hold the network infrastructure.

The proposed architecture thus aims at providing a link between the dataspace and the underlying network. It is worth noting that the network infrastructure may be shared between multiple Communication Service Providers (CSPs), and implemented like presented in Subsection 2.4.3. To provide this link, a function named "Data Space Optimisation (DSO)" is deployed. The optimisation process, illustrated Figure 2.4 works as follows:

* At first (step **0**), the DSO function receives a request of optimisation from the dataspace users, containing identifiers of data providers and consumers, as well as application identifiers, all needing to be optimised according to their needs. The request also includes identifiers of the data themselves that are needed to be processed in a secured and trusted environment (step **1**).
* The DSO then sends a request to telcos, in order to get operational informations about the infrastructure. These pieces of information include Round Trip Time (RTT) between connectors, available/used bandwidth. In a similar way, the DSO retrieves informations about cloud/MEC capabilities to run applications (step **2**). In a collaborative network infrastructure like presented in Subsection 2.4.3, the two brokers may be implemented either at the *service providers*, or even directly as the *asset providers* if the DSO is then able to query the marketplace search engine.
* The DSO then selects infrastructures offering the best QoS, fitting the needs of the initial request of optimisation. The DSO may also choose MEC infrastructures as close to the data sources as possible for improved performances and reduced load on the network itself (step **3**).
* Then the DSO calculates the QoS for each possible infrastructure usage, as well as the deploy-

ment costs. At this step, the DSO also queries the broker to seek applications already deployed to avoid their duplication, and about applications needing to be deployed with special QoS constraints (step **4**).

* Finally, the DSO deploys the connectivity services on the selected infrastructures, then deploys the components necessary for supporting the dataspace (step **5**).

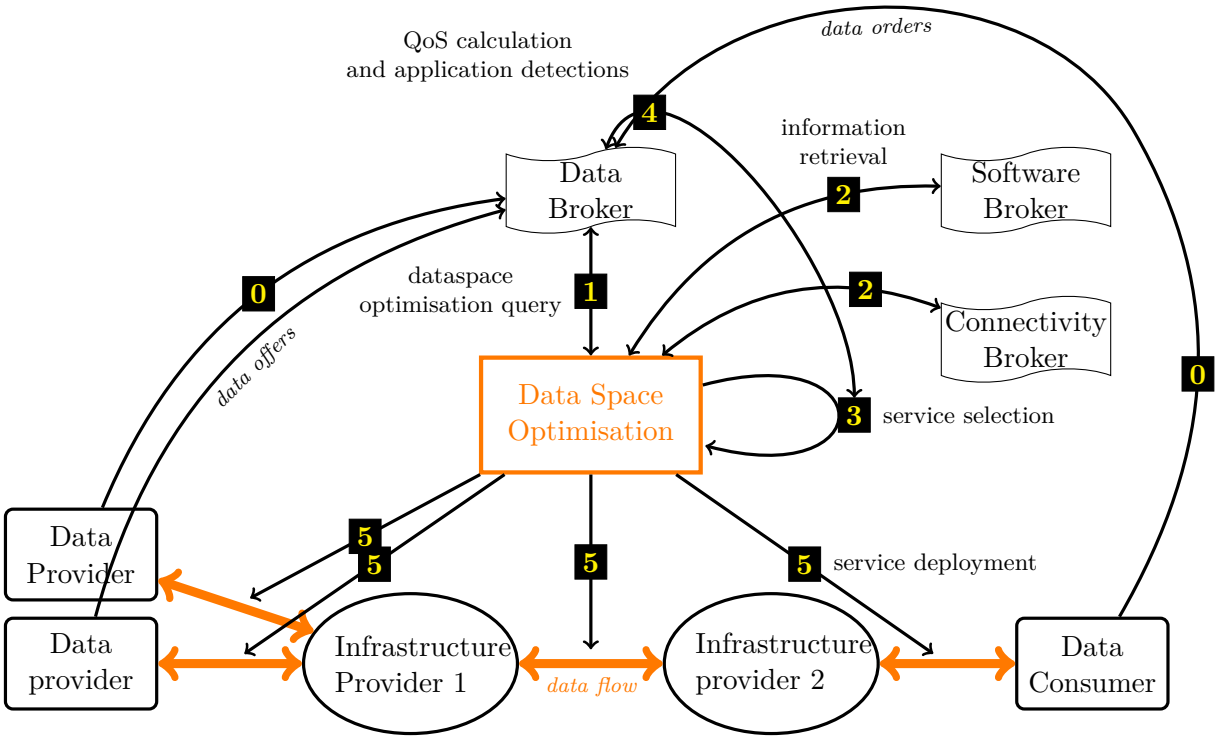


Figure 2.4 – The dataspace optimisation process

This optimisation process is then a way to guarantee that multiple partners can successfully and safely exchange the data they need. Such a function can then help partners like telcos to collaborate, by providing a safe and optimised infrastructure for data exchange.

Multiple other steps could then be performed to further improve the infrastructure for collaborative systems. In particular, an accurate time synchronisation of nodes involved in a collaborative environment, like IDSA connectors, can be the assurance of successful collaboration. Indeed, data exchanged for collaboration might need to be precisely timestamped as per the requirements of underlying use-cases. Events happening on a collaborative system may also need accurate timestamping for archiving, so that the log of events may be used for billing or legal purposes. On this thesis work, a decentralised time synchronisation protocol has also been explored. This protocol, presented in the next section, aims at providing a secure, trust-

worthy mechanism for nodes involved in a collaborative environment (like IDSA connectors) to synchronise their clocks on a common time source).

2.3.3 Time synchronisation in a decentralised way

On this thesis work, "*COnnectors COllaborative Synchronisation (COCOS)*", a proposal of a decentralised time synchronisation protocol has also been made, and has led to a patent [6].

This method solves the need for data exchanged on a collaborative infrastructure, between actors not trusting each other, to be accurately timestamped as it might be required by the collaboration use-case. The proposed method considers an environment like defined by the IDSA with every peer identified by a regulator. Yet the process can be extended to other contexts, like a set of nodes interacting with a DLT.

Usually, while exchanging data in a collaborative environment, peers (IDSA connectors, DLT nodes, etc.) use their local clock to timestamp the data they emit. Peers further use an external time source to synchronise their local clock. External time sources include but are not limited to satellite navigation systems (GPS, Glonass, Galileo, ...), or time synchronisation protocols like Network Time Protocol (NTP). These different time sources may however diverge, become compromised etc., which may lead to synchronisation issues between peers, and badly timestamped data. As a consequence, peers involved in data exchange may need to synchronise their local clock using a common time source to avoid time-related issues. This may lead to the addition of a single point of failure within the system, as the common time source can then become compromised. Multiple truthfulness solutions are then being imagined to provide secure time informations [34, 35]. However, these solution are operated on an environment with a limited number of participants, and with already existing trust among them.

The COCOS method, step by step

The proposed method then describes a protocol allowing a decentralised time synchronisation on a collaborative environment, on which peers involved in collaboration may become turn-by-turn the reference time source.

Figure 2.5 gives an overview of the COCOS method.

On step **1**, each connector implied on a COCOS protocol runs a **Consensus loop**. The purpose of the consensus loop is to select a peer that will then act as a *synchronisator*. This mechanism consists in a challenge to solve, and implements cryptographic mechanisms to avoid attacks like Distributed Denial of Service (DDoS), and as an attempt to give to each peer a chance of being selected as a *synchronisator*. This consensus loop can be implemented using a Proof of Work (PoW) like implemented in the Bitcoin cryptocurrency [8], where the challenge is about finding hash collisions.

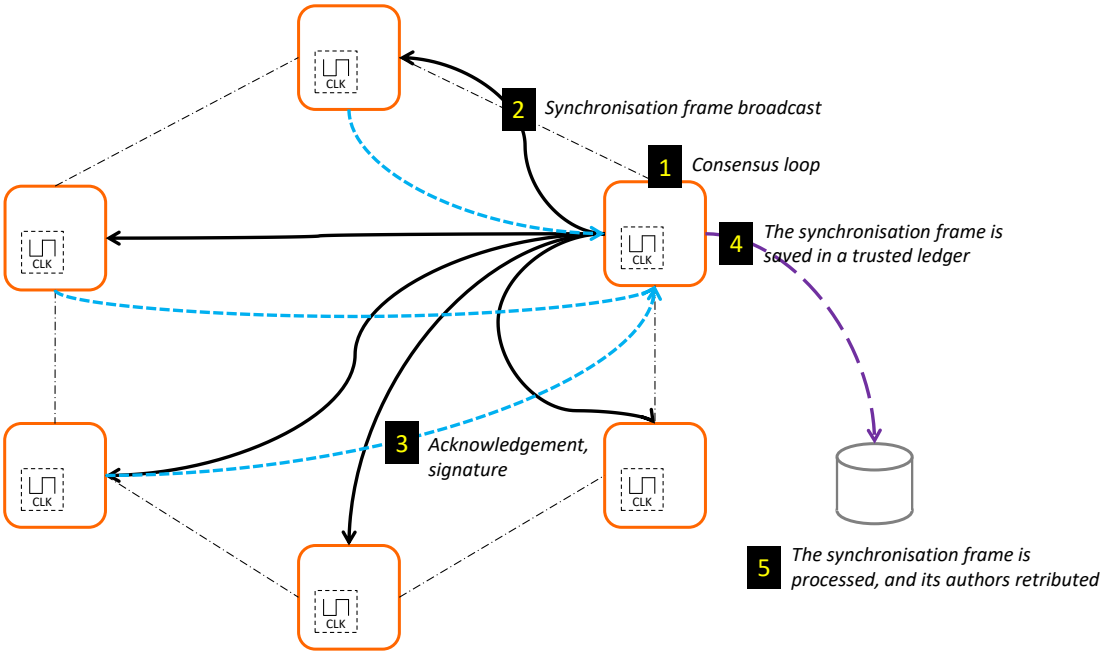


Figure 2.5 – An Overview of the COCOS method

On step **2**, a **synchronisation frame** is sent by a peer that has solved the consensus loop. The frame contains the time of emission of the frame, as well as a value testifying on the resolution of the consensus loop, and the digital signature of the synchronisator peer.

On step **3**, each peer receives the synchronisation frame. First, the digital signature of the frame is checked, as well as the consensus loop result. Upon their validity, the peer then checks the time of emission of the frame. Upon the difference between the time of emission contained in the frame and the time of reception of the frame, calculated with the peer's local clock is within a defined threshold, the peer then updates its local clock, choosing as a reference the time put into the synchronisation frame. The peer may also take into account the propagation delay of the frame while adjusting its local clock. If the whole process is successful, the peer then sends back an acknowledgement frame with its signature.

Nonetheless propagation delays must be known beforehand to be taken into account. It is further worth noting that propagation delays might vary a lot over time, depending on the nature of the underlying network infrastructure (fixed, mobile, etc.). Yet in this work it has been assumed that the proposed method is deployed on an environment without significant delay variations. Round Trip Time (RTT) measurements between peers might then give a simple estimation of the propagation delays.

On step **4**, the synchronisator then collects the acknowledgements of the other peers. After having received a sufficient number of signed acknowledgements, the peer then sends the

synchronisation frames with signed acknowledgements to a trusted ledger.

Then finally, on step **5**, conformity of the complete frame is checked by a set of predefined policies on the ledger. More particularly, the synchronisation frame itself is checked. The validity of all acknowledgement signatures is also checked. Also, it is at this step that the necessary quorum of acknowledgements is enforced. If the frame is valid, it is then saved onto the trusted ledger, for archiving and auditing purposes. Auditing can be proven useful to detect any flaws in a peer sending badly timestamped data, as a trusted trace of synchronisation and subsequent approvals from other peer would then testify on the peer malicious intent. A reward might also be payed to the synchronisator peer, as an incentive for peers to keep the method going, as a PoW-like consensus loop might be expensive to operate. This reward can be implemented with a system of "tokens" that peers may exchange for value afterwards, like in a crypto-currency system [8].

Then the process repeats at a regular time interval established beforehand, so that peers take turns at becoming a synchronisator for each synchronisation frame.

On implementation choices

While the proposed method is overlayed over a system of peers exchanging data, and is thus agnostic to the underlying architectures, the nature of the latter might have an impact on the validation policies chosen by the peers.

At first, the design of the consensus loop may be related to the environment the method is deployed on. As an example, on an IDSA dataspace, the mechanism can be regulated by a central regulator designating a synchronisator at a regular time interval. The mechanism can be further implemented as a Byzantine Fault Tolerant (BFT) voting mechanism as all identities are certified by a central authority. On the contrary, on a permissionless environment like a DLT with peers whose identity isn't checked, a more secure consensus loop like a PoW is necessary to avoid Sybil attacks. In that case, the difficulty of the challenge would be pondered so that a synchronisator would be selected at a regular time interval, in a similar way to Bitcoin [8]. Furthermore, on such a permissionless environment, further Sybil mechanisms would need to be set up for authenticating the acknowledgements, like the "Mana" system implemented in IoTa 2.0 [25]. Such mechanisms, primarily designed for DLTs, are discussed later on this thesis, in Chapter 3.

Also, the threshold chosen for time validation, as well as the propagation delay corrections between peers can be either set up by the peer local policy, or be globally enforced by a common policy shared by every peer in an environment like an IDSA dataspace. Furthermore, the delay corrections can also be imposed by a central authority, and then enforced by the peers receiving synchronisation frames/the trusted ledger. On a permissionless environment, delay corrections could be determined at service establishment, and eventually recalculated later on, or they could

be let up to the nodes receiving synchronisation frames. Yet on such an environment, knowledge of the network infrastructure would be necessary to account for delay variations.

Finally, the trusted ledger can be implemented as a data clearing house on a IDSA dataspace, with a set of policies pre-established regarding full synchronisation frame validation. Another approach to avoid centralisation would be to use a DLT with a customised smart-contract [22] implementing the validation policies. Regarding validation policies, an environment with a limited number of peers could require 100% of the peers to send an acknowledgement frame to consider a synchronisation valid. On the other hand, on an environment with a big and fluctuating number of peers, like a permissionless environment or a big IDSA dataspace, a lower quorum rule could be established, with for example only n acknowledgements with the n to be determinated, or 75% worth of the total number of peers required to validate a synchronisation frame.

2.4 Collaborative systems for enhanced connectivity services

Whereas the previous section described some solutions allowing a network infrastructure to efficiently support collaboration, this section presents multiple scenarii of collaboration where distinct actors share their resources in order to build better services. In this section, usages of the Distributed Ledger Technology (DLT) to achieve such goals are further considered. First, existing DLT-based collaboration use-cases are explored. Then, a first contribution of this thesis is presented, allowing multiple MNOs to share their infrastructure in order to save energy. MNOs take both advantages of MARL algorithms to learn how to trade their resources in a fair way, and usage of the DLT to provide a trusted source of information about the network's energy consumption. Finally, another thesis contribution is presented, introducing a "federated connectivity marketplace". The proposed architecture enables the creation of decentralised network service chains built by multiple "asset providers" that do not trust each other, and used by "service providers" that do not own the infrastructure. The proposed solution takes advantage of the DLT to achieve truthfulness among partners involved in the service chain.

2.4.1 On DLT-based resource sharing

As discussed in Section 2.2, the change of paradigm induced by virtualisation has enabled multiple sharing scenarii to exist. Furthermore, new technologies such as the DLT allow the creation of a trusted environment between distinct stakeholders, allowing them to securely cooperate in a decentralised way by exchanging trusted data. First democratised with the cryptocurrencies like Bitcoin [8], the DLT allows the creation of a trusted, distributed database on a peer-to-peer network, working on a add-only basis. Participants are able to add new data in the format of *transactions*, that are then validated by other participants thanks to cryptographic

	Helium [36]	Ammbr [37]	Bubbletone [38]	Edgechain [39]
Purpose	Decentralised wireless network	Decentralised wireless network	International roaming marketplace	MEC marketpace
Type	Industrial project (open-sourced)	Industrial project (closed-sourced)	Industrial Project (closed-source)	Research project
Status	Online	Unknown	Stalled	N/A

Table 2.2 – DLT-based solution compared

mechanisms. Such data can then be for example negotiation messages (offers, request), service contract establishment/modification/termination that can be intercepted to automatically deploy/modify/withdraw virtualised network resources, etc. Up to now, it is then impossible to alter or delete validated transactions, thus making the content of the Distributed Ledger secure, trustworthy and easily auditable by every actor.

Table 2.2 lists some network or cloud-related technical solutions taking advantage of the DLT. Helium [36] aims at creating the "world first decentralised mobile network". The Helium network is particularly composed of interconnected "Miner" nodes providing a coverage to the users. Communication between the devices using the network and Miners is performed thanks to a customised wireless protocol. Miners are registered onto the system by providing their location, as well as their coverage capabilities and the price of their service. Users of the network are then able to select through the system the resources they need.

The system is made decentralised thanks to a Distributed Ledger used to track all of these events, and a dedicated currency called Helium Network Token (HNT) that users spend to retribute the Miners. Moreover, the Distributed Ledger is managed by the Miners, thanks to a novel protocol called the "Proof of Coverage". This protocol aims at retributing the Miners for the coverage they provide, in order to incite them to participate in the network. The protocol consists in the exchange of multiple frames with neighbouring Miners as a way to assess the provided coverage. Currently the network is live, with the HNT exchanged on specialised markets [1], and with network coverage already offered [2]. The Helium network is rapidly expanding, with that tens of millions of Proofs of Coverage produced since the beginning [40]. The Helium network now also provides 5G mobile coverage in the US, with 5G routers getting rewards for the coverage they provide [41].

Ammbr [37] is a similar project aiming at providing a decentralised wireless network. For that purpose, routers with customised hardware are provided, capable of quickly performing a "Proof of Velocity" algorithm within a limited timeframe while the algorithm would take way more time to be performed without Ammbr's customised hardware. According to the project's

1. `https://coinmarketcap.com/currencies/helium/`
2. Coverage may be checked live at `https://explorer.helium.com/`

claims, this feature would secure the underlying Distributed Ledger, hence allowing the creation of a secure, decentralised mesh network. However, it seems that contrary to Helium, the project hasn't overcome the stage of design. It is not yet known why Ammbr doesn't have the same fate as Helium, as the two project are similar in many way. One explanation could be that the Ammbr project remained closed, whereas Helium is open-sourced, hence fostering the adoption of the latter. Another explanation would be that the Ammbr project required higher investments to develop the routers' customised hardware, whereas such investments weren't necessary for Helium. Yet although the Ammbr project has stalled, it has shown the growing interest on decentralised, DLT-powered network infrastructures.

Bubbletone [38] is an initiative, designed to facilitate international roaming. For that purpose, the project provides a DLT-based marketplace where telcos may publish "offers" of connectivity services. As a result, any *home operator* can select an offer of a then-*visited operator*, to provide connectivity services to a customer abroad. The whole roaming service lifecycle is automated and proofed thanks to the DLT, hence considerably reducing reconciliation time, and reducing overall cost. This project has yet stalled due to an unsuccessful Initial Coin Offering (ICO), whereas it showed the growing interest for telcos to enable better cooperation opportunities and gaining trust.

Another initiative worth citing is EdgeChain [39]. This project aims at providing a marketplace for edge application placement onto Multi-access Edge Computing (MEC) infrastructures. The application is also made fully decentralised thanks to the DLT. The system is designed so that the deployment of any edge application requested by a user is fully automated. For that purpose, EdgeChain also provides an algorithm to automate the placement of edge applications on the optimal infrastructures.

Various other initiatives are led on standardisation side. As an example the Metro Ethernet Forum (MEF) defines specifications and requirements for using a DLT-based framework, in billing and settlement of connectivity resources [42]. The European Telecommunications Standards Institute (ETSI) also studies the DLT and its opportunities for telecommunication-related use-cases [43].

All the cited contributions show the growing interest on DLT-based systems thanks to the DLT ability to build trust at low costs. The next sections will then present contributions of this thesis powered with the DLT and its ability to build trust.

2.4.2 Using reinforcement learning to help distinct telcos share resources to save energy

During this thesis work, collaborative mobile network sharing use-cases were considered. It is indeed technically possible for Mobile Network Operators (MNOs) to redirect their subscribers to partner MNOs, using techniques such as national roaming or RAN-Sharing. This work

considers a cooperation scenario where MNOs use a collaborative framework based on Multi-Agent Reinforcement Learning (MARL) to cooperate in order to temporarily support their subscribers, in order to save energy. The usage of the DLT is further considered to implement the proposed solution in a decentralised way and avoid the use of trusted third parties.

Motivations

Reducing network energy consumption has now become crucial, not only to reduce operational costs, but also to build more sustainable networks, as per UN's goal of "Building resilient infrastructure, promoting sustainable industrialisation and fostering innovation" [44]. This statement is specially true for mobile networks, as on a typical mobile network about 73% of the total energy is consumed by the RAN [45]. As a result, MNOs have been striving to reduce the energy consumption of mobile networks. For that purpose, multiple approaches are possible, like shutting down elements of the RAN when unused [46], or putting some RAN functions higher in the network, in mutualised Cloud infrastructures [47]. Nonetheless significant part of the access infrastructure remains online during low activity periods (per example during the night). During such periods, a single MNO could let its access infrastructure *on-guard* to serve users of other partner MNOs, allowing them to shutdown elements of their RAN and save energy. In exchange, the MNO could then shutdown its infrastructure during another low activity period, and have its users served by another partner MNO remaining on-guard. All partner MNOs could then take turns at each low-activity period to remain on-guard and serve partner MNOs' users. However, it is not evidenced that MNOs would cooperate in such a way, as each MNO might lack incentives to serve other MNOs' users. The purpose of the contribution is then to help MNOs to share their infrastructure, and evaluate scenarii allowing them to cooperate in the best way to maximise energy saving.

Presentation of the proposed infrastructure

To solve the problem, a collaborative framework based on MARL is investigated to allow MNOs to cooperate. MARL is then used for negotiation messages, allowing each MNO to learn a cooperation policy. Then for each low activity period, telcos run their learned policy to negotiate and decide whom should remain *on-guard* during the period. It is then expected that after the training period, MNO should collaborate in a fair way, with the on-guard periods evenly balanced between telcos, and a maximum of energy saved overall. All negotiation messages between MNOs then go through an "*on-guard service*". Following assumptions are then made:

* The service is observed by a "*regulator*", usually a national telecommunication authority like the French Autorité de Régulation des Communications Électroniques, des postes et de la distribution de la Presse (ARCEP). The regulator is then able to set specific rules, and eventually configure blacklisting policies.

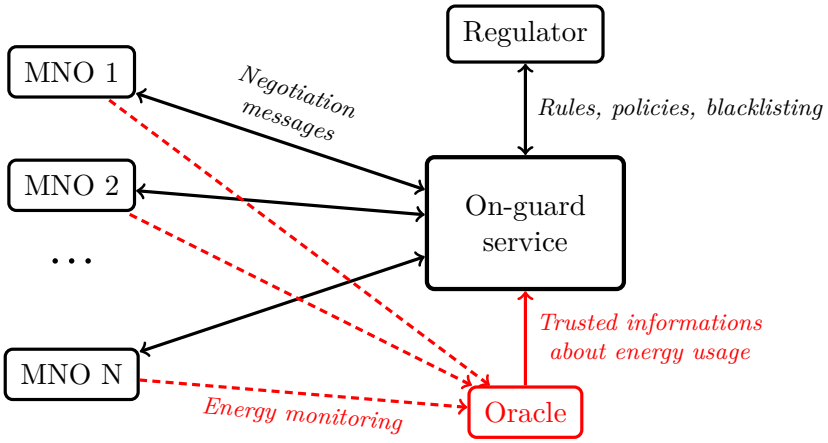


Figure 2.6 – The cooperation architecture

* And a trusted "*oracle*" reports reliable and trusted measurements of the energy (KWh) consumed by MNOs. These reports are mandatory to evaluate the fairest sharing between MNOs. To avoid centralisation issues, this oracle might be implemented using the DLT. This approach of reliable data reporting from a trustless environment is considered later in this thesis, on Chapter 4.

The proposed architecture is illustrated on Figure 2.6. The environment can then be configured in one of the following modes:

* **Free**: the service only relays MNOs' actions, that then decide by themselves which MNO(s) should remain online during the next low-activity period;
* **recommended** : For each new negotiation, the service suggests which MNO should remain on-guard, selecting the one that has least contributed (that has contributed to save the lowest amount of energy). Nonetheless MNOs remain free to follow or not the recommendations.
* **Imposed** : The service enforces offers coming exclusively from the recommended MNO(s). The service also blacklists a MNO refusing to be on-guard more than m times by preventing them to use another infrastructure, and reintegrates it after a blacklisting period is finished.

Following events can then happen on the platform for service negotiation:

* A MNO may place an *offer* (ie it offers to be on-guard for other MNOs)
* A MNO may also send a *demand*, where it requests another MNO to be on-guard (ie for serving its users).
* Then the service sends *observation* messages to MNOs, summarising in a matrix all *offers* and *demands* from other MNOs, as well as a *reward* scalar calculated so that it (proportionally) represents the amount of energy saved (reward is positive) or consumed (reward is negative)

at current state of negotiation. A negative reward is also added everywhere at each iteration to speed up negotiation.

Then MNOs attempt to learn an optimal cooperation policy using reinforcement learning algorithms. Each MNO then uses its learned policies to decide which actions to make, between *offers* and *demands*, based on the *observations* and *rewards*.

A detailed presentation of the model as well as the results are presented on Appendix A. The proposed architecture is run on a simulation environment with either 3, 4, 8 and 10 agents (MNOs) for each modeset (*free*, *recommended* and *imposed*). Once the agents are trained, simulations are run on 100 negotiation, portraying the low activity periods (e.g. nights). Overall results are then extracted using different metrics, illustrating how well did MNOs collaborate. The presented work suggests four metrics (efficiency, safety, incentive-compatibility, and fairness) to determine the most important facets of a successful collaboration from the regulator perspective. To calculate such metrics, it is essential to determinate the amount of energy saved by each player, hence the importance of the oracle reporting reliable operational data.

Results & discussion

The results presented in Appendix A show that MNOs showed little to no collaboration in a *free* mode. This absence of collaboration might be explained by a mistrust between MNOs, as no actor wanted to take much risk in cooperating. On the over hand, in an *imposed* scenario, results show that collaboration did emerge among actors, and lasted over time, as per the amount of energy saved was close to optimum. Results for the *recommended* mode are more mitigated, as the results did not show a general trend on how successful was the collaboration. Yet results seemed to indicate that the more involved MNOs there are, the more successful the collaboration is.

Results thus tend to indicate that a framework is necessary for collaboration to emerge on such an environment, taking the shape of the *on-guard service*. The on-guard service further requires trusted and reliable energy readings from the network infrastructure itself to operate, provided by the *oracle*.

Various implementation choices can be then made to operate the proposed architecture. At first, the on-guard service may be solely operated by a central authority like a regulator implementing service enforcement and MNO blacklisting like suggested by the *imposed* mode. In such a scenario, the oracle could provide the necessary trusted energy data thanks to an IDSA-like architecture. Yet such a deployment relies heavily on trusted third parties (regulator enforcing the environment, IDSA data clearing house, etc.) to operate. They may be costly to operate, and induce single points of failures in the system.

On the other hand, the system could be implemented using the DLT. In this concept, the on-guard service policies can be enforced with the help of smart-contracts automatically making

recommendations and (un)blacklisting. The energy oracle might also be powered thanks to the DLT, as the technology enables the creation of a trusted, easily auditable database that may serve as a "decentralised data clearing house". With such an approach, the role of the regulator would be minimised, and most of the system could be operated in a fully decentralised way. The log of events happening on the system would further remain easily auditable.

The presented work then showed that given the right environment and policy enforcement, MNOs can cooperate in a way to make more efficient networks from an energy consumption perspective. While the results showed the need to implement an environment enforcing cooperation, the latter can be operated in a decentralised way thanks to the DLT. Furthermore, a secure data framework like defined by the IDSA can help providing reliable and trusted energy reports from the environment, and may also be implemented in a decentralised way thanks to the DLT.

The contribution proposed on this section then proposes a framework beneficial to evaluate cooperative scenarios, taking as a use-case infrastructure sharing among MNOs to save energy. While the presented study was limited to energy saving, it could be extended to other metrics like the QoS for a better evaluation of cooperation.

The contribution presented in the next section also introduces a cooperative architecture, extended to any virtualised network infrastructure. The usage of the DLT is further considered in the next contribution to allow the operation of such an architecture in a decentralised way.

2.4.3 Proposal of a fully decentralised marketplace

The growing interest on the DLT has further led to a contribution on an international cooperation with multiple partners during this thesis work, aiming at building a decentralised network infrastructure [7]. This solution further proposes a federated marketplace architecture connecting multiple Communication Service Providers (CSPs), in order to allow the creation of decentralised End-to-End (E2E) network service chains, on what can be seen as "*Connectivity as a Service*".

Motivations

With the evolution of telecommunication networks and new needs and use-cases arising with 5G, the revenue made by telcos is expected to significantly increase [48]. The proposed architecture aims at helping telcos capturing these new revenues. Indeed, making telcos able to share and maximise the usage of their infrastructure will help them to reduce the CAPital EXpenditure (CAPEX) needed to deploy new infrastructures. Furthermore, the virtualisation ("*softwarisation*") and disaggregation of the network infrastructure now allows any E2E connectivity service to be split into multiple infrastructures, eventually belonging to distinct CSPs.

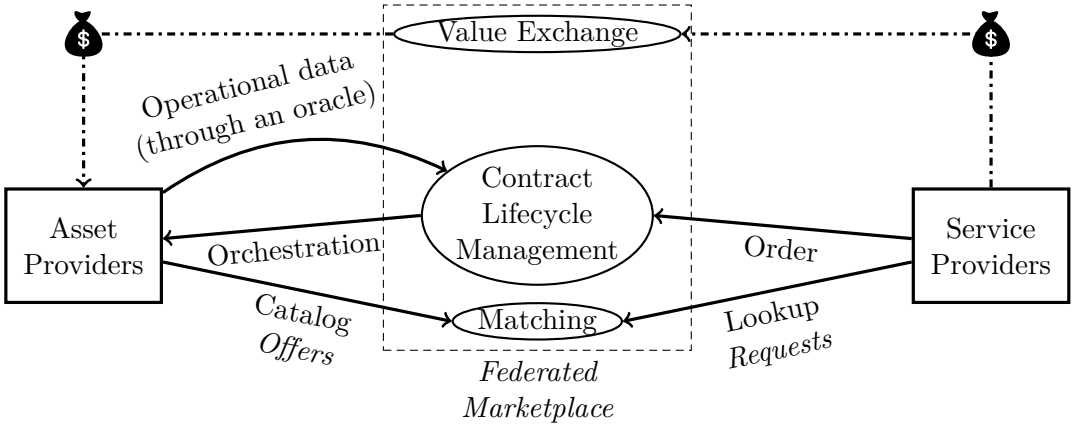


Figure 2.7 – The proposed marketplace architecture

Presentation of the architecture

First, the proposed architecture provides a marketplace environment. This environment, illustrated Figure 2.7 is notably composed of the following elements:

* Multiple CSPs (or *"asset providers"*), capable of providing various resources (Cloud, optical connectivity, ...). They expose the proposed resources and their characteristics into the marketplace.
* Multiple *"service providers"*, looking for specific connectivity resources to deploy their services. As an example, a company requiring a private mobile network might require edge resources to support a virtualised network infrastructure. Furthermore, a telco may need a temporary increase of capacity to match his customers' needs.
* A *search engine* is also provided to allow service providers to search for asset providers fitting their needs.

Such a marketplace can then help all the actors involved in a system to trade resources, e.g any service provider can browse the catalog of asset providers to select infrastructures belonging to different CSPs, fitting his needs.

In this work, the use of the DLT is considered to power this marketplace, as well as the other components in a decentralised way. All tasks can indeed be automated with the help of smart-contracts, while the DLT can store in a trusted way all events happening in the marketplace. Figure 2.8 presents the whole architecture.

Amidst the marketplace-related components (catalog, offers, search engine), the DLT also gives host to the components required for asset onboarding (deployment of virtualised network functions) and management as well as identity management (e.g e-Subscriber Identity Module (eSIM)). The DLT also hosts operational functions to automate service orchestration and service chain design. It is also used to store the negotiated Service Level Agreements (SLAs),

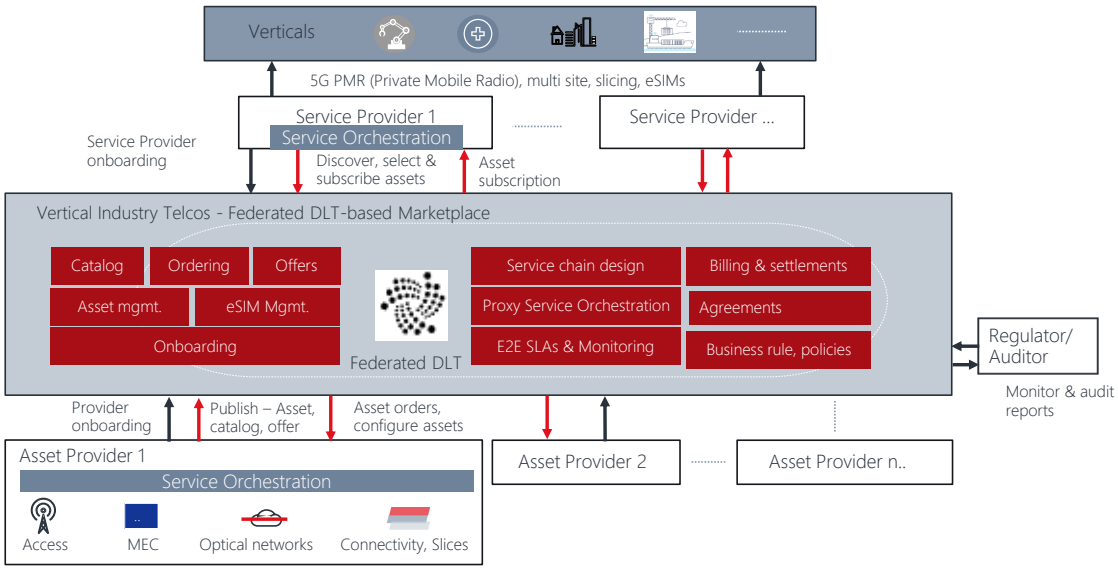


Figure 2.8 – The whole federated marketplace proposed in [7]

the negotiated policies and agreements passed between actors involved in a service chain. Furthermore, the DLT is also proposed to host analytic and monitoring capabilities for the then created services. Accounting and billing may also be implemented and automated thanks to these performance reports, if negotiated in the connectivity SLAs.

Figure 2.9 then gives an example of a service being established using the proposed architecture. In this example, a private mobile network is first requested by a company named "ACME" (step **1**). The request is handled by its local telco (a *service provider*), that then decides to outsource some resources to third-party infrastructure providers, through the marketplace. The telco then uses the search engine to select the appropriate resources (step **2**). Negotiated assets can be of various nature, spanning from "towercos", offering antennas to host a mobile access networks, to telco cloud providers capable of providing an execution environment to operate both access and core network [49]. Service providers might also provide optical network capabilities, with fibers being shared thanks to technologies like WDM. At this step SLAs are also passed between ACME's telco (the *service provider*) and the different infrastructure providers (*asset providers*), describing the delivered services as well as their key characteristics (delivered QoS, billing, location, etc.). Once resources are negotiated, and the SLAs passed between ACME's telco (the *service provider*) and the different *asset providers*, the virtualised network is deployed (step **3**) and then delivered (step **4**).

In this work, the need for guaranteeing the operation of the network according to the negociated SLAs has also been identified. To fulfill this "service assurance" requirement, above all trusted, reliable data on the performance of the connectivity services is required. Hence the

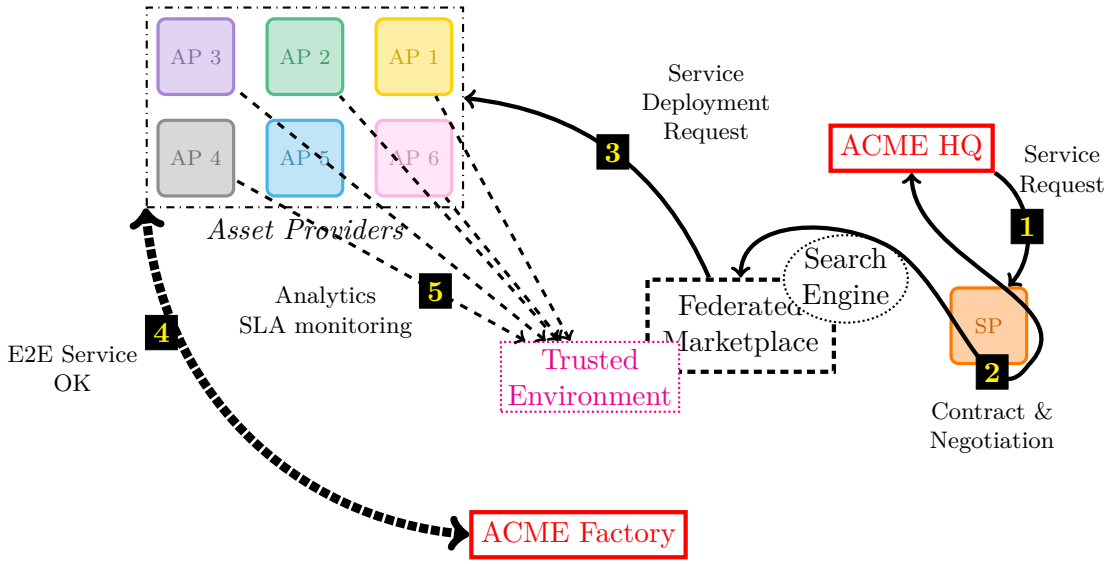


Figure 2.9 – A simplified example of service operation using the proposed architecture

deployment environment is monitored in a trustworthy way to provide service assurance to the service provider, and the final consumer (step **5**). This particular component has led to another contribution [12], covered later on in this manuscript, in Chapter 4.

Technological choices

Open APIs In order for such a system to be successfully implemented, all the actors in place need to implement common Application Programming Interfaces (APIs) in order to successfully perform automation. This project makes use of standardised TMForum APIs [3] to facilitate interactions between the DLT and the components of the marketplace. The standardised APIs can notably manage the whole lifecycle of assets (infrastructures offered onto the marketplace), and their ordering, as well as their orchestration. The usage of standard API vastly facilitates the interoperability of the system, as a given marketplace can then be onboarded on any DLT supporting the standardised interfaces. It further helps the adoption of the marketplace for asset or service providers wanting to join it. The opposite is also true, as a given distributed ledger is then able to interact with any marketplace instance supporting the standardised APIs. Yet to achieve full interoperability, an "*API gateway*" is necessary to translate calls from standardised TMForum APIs to the underlying DLT API calls, and the other way around.

Thanks to this approach and the use of an API gateway, both DLTs and marketplaces are agnostic to the technology used to support the proposed architecture.

3. TMForum APIs can be found online at `https://www.tmforum.org/resources/?yith_wcan=1&filter_document-type=specifications&query_type_document-type=or`

On the DLT usage Choice has been made to use the DLT to support the proposed architecture, as such a technology allows the deployment of such a service in a decentralised way. Indeed, the DLT allows the creation of a fully decentralised trustworthy database, easily auditable. All events happening on the marketplace can be then stored onto a distributed ledger, with the help of the standardised APIs presented above. This then allows to easily track the lifecycle of the connectivity services deployed, and thus detect any flaws or fraud attempts that might occur on such a decentralised system. Furthermore, most of the building blocks of the proposed architecture can be automated and deployed as smart-contracts in the DLT, to be then run on a decentralised environment.

Nonetheless for a successful operation of the marketplace, a "data layer" between the network infrastructure and the marketplace supporting it must be provided, to provide reliable and trusted performance reports of the network infrastructure. Such usage reports are indeed necessary to pro-actively manage created E2E services in a decentralised way, as well as providing "service assurance" to the final customers, testifying on the compliance of the infrastructure with the passed SLAs. Such performance reports may also trigger penalty mechanisms if measured performance do not match SLA constraints, and may also be used for accounting. This topic is covered later in this thesis, in Chapter 4.

2.5 Conclusion

In this chapter, various connectivity-related collaborative scenarii have been explored, as well as improvements of the infrastructure to sustain collaboration. The state of the art shows that novel technologies like the DLT further enhance collaboration between actors not trusting each other. Moreover, multi-agent algorithms can help designing efficient collaboration policies and algorithms. On the matter, on this chapter multiple contributions of this thesis work regarding collaborative architectures have been presented:

* At first, an architecture linking a standard IDSA dataspace with the network infrastructure supporting it is proposed, and patented [4]. This architecture further allows the adaptation of the network infrastructure to enhance the QoS in order to enhance the journey of data through the dataspace. Such a contribution can then help powering the above contributions, as it helps providing a safe infrastructure to allow multiple actors to exchange data, and then build up collaborative services.
* Then a method allowing a decentralised and collaborative time synchronisation protocol has been proposed and patented [6]. This architecture allows multiple peers involved in a collaborative system like an IDSA dataspace or a DLT to safely synchronise their clocks to timestamp data, without having to rely on a single trusted time source. This contribution can thus further enhance trust on collaborative architectures as the time of emission of any data needed

Name	Purpose	trusted data source	Decentralised
Helium [36]	Decentralised wireless connectivity	Helium hotspots with the *Proof of Coverage* protocol	Yes
Ammbr [37]	Collaborative mesh connectivity	Ammbr routers with the *Proof of Velocity* algorithm	No
Inter-MNO energy saving [2]	Mobile infrastructure sharing to save energy	*Oracle* to be defined	?
Vertical Industry Telcos [7]	Federated connectivity marketplace	DLT-based *data layer*	Yes

Table 2.3 – The collaboration architectures explored in this chapter, and their respective trusted data sources

to support collaboration can then be secured.

* Then a collaborative connectivity scenario has also been explored, allowing multiple concurrent MNOs to take advantage of MARL to learn how to collaborate on mobile network sharing. This work has led to a conference paper [2]. This work further showed that MNOs have incentive in sharing their infrastructure to save energy in low-activity periods, in order to be able to shut down under-used antennas.

* And finally, a scenario of a federated CSP marketplace has been explored thanks to a contribution with multiple telco and non-telco partners [7]. The proposed architecture takes advantage of the virtualisation and automation of networks infrastructure making them more elastic and prone to such scenarii, and on the DLT to bring truthfulness between telcos in a decentralised way.

The presented work showed that many of the collaborative connectivity initiatives require trusted operational data (energy reports, performance indicators, etc.) to be successful. The explored collaboration use-cases and their respective trusted data sources are summarised on Table 2.3.

On the thesis contributions presented in this chapter, the proposed federated connectivity marketplace presented in Subsection 2.4.3 requires a "data layer" providing trusted usage reports on the network operation, and the proposed mobile infrastructure sharing scenario presented in Subsection 2.4.2 requires a trusted oracle to report the reliable energy usage of the MNOs involved in the collaboration.

In this thesis work, is it considered to use the DLT to provide such trusted data sources in a decentralised way. The next chapter, Chapter 3 then presents the DLT more in details. This

chapter presents firstly a literature review of the technology and some of its existing variations, and secondly contributions related to the technology made in this thesis work. Then, Chapter 4 presents novel DLT-based architectures to provide such a trusted data source.

Chapter 3

On the Distributed Ledger Technology

Contents

This chapter details more in depth the Distributed Ledger Technology. State of the art technology is described in the first section, while contributions of this thesis are presented afterwards. These contributions are about a simulator of the Tangle, a DAG-based DLT and a decentralised "archiving" system enabling nodes participating in a Distributed Ledger to prune their local storage, thus freeing memory.

3.1 Blockchain-based Distributed Ledger Technologies

This first section presents the Blockchain technology, as well as some of its evolutions and limitations.

3.1.1 The premises of Blockchain

The emergence of the Internet and the associated new telecommunication technologies has led to a deep transformation of our society.

One of the main strength of the Internet is its ability to connect people at a global scale, as worldwide interactions are now greatly facilitated, almost instantaneous and cheap.

With the ability to communicate at a global scale came also the ability to share global value more easily. As an example, the Society for Worldwide Interbank Financial Telecommunication

(SWIFT) enables financial institutions through the world to exchange valuable assets almost instantaneously, and without relying on paper anymore.

This major technological breakthrough also allowed new marketplaces and collaborative models to emerge.

Marketplaces like Uber, Amazon, Ebay, AirBnB etc. indeed enable their users (acting as *"providers"* in this scope) to share valuable assets, or offer services (car ride, flat rental, etc.) through a marketplace and reach other users (acting as *"consumers"* in this scope) buying their goods/services. Here the platform acts as an intermediary connecting offer and demand, and takes advantage of the ability to instantly process transactions between providers and consumers.

Along with this wave of innovation and new business opportunities, smart-contracts have been first theorised back in 1994 [22]. Smart-contracts are infinite state-machines able to automate the lifecycle of any contractual agreement between cooperating players. Agreements passed are translated into a set of policies and rules that are used to design the state machine, and then any event happening within the contract (i.e state change) is recorded through transactions.

One of the most widely used, and most basic smart-contract is digital currency. The Unspent Transaction Output (UTXO) model [50] has been imagined as a way of managing a digital currency and its transactions. This model particularly allows to precisely track the virtual money units ("coins"), thus solving the double spend problem. Figure 3.1 shows an example of UTXO-based transactions. UTXOs are non-divisible amounts of currency, represented as circles in the figure. Transactions ("TX" rectangles) then spend and consume one or multiple of these "unspent outputs" (then called "inputs" of the transactions), and then generate one or multiple new UTXOs out of them (then called "outputs" of the transaction). Multiple outputs are necessary if a transaction has multiple recipients. They can be further useful if one wants to spend only part of an UTXO. Indeed, in that specific case the transaction emitter will need to generate a transaction spending the whole UTXO as input, and generating at least one output of the desired value for the recipient, and one other output containing the remaining value, attributed back to the emitter. This model allows to simplify the verification of transactions, as a double-spend attempt results in a UTXO being used multiple times. Yet this system leads to more complex value exchange, as a simple exchange between two parties can then result into multiple transactions, and multiple UTXOs being spent on the way. It should also be noted that users do not directly own the currency, but rather own one or multiple UTXOs, whose total value represent their balance. This model has successfully been integrated in multiple crypto-currencies [50].

On the other side, many Peer-to-Peer (P2P) architectures have been designed as a way to get rid of single points of failure. This makes some Internet services like file sharing more decentralised than classical client-server architectures, as P2P services are then more resistant to outages.

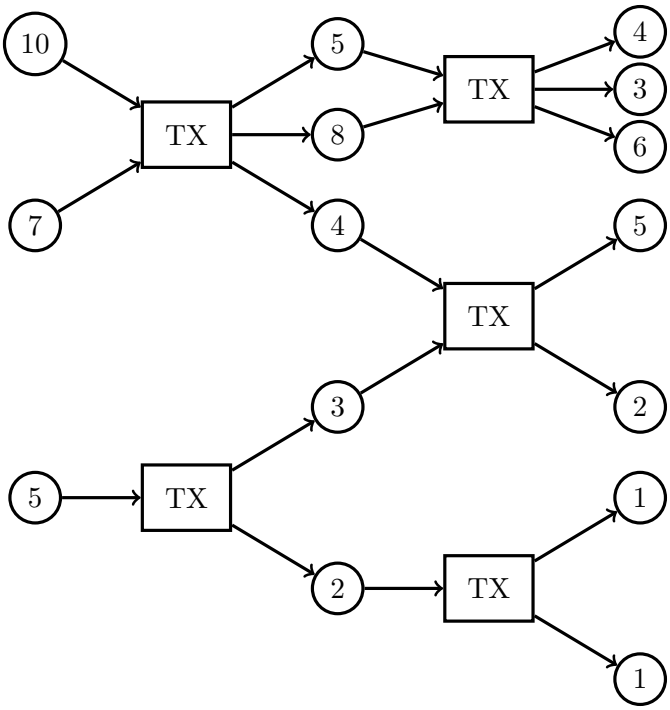


Figure 3.1 – UTXO-based transactions. Circles represent UTXOs as non-divisible amount of Bitcoins, spent in transactions having multiple inputs, and multiple outputs if there are multiple recipients

The Bittorent protocol [51] is an example of a P2P file sharing solution. Files shared thanks to this protocol are not stored into a single server, but rather sparsely shared onto the whole network of users. When first uploaded to the network, a file is split into multiple chunks of data, that are transferred directly to the users requesting the file. Upon download, users are then able to share the parts of the file they have received so far, as long as they are online. Arriving users can then access the file from any other user on the network, given they have first downloaded the file, and they remain online. Such a protocol then makes the shared contents almost undeletable, as multiple copies of shared content exist through the whole network, directly hosted by the users.

Distributed HashTables (DHTs) [52] are yet another example of decentralised storage. They consist in a set of decentralised storage solutions allowing to store data on multiple machines connected to the Internet. DHTs further define algorithms used to locate on which machines a file is stored, eventually split into chunks deployed on multiple machines and/or having multiple replicas scattered through the network. They thus ensure the reliability of data, and eventually guarantee a fast availability to data despite the network latency. They are now found in various applications [52].

All of the cited innovation have led to a proposal of a decentralised digital currency called Bitgold [53]. Bitgold is a digital, decentralised currency designed to be byzantine-fault tolerant thanks to cryptographic mechanisms. It implements a transactional smart-contract similar to UTXO. The technology also first takes advantage of distributed storage capabilities to avoid a central entity to hold the transactions associated with the digital currency service. Furthermore, this design is more particularly based on a Proof of Work (PoW) [54], a cryptographic challenge relying on a one-way function, which requires high computing capabilities to be reversed. Although Bitgold has never overcome the state of design, it is considered as the premises of Bitcoin and so-called crypto-currencies.

3.1.2 Bitcoin, and the first Blockchains

Data structure at a glance

The Bitcoin solution and digital currency has been brought to life back in 2009, with the first white paper released on October 2008 [8], and the first running networks of nodes in 2009. It is described as being heavily based on Sbazo's Bitgold design, as well as various other digital currencies attempts. Bitcoin primarily relies on a **Blockchain**, a distributed transactional database solution relying on blocks of transactions chained with each other, illustrated on Figure 3.2. A block is defined by its **height** in the chain, as the distance from the first block (also referred to as the **genesis**), and by its **depth**, as the number of subsequent blocks confirming it (also referred to as "number of confirmations"). On a given block, transactions are stored in a Merkle tree, a data structure illustrated on Figure 3.3.

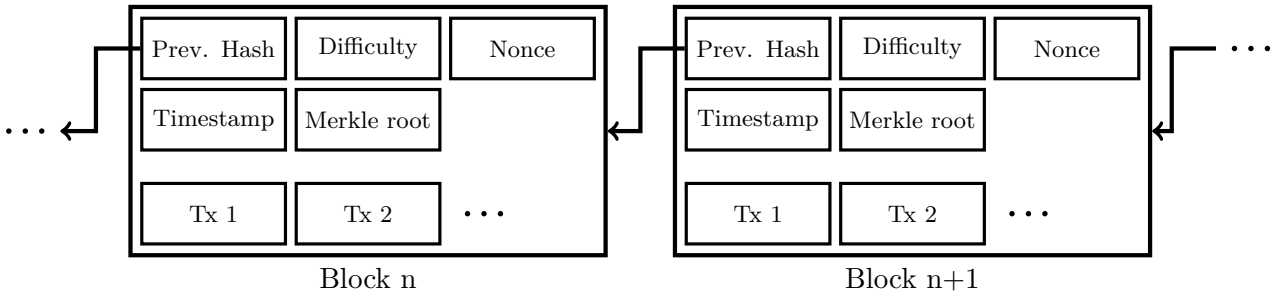


Figure 3.2 – The Bitcoin Blockchain structure

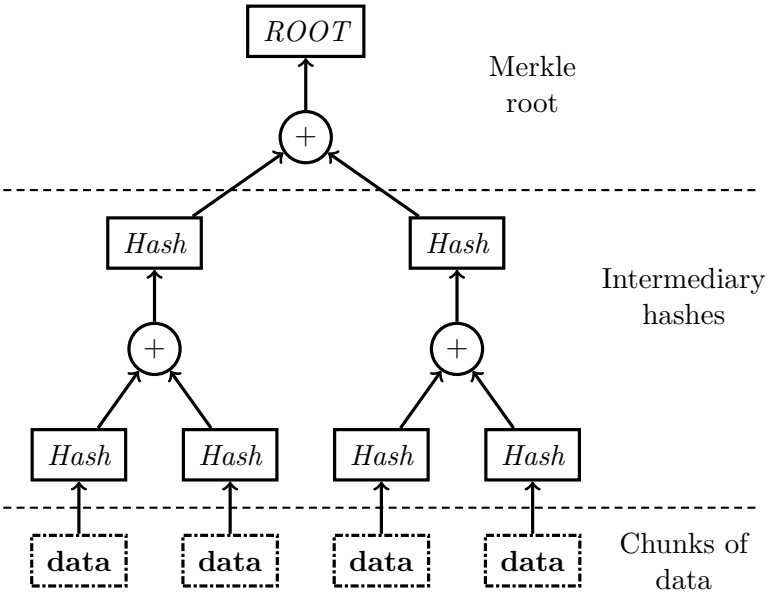


Figure 3.3 – A Merkle tree structure

On such a structure, all chunks of data (transactions in this case) are first put in pairs. A hash of each transaction is calculated, and the hashes of the pairs are concatenated. Then, subsequent hashes of the concatenated pair hashes are calculated. The resulting hashes are then again put in pairs, concatenated, and the process repeats until there is only one hash remaining (the "Merkle root"). As a result, the whole structure forms a tree, with transactions as leaves, hashes as nodes, and a single root which testifies on the integrity of the whole structure. Indeed, any change on the initial data then have cascading repercussions on any hashes directly or indirectly calculated using said data, as a consequence modifying the Merkle root.

Along with their payload, blocks have also a header, containing various types of metadata. Block headers notably hold the previous block header hash as a way to validate former transactions, and the Merkle tree's root. This approach thus ensures that any block validates every of its ancestors, as a single change onto the Blockchain content will then alter this whole chain of interconnected hashes. The structure of a block is further described on Figure 3.2.

The Proof of Work consensus protocol

As the integrity of the data is secured against a chain of hashes, the non-repudiation of the chain must also be ensured as a mean to bring trust into the system. More specifically, the protocol must make double spends as difficult as possible and make transactions unalterable and non-repudiable once committed for implementing successful value transfers. For that purpose, a Proof of Work (PoW) mechanism is implemented for producing new blocks of transactions, whose purpose is to elect a single "leader" producing and proofing a new block.

Specifically, the PoW relies on a hash as a one-way function, easy to calculate but almost impossible to revert.

To commit a new block, one must find a nonce x so that:

$$SHA256\left[H_d(\,T_s, Mk_{root}, Hsh_{n-1}, d, x\,)\right] < t \tag{3.1}$$

With:

* $SHA256$ the Secure Hash Algorithm (SHA) version 2 hashing function generating a hash of 256 bits
* $H_d(...)$ The Block header (here, only relevant parameters are illustrated on the equation),
* T_s a Linux Timestamp,
* Mk_{root} the Merkle root of the block's transactions,
* Hsh_{n-1} the hash of the previous Block Header,
* d a *difficulty* value evolving through time.
* t a target value whom challenge result must be inferior, calculated with the current difficulty.

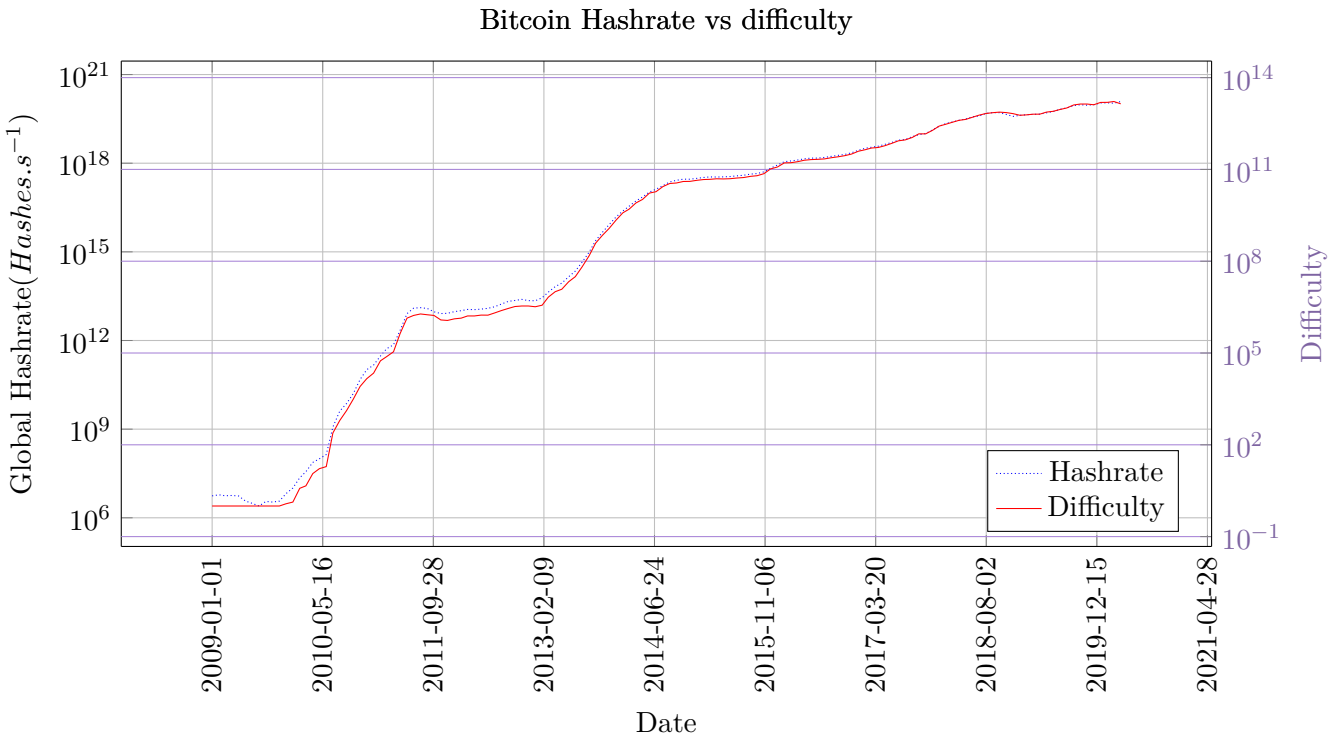


Figure 3.4 – The evolution of Bitcoin's network difficulty and hashrate through time (scale is semilog). Source: `https://data.bitcoinity.org/`

* and x the nonce to find.

Due to the characteristics of the hash function, the only way to solve this dilemma is to actually try out all possible *nonce* (x) values until the challenge is solved. This process then requires a significant computing power. Furthermore, the *target* (t) value directly weighs the difficulty of the challenge, as the number of possible solutions to the challenge decreases when t decreases. The *target* is directly calculated from the *difficulty* (d), an abstract representation about how difficult the challenge is (the lower d is, the easiest the challenge is). This value is dynamic, and adjusted every 2016 blocs, in order to keep the block generation time around 10 minutes.

As illustrated on Figure 3.4, the Bitcoin Blockchain's difficulty has vastly increased, as well as the network's computing power (*hashrate*, in hashes per second). Furthermore, one can notice a correlation between the two values, as the difficulty increases with the available computing power. The PoW thus makes altering the Blockchain contents challenging, as any attacker will need to gather at least 51% of the total network's computing power in order to be able to compromise the Blockchain operation [8].

The lifecycle of a transaction onto the Bitcoin network

The Bitcoin Blockchain is run on top of a peer-to-peer network of nodes. The participants may be put in one of three main categories:

* *"Light nodes"* are simple users of the Blockchain. They are able to produce transactions and sign them, but they will need to rely on other nodes to emit them onto the Blockchain. Nonetheless as they hold the private keys used to produce transactions, they are still in control of their assets. Light nodes do not require any significant resources, and may be hosted on any device.
* *"Full nodes"* are nodes hosting a replica of the Blockchain, connected to the peer-to-peer network, they participate in the validation of blocks and transactions by managing their local copy of the Blockchain. They thus need higher resources than light nodes, especially storage and network access. However they cannot produce new blocks as they don't perform any PoW.
* And finally, *"miners"* are full nodes running the PoW, thus able to produce new blocks. They are however the most resource consuming nodes, as the PoW process ("mining") is resource consuming by its nature.

Figure 3.5 illustrates the lifecycle of a transaction.

When a user wants to spend bitcoin, it generates a signed transaction using his private key, and sends it over to the network using a lightnode (step **1**). The transaction is then intercepted by a miner and put in a queue, then included into a new block. Miners then perform the PoW over the blocks they attempt to create with the transactions they receive. Upon successful creation of a block, a miner will add it to its local Blockchain copy, then broadcast it over the peer-to-peer network (step **2**). Each full node then checks the validity of the block, especially regarding the PoW, and the block's transactions. Upon acceptance, each full node adds the block to its local copy (step **3**).

The specific case of "forks" As the Blockchain is decentralised over a big peer-to-peer network, desynchronisation can occur between nodes. This can lead to multiple blocks being created at the same time, thus having the same height in the chain. The chain then splits into multiple branches. When such a scenario occurs, a miner must choose which branch to validate to continue the chain over. For making this choice, the "longest chain" rule prevails as nodes will try to stick with the branch having the more blocks. Other branches are then discarded, and their transactions put back in the queue, to be incorporated in future blocks. Nonetheless the PoW and the ten minutes block time prevents in many cases such scenarii to happen. The PoW ensures also the stability of the chain as long as the majority of miners act in an honest way to keep the Blockchain running, and use the same set of rules for validating transactions and dealing with forks.

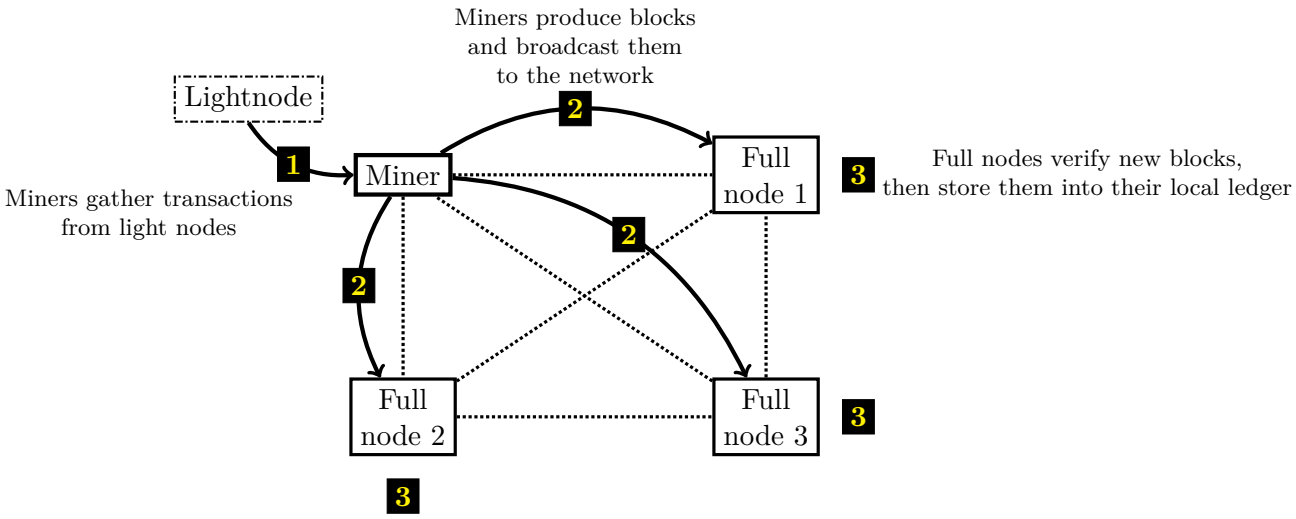


Figure 3.5 – The lifecycle of a transaction onto the Bitcoin network

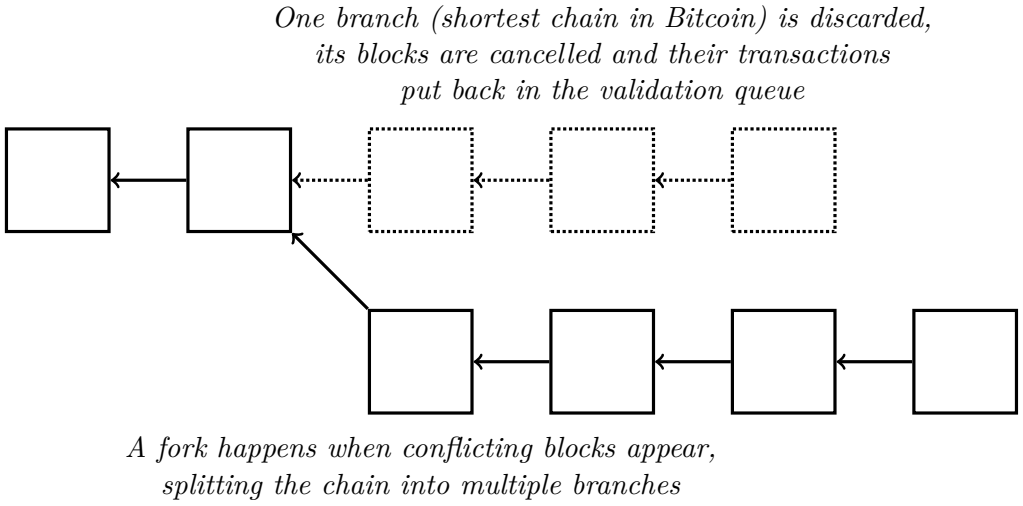


Figure 3.6 – A Blockchain fork

On block/transaction security With the decentralised nature of the Blockchain, the commitment of a block, or the point at which its transactions can be considered as validated, cannot be guaranteed. There is indeed no central authority to testify on the commitment of a block. Their adoption onto the Blockchain is up to the consensus reached by the whole network of participating peers. The latter may choose at anytime to discard, thus not take into account any block they might consider non-valid. Yet the PoW mechanism makes block production resource consuming and difficult. This makes the chain secure as an attacker needs to hold computing power to produce corrupt blocks. As per current implementation of the PoW and the "longest chain" rule put in place, there is a probability for a block to be discarded, eventually allowing double spends to occur. This probability is depending on the attacker available relative power (in fraction of the total available power, considering it being **less than 51%**) [1], and on the block depth. Indeed, the deeper a given block is, the harder it will get for an attacker to build blocks forming a chain long enough to cause a fork and discard the honest blocks. The probability of success of such an attack is also pondered by the attacker's computing power, as thanks to the PoW mechanism, any attacker's available computing power has also a direct influence on the ability to build blocks "fast enough" to overcome the honest branch. As a result, according to [55], this probability can be estimated by the following equation:

$$p = 1 - \sum_{m=0}^{n} \binom{m+n-1}{m} \left((1-q)^n q^m - (1-q)^m q^n\right) \tag{3.2}$$

with p the probability, n the depth of the block, q the fraction of attacker's available power. The discard probability is plotted on Figure 3.7 in relation to block's depth, for multiple value of attackers' available relative computing power.

As illustrated on the figure, as long as the 51% power requirement is met the probability of a block being discarded decreases exponentially with the increase of its depth, to then become negligible. The figure further shows the impact of an attacker's relative computing power. Thanks to this behaviour, it is common practise to wait for some confirmations of a block to consider its transactions secured enough. To summarise this section, although a block is never committed and secured deterministically, the PoW makes the discarding probability decrease exponentially then become negligible, as long as the nodes holding the majority of the computing power are honest.

The Bitcoin incentive model

To make a majority of miners collaborate in an honest way, Bitcoin implements incentive mechanisms. More precisely, any miner creating a block is given the right to create new coins

1. As said in Section 3.1.2, any attacker harnessing more than 51% of the network's computing power is able to fully compromise the Blockchain, hence in this scenario the probability of blocks being discarded is 1 regardless of other parameters

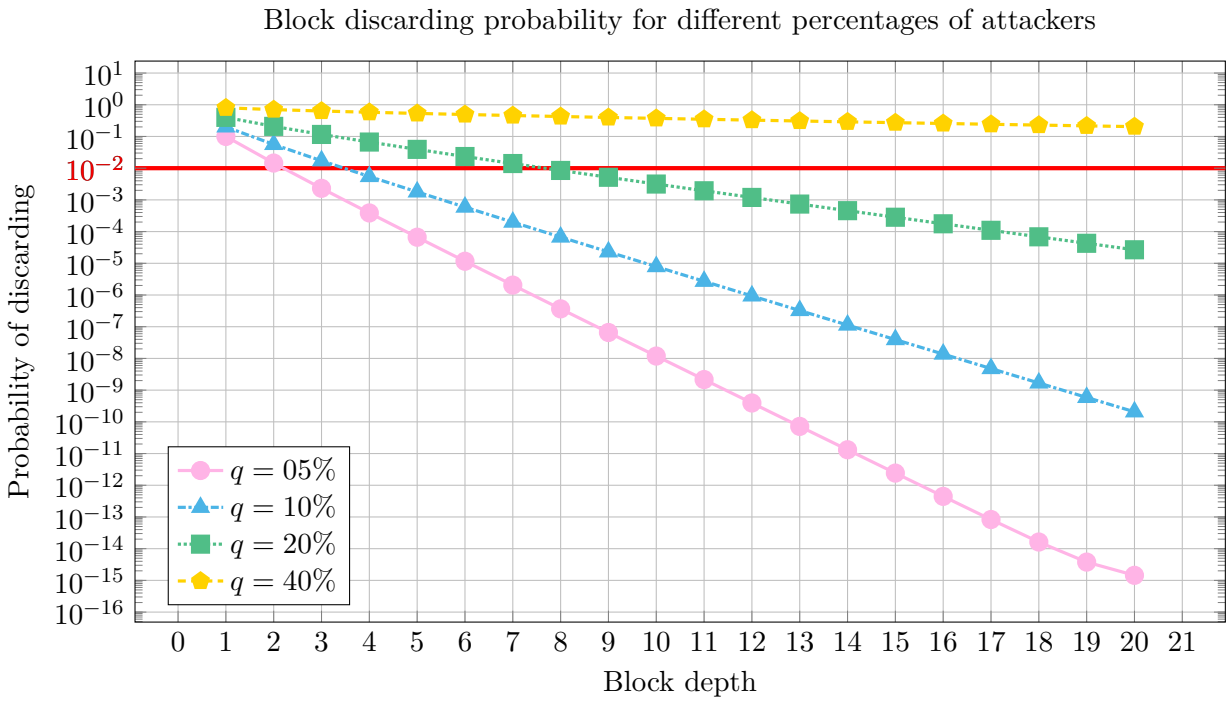


Figure 3.7 – The probability of a block being discarded in relation to its depth (probability scale is semilog)

("mine" them), and spend them as retribution. Furthermore, fees must be paid to emit a transaction on the Bitcoin network. These fees will also serve as a reward for the miner of the transaction's block. It is up to the transaction emitter to choose the exact amount, yet miners usually choose transactions with highest fees to improve their reward. While starting at 50 Bitcoins, the fixed reward for miner is divided by two every 210,000 blocks, or roughly 4 years considering an average inter-block time of 10 minutes. This mechanism has been set up to enforce Bitcoin's price inflation, and to fix a hard limit on the amount of Bitcoins in circulation. Indeed, as the reward is constantly decreasing, it will then eventually disappear when it will drop lower than one "Satoshi" (10^{-8} Bitcoin, the lowest possible division of one Bitcoin) by the year 2140. At this point, the amount of spendable Bitcoin will have reached its maximum amount of about 21 millions. It is also worth noting that miners usually work grouped in "pools" in order to share their computing power, making it more likely for a mining pool to produce new blocks, and then share mining rewards. This whole retribution chain is a way to keep a Nash equilibrium on the network, as each miner has an individual interest in maintaining the system, as it will make more profit in spending the mining reward rather than compromise the Blockchain [56].

Bitcoin is now a major digital currency, whose operation remains at this day fully decentralised. Although its primary use is for crypto-currencies, some users are also able to use the

Blockchain for storing data needing to be secured. This feature is particularly true for "layer 2" systems. "Layer 2" systems are applications and smart-contracts that aren't deployed directly on the Blockchain, but that use its capabilities to achieve truthfulness on data (that may be seen as "digital notary acts") needing to be persisted in a trusted ledger [57]. In the cited example, any data stored into a regular database can be secured by storing its hash on the Blockchain, without having to store the data itself. Such applications may also be referred to as being "off-chain", as they are operated beyond the boundaries of the Blockchain. The concept of "layer 2" is covered later on in this chapter, in Section 3.2.1.

3.1.3 General Smart-contracts integration

While first Blockchains were designed for digital currencies, many initiatives have been conducted to allow and facilitate the automation of assets exchange. Ethereum is the first major Blockchain allowing the creation and operation of customised smart-contracts. Customised smart-contracts not only allow the safe storage of custom data onto a Blockchain, but also allow to set up custom rules for data validation, that nodes can implement while attempting to build new blocks.

For that purpose, users of Ethereum are able to implement and deploy Turing-complete programs implementing their contractual needs, and interact with them by emitting special transactions. Their code is thus executed by full nodes and miners creating/validating blocks. As a consequence, as the processing of a given smart-contract transaction should give the same result regardless of the node executing it, a smart-contract must have a deterministic behaviour. Such applications are said to be "on-chain", as they are fully run within the boundaries of the Blockchain, with all nodes participating on it.

To accommodate smart-contracts, the original Ethereum design is similar to Bitcoin, with few differences:

* The PoW uses a customised hashing function called "Ethash";
* Block reward for miner is constant (never halved);
* Block time is lessened to 5 seconds to improve transaction throughput and reduce transaction processing speed;
* And in order to fairly reward the miners executing the smart-contract, a "gas" system is implemented, as an abstract representation of the computing power consumed per transaction. This value is fixed and hardcoded for classical value transactions, while it is calculated upon execution for smart-contract transactions, using the bytecode operations performed as a reference. Each operation performed (addition, comparison, etc.) is attributed a fixed amount of gas, then used to calculate the required gas for a given transaction. In a similar way to Bitcoin, transaction emitters can modify a "gasPrice" value to hold control on transaction

fees. They then pay an amount of "gas × gasPrice" Ethers that miners get as extra reward.

The operation of smart-contracts is then made easy, with dedicated programming languages and compilers, and a full Application Programming Interface (API) to interact with them. Some notorious smart-contracts have vastly spread as they demultiplicate the possibilities and uses of Blockchain, by allowing the creation of **decentralised applications**. In this thesis a decentralised application is defined as an application running in a fully decentralised way, by using a set of smart-contracts on the Blockchain.

The Ethereum Request for Comments #20 (ERC20) standard [58] provides a set of guidelines for allowing the creation of "tokens", digital currencies implemented as applications on the Ethereum public Blockchain. Nowadays many ERC20-based tokens are deployed on Ethereum for various purposes and initiatives. These tokens are deployed and managed thanks to smart-contracts implementing the ERC20 guidelines.

Smart-contracts have also permitted the rise of Non Fungible Tokens (NFTs). NFTs are digital assets testifying on an asset property with the Blockchain, taking the shape of unique, non-divisible tokens. They can then be used as a proof of ownership of various valuables like images, music, etc. Similarly, it can be implemented on Ethereum Blockchain using smart-contracts. For that purpose, the ERC721 standard can be used to implement NFTs [59].

Nonetheless the immutability of the Blockchain also makes smart-contracts potential targets for attackers. Indeed, once a contract is deployed on the Blockchain, it is impossible to stop, making it vastly vulnerable to failures or various attacks. As an example, a decentralised application named "TheDao" has been subject to a major attack back in 2016, where an attacker has been able to exploit a vulnerability in the smart-contract code to steal one third of the funds of the smart-contract. This has led to a "hard fork" of the Blockchain, as the Ethereum foundation decided to revert the attack by updating the node software. However, this decision was controversial and other members of the community have decided to continue over, thus creating a separate currency called "Ethereum classic" [24].

For some decentralised applications, further issues arise when an **oracle** is required. An oracle is a trusted entity, having knowledge about events happening outside of the Blockchain. They may then be required for some decentralised applications whose purpose goes beyond the boundaries of the Blockchain. Examples of such applications include car sharing, sport bets, flat rental, etc. More notably, the need for oracle has been presented in Chapter 2, where in the presented network-related collaborative systems [2, 7], such a trusted data source is required to testify on the accuracy of operational data.

Adding an oracle to a decentralised application should only be considered with care, as such an entity acts then as a single point of failure, that may compromise the smart-contract - and then the Blockchain - if compromised. However, truthfulness mechanisms can be implemented to solve the oracle problem, usually by involving multiple actors. Examples of such mechanisms are

Helium's proposal of a Proof of Coverage [36], or the Torcoin's proposal of a Proof of Bandwidth (PoB) [60].

3.1.4 Evolutions of the Nakamoto protocol

From the initial Nakamoto proposal, various innovations have been proposed to further enhance many characteristics of the initial PoW-based Blockchain technology. Performance improvements are more particularly considered, to make the Blockchain operate faster, more efficiently, and to mitigate resource consumption while allowing the technology to scale better. In this section, various improvements are discussed, covering fork resolution alternatives, and alternative consensus protocols to the PoW.

Forks Management

At first, Ethereum reduced inter-block time from 10 minutes to 5 seconds to improve the time required to process a transaction. This came with the cost of a higher fork probability. Indeed, [61] has shown that on Bitcoin-like Blockchains, the probability of a fork happening increases as the inter-block time decreases.

This has led Ethereum to implement a novel rule replacing the Bitcoin's "longest chain" principle, called GHOST [62]. This protocol notably allows miners to take into account "uncle blocks". These blocks are orphaned blocks not taken into account in the main chain, but still taken into account for difficulty calculation and miner reward. As a miner of an "uncle block" gets a reward, the purpose of these blocks are to keep giving incentives to miners on producing new blocks, despite the higher discarding probability. This change of protocol also allows to increase the maximum transaction throughput, known as the Transactions per Second (TPS) rate, from 7 TPS to 15 TPS.

On the Proof of Stake and other PoW alternatives

Furthermore, alternate consensus mechanisms have been proposed to replace the resource and energy-consuming PoW. At first, vote-based consensus mechanisms have been proposed as an alternative. However, as suggested by [9], these protocols are prone to centralisation as the number of nodes allowed to vote on new blocks/transactions are by nature limited. Vote-based consensus mechanisms are implemented mostly on Blockchain technologies designed to be used by a consortium of fixed actors/privately, implying a limited number of nodes. Hyperledger [63] or Corda [64] are example of such "consortium" Blockchain technologies. They may also be implemented on Blockchain publicly available for use, yet whose governance remains limited to a defined set of nodes, like Ripple [65] or Stellar [66].

Name of project	Name of protocol	PoS flavour	Status	Nothing at stake resolution
Ethereum 2 [62]	Casper	PoS	In development, with test network	Ether staking/fraud detection
EOS [67]	dPoS	Hybrid voting/PoS	In production	Centralisation, limited number of validators
Cardano [68]	Ourobouros	Delegation with staking pools	In production	Ledger audit/incentives
Algorand [69]	PPoS	Weighted lottery	In production	VRF/Balance-based weight

Table 3.1 – Proof of Stake implementations

Going back to proof-based consensus mechanisms, the most popular alternative is the Proof of Stake (PoS). With the PoS, the chance to create a new block isn't anymore driven with the computing power of the node, but rather by his stake on the Blockchain. The stake is an abstract concept, that can, depending on the implementation, include for example the validator's balance, the validator amount of transactions being emitted, etc. The exact definition of a PoS indeed varies from implementation to implementation.

[9] suggests that PoS-based systems make also forks less likely to happen, thus allowing to decrease the inter-block time and improve performances.

There are however various issues arising on PoS systems. The "*nothing at stake*" is one of the main issues. It is indeed theorised that if a fork happens, the Nash equilibrium gets ruptured as it can be of a validator interest to maintain the two parallel chains instead of choosing a resolution, whereas in the case of PoW they are more incited to choose a single branch to avoid splitting their computing power.

Various solutions have been proposed to overcome this issue, depending on the actual implementation of the PoS mechanism. Table 3.1 lists some of the main PoS implementations.

Among these implementations, Casper was the first proposal of a PoS, to power Ethereum 2 [17]. It is implemented as a Solidity smart-contract. Casper solves the nothing at stake problem by making validators stake Ethers to validate a block. As the whole process is auditable on the Blockchain, one can easily detect if a given validator has staked Ethers on conflicting blocks thus creating a fork, and if such an event happens other validators are able to destroy the staked Ethers.

Another solution already implemented in projects like EOS [67] is the delegated Proof of Stake (dPoS). In this approach, the PoS is limited to a finite list of approved nodes, chosen and voted by the rest of the network. As a result, this approach induces centralisation in a similar way to Ripple [65].

The Cardano project also came with a PoS-based consensus protocol named "Ourobouros" [68]. This protocol is somewhat hybrid between a delegated Proof of Stake and a standard one like Ethereum's. Validators are placed in "staking pools" whose participants can delegate their

validating power to a limited set of nodes, "*delegates*". However, compared to EOS the list of *delegate* nodes isn't fixed, and the validating process by itself is closer to bitcoin mining pools. An incentive scheme is put in place to prevent attacks, and make nothing at stake less likely to happen.

Finally, Algorand introduced a "Pure Proof of Stake" (PPoS) protocol [69]. Similarly to other protocols, a voting-based proof of stake is used to produce new blocks. However, voters identities aren't fixed and are selected among everyone using a Verifiable Random Function (VRF), a weighted lottery where users having the greatest balance have more validating power. As a result, this protocol isn't bound to a limited number of nodes as per the dPoS. The nothing at stake problem is here solved thanks to the VRF continuously generating random committees, and the weight based on account balance.

While PoS-based consensus mechanisms allow improved performances and efficiency, the Blockchain structure itself is also limiting by nature, due to the chain being able to grow only in a single direction, and nodes to be kept in sync to avoid forks and their potentially resource-consuming resolution.

As a result, improvements and novel data structures are being considered to improve or replace the Blockchain. These alternatives are covered in the next section.

3.2 Blockchain alternatives and evolutions

3.2.1 Interconnected Blockchains, Sharding & Layer 2

This section first presents improved Blockchain architectures, aiming at improving the technology while still keeping Blockchain or Blockchain-like data structures.

Avalanche

Avalanche [70] is yet another DLT initiative. It first uses a novel, PoS-based consensus mechanism as an attempt to solve the DLT trilemma between security, scalability and decentralisation. When a validator broadcasts a transaction, it first samples a random set of validators for agreement, that then repeat this sampling procedure (they "gossip" the transaction) until ultimately the network as a whole validates the transaction, thus reaching consensus. Then, for the sake of better scalability, Avalanche uses three interconnected Blockchains, respectively for value exchange, smart contract execution and validator coordination. Furthermore, the Avalanche ecosystem allows users to create an unlimited number of customised inter-operable Blockchains ("subnets") to enhance performance. The usage of an enhanced data structure compared to regular Blockchains thus allows further improvement of the technology and its performances.

Ethereum 2

A **sharding** approach is considered for Ethereum 2, in a similar way to Avalanche's "subnets". It is indeed planned to split the Blockchain into multiple instances (the "shards"), piloted by a main "beacon chain", managing the consensus. At first, the shards will only serve as distributed databases for the main Blockchain and won't be able to handle currencies or smart-contracts. Yet they will still improve the general performance, as Ethereum will then implement a "rollup", i.e a protocol to bundle transactions and merge them into a single transaction summarising all the previous ones. Raw transactions can be then held in shard, and be validated and proofed thanks to the produced "rolled-up" transactions then emitted on the beacon chain.

Lightning, an example of a layer-2 solution

"Layer 2" applications have also been imagined, deployed as applications overlayed on existing Blockchains/DLTs. Such applications aim at enhancing existing DLTs, by interacting with them and taking advantage of their truthfulness capabilities, while running in a separate environment. From a technical side, a layer 2 application should thus be agnostic to the underlying DLT used. The interaction between a layer 2 application and the underlying DLT is then analog to the interaction between layers of a typical communication network (e.g the Internet Protocol (IP) can be overlayed above either Ethernet or Wifi protocol, etc.), hence the terminology of a "higher level" protocol. As a typical example, the Lightning network [57] is a solution built above the Bitcoin Blockchain, aiming at improving its state-of-the-art performances. Lightning is described as "a network of micro-payment channels". In a similar principle than Ethereum 2, users of the lightning network create direct micro-payment channels and attribute them some funds. Participants in a given channel then interact directly, without using the Blockchain to exchange resources. Then, upon closure of the channel, a single transaction summarising all exchanges is then broadcasted to the main Bitcoin chain. It is worth noting that a channel works only if its users trust each other's balance during its operation, as micro-transactions are then not proofed with the Blockchain before channel closure.

3.2.2 On Distributed Ledger Technologies based on Directed Acyclic Graphs

Whereas "enhanced" Blockchain architectures have been presented in the previous section, alternate solutions to the state of the art Blockchains are also explored, going beyond the Blockchain structure to further enhance efficiency and performances. These novel technologies imply novel data structures based on Directed Acyclic Graphs (DAGs). Such solutions cannot be then considered as pure Blockchains, and will hence be referred to as Distributed Ledgers.

Nano

Nano [71] is the oldest proposal of a DAG-based Distributed Ledger, initially named "rail-blocks". To hold transactions, the Nano ledger uses a "block lattice" structure, that can be seen as an ensemble of multiple interconnected Blockchains.

More precisely, each account has its own Blockchain, and only one account's holder can modify it. Two transactions are then required to transfer funds, on the two implied account holders (one for *sending* and one for *receiving*). These individual transactions work in a standalone way, and are directly chained together without being aggregated into blocks. With such a structure forks should not happen. They are indeed quickly resolved as only a given account holder can modify its chain. If a fork does happen due to a malicious user trying to perform a double-spend and revert a transaction, a voting-based consensus is set up by other nodes (named "representatives") to solve it. Vote weight is computed using voters balance as reference, hence close to PoS-based systems.

As a result, the fate of the ledger lies in the "representatives" nodes that share voting weight based on their balance. It is also worth noting that a PoW is implemented in Nano for emitting transaction. Yet this PoW is only used as an anti-spam tool. It thus requires less power than the Bitcoin PoW as its purpose is different.

According to the creator, the Nano technology may scale to $10,000$ transactions per seconds. Figure 3.8 illustrates the Block Lattice proposal: Each account holder has his own Blockchain, where each transaction sent/received results in a block. The account chains are then linked together, as a "send" transaction on a given account chain will result in an associated "receive" transaction on another chain as the funds are then transferred.

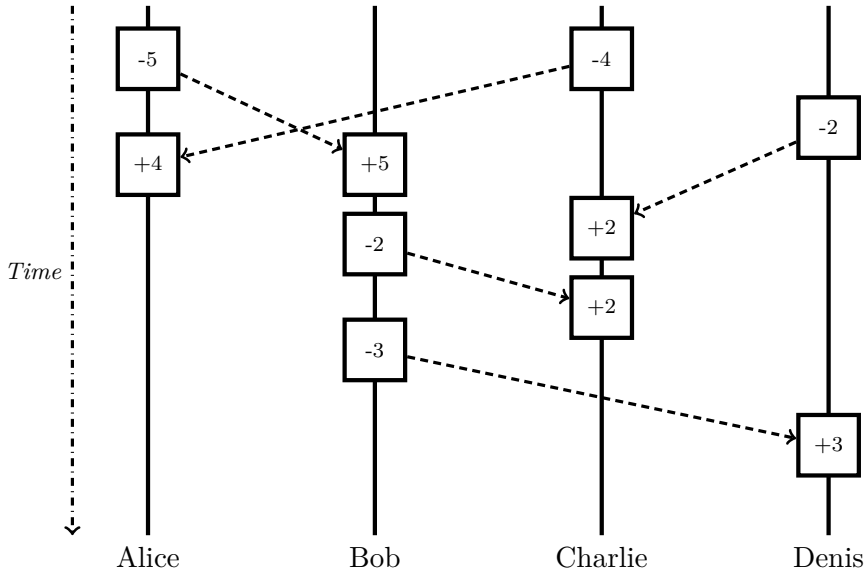


Figure 3.8 – The Nano's "Block Lattice"

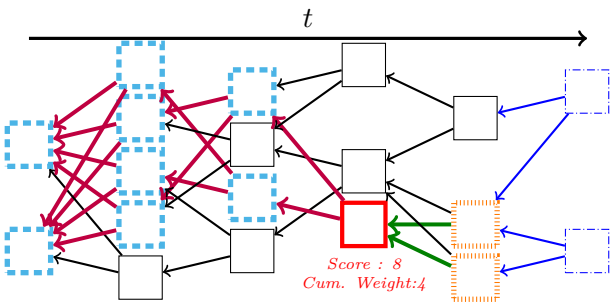


Figure 3.9 – The Tangle structure

Tangle

Presentation of the Tangle model An other DLT arose back in 2016, called the "Tangle" [11]. This technology has purposely been designed for the IoT world, requiring to process a vast amount of micro-transactions without using too many resources. Similarly to the Nano's block lattice, transactions work in a standalone way. However, a pure Directed Acyclic Graph (DAG) is used to store transactions, without any Blockchain-like structure. The graph is rather organised so that every transaction is attached to (and then validates) two of its predecessors. When a user produces a transaction, it will choose on the Tangle two transactions out of the list of non-validated ones (referred to as "tips" of the Tangle).

As a result, the DAG does not grow in a deterministic way like a Blockchain, but rather in a stochastic way as attachment possibilities for transactions are multiple. Yet Tangle's creators suggest that the graph should remain stable over time, and that the number of unconfirmed transactions ("tips") should not escape to infinity. Furthermore, the IoTa foundation suggests that the security of the technology lies in its Tip Selection Algorithm (TSA), the algorithm nodes use to select two tips at a given time [11].

Figure 3.9 gives an overview of the Tangle data structure. As illustrated the first key property of a given transaction is its **score**, representing the total number of prior transactions referenced by the transaction. Then another key property of a given transaction is its **cumulative weight**, representing the total amount of subsequent transactions referencing it, hence validating it. As an example, the thick red transaction has a score of 8 (dashed blue transactions), and a cumulative weight of 4 (dotted orange + the two "tips" thin dashdotted blue transactions).

Thanks to its asynchronous behaviour, the scalability of the Tangle structure is theoretically unbounded. Yet contrary to a Blockchain-like structure it may be difficult to testify on the commitment of a given transaction as multiple branches of the graph can coexist, and their convergence is non-deterministic. It is suggested that the Tangle should remain stable over time, and that every new transaction should indirectly validate any transaction old enough after a finite time, called the "adaptation period". When watching a specific transaction, this

convergence can be observed at the point when each new transaction indirectly validates it. Still an attacker may perform a double spend by various techniques, all implying to make the branch containing the legitimate transaction orphaned.

As a result, mechanisms need to be set up to ensure an adequate security of the transactions, and make them robust to attacks.

Tangle 1 & Coordinator On the first version of the Tangle, each transaction is given a weight based on the difficulty of the PoW used to produce them. The cumulative weight of a transaction is then calculated by summing all the individual weights of subsequent transactions validating it.

A Markov chain Monte-Carlo algorithm is then used as the TSA used to select the two tips, to make it highly unlikely for nodes to select malicious tips. This algorithm is further detailed in Subsection 3.3.2. Indeed, to cancel transactions and make a branch of the Tangle orphaned, an attacker will need to create a branch "heavy" enough to get his tips more likely to be chosen by the TSA, and hence requires computing power.

However, an attacker can compromise the ledger by making a double spend by owning only 34% of the total network's computing power (instead of 51% for regular PoW-based Blockchains). Furthermore, this assumption is made given that *honest nodes continuously emit transactions*. While this is usually true for PoW-based Blockchains where (possibly empty) blocks are emitted at constant time intervals, on the Tangle transactions are standalone and users are likely to emit them only when needed. Due to this weakness of the protocol, the IoTa foundation has decided to implement a "coordinator" centralised node. This node emits special trusted transactions called "milestones" at regular time intervals that help the Tangle to grow properly, and secure it. As a result, any transaction indirectly referenced by a milestone can be then considered as "committed" as it is taken into account by the coordinator. Yet this architecture poses serious evident centralisation issues, leading the Iota foundation to update their protocol to remove the coordinator [25].

Tangle 2 Iota 2 is the next, coordinator-less version of Iota, bringing many new innovations to overcome previously discussed security problems [25].

In this version, consensus becomes vote-based in a similar way to Nano. However, all nodes may participate in the vote when required, thanks to an algorithm called "Fast Probabilistic Consensus" allowing nodes to quickly make decisions by communicating with their neighbours. In the Tangle structure, the purpose of the consensus vote is to decide which transactions are considered valid, as unlike a regular Blockchain, per design, the Tangle may hold conflicting transactions.

To weight the votes, a specific token is implemented within the Tangle, named "mana". The purpose of this token is to act as a reputation system and as a pure representation of the stake

of the node. The system is designed so that the more a node contributes to the network, the more mana it holds. There are some key differences between mana-based consensus and pure token-based consensus:

* Users generate "pending mana" at a constant rate, proportional to their stake (i.e token balance). However pending mana cannot be used immediately as tokens will need to be spent to unlock it.
* Mana and pending mana are decaying exponentially over time, to prevent a node for holding too much weight in the vote.

The cumulative weight of a transaction is then represented by the total mana percentage of transactions referencing it, hence helping application designers to assess the validation of it.

Yet the IoTa 2 protocol is still at design stage as some parameters still require fine tuning.

Hashgraph

Hedera's Hashgraph is another DAG-based DLT, mainly designed for B2B agreements [72]. The Hashgraph presents similarities with Nano's block lattice, as each validating node holds a chain of events ("hashes") that are interconnected together. Yet by nature, the Hashgraph is more centralised due to its underlying consensus mechanism. Indeed, an Asynchronous Byzantine Fault Tolerant (aBFT) mechanism is used to solve conflicts and then participate in network consensus, instead of Nano's voting-based mechanism or IoTa 2 fast probabilistic consensus. This mechanism is more centralised by conception, as only a limited set of nodes are able to participate in consensus. On the other hand, for Nano everyone may host a "representative" node and participate in the consensus, and for IoTa every node participates in voting with a weight equal to node's mana. Any user willing to host a validating node must be approved by the Hedera council and must submit to advanced requirements [2]. This makes Hedera Hashgraph solution closer to solutions like Stellar or Ripple discussed above, that enables better performance and security, while being more centralised to achieve these goals. Furthermore, unlike main DLT projects, the Hashgraph consensus remained copyright-protected by a centralised governance until the beginning of 2022, thus preventing the use of the technology for private purposes until recently.

Table 3.2 summarises the DAG-based DLTs discussed above, as well as their key characteristics.

While still at experimental stage, the Tangle (in version 2) appears as the most decentralised solution, as all nodes participate in the consensus. Performances are also fairly good compared to other decentralised alternatives like Nano's Block Lattice. Furthermore, this solution has been able to gain maturity over time thanks to its relatively old first implementation, and has

2. Whom can be checked on the foundation website as `https://docs.hedera.com/guides/mainnet/mainnet-nodes/node-requirements`

Name	**Max throughput**	**Consensus**	**Consensus nodes**	**Transaction validation**	**Status**
Block Lattice	10,000 TPS	Vote-based	hierarchical (representatives)	Approval weight	In production
Tangle (1.0)	1000 TPS (observed)	PoW & Specific TSA	Centralised (coordinator)	Coordinator milestones	In production
Tangle (2.0)	$>$ 10,000 TPS (observed) [3]	Vote-based	All nodes	Approval weight	Prototype
Hashgraph	Very high in lab (500k TPS)	Asynchronous BFT	Limited set of nodes	Comittee validation	In production, open-sourced very recently

Table 3.2 – A comparison of different data Structures

been fully open-sourced from the beginning, compared to Hashgraph. All the above reasons have led to consider and focus on this technology for this thesis work.

3.3 Modelling the Tangle's Directed Acyclic Graph

While the Tangle proposal is promising, its asynchronous nature makes modelling needed to assess the behaviour of the resulting graph. Indeed, unlike a Blockchain system behaving in a deterministic way, validation of transactions and consensus are reached in a stochastic manner as attachment possibilities for new transactions are multiple. This has led to the design and implementation of a Tangle DAG growth simulator in this thesis work, based on state-of-the-art models. In this section various state-of-the art models of the Tangle are discussed, then the contribution on the modelling and simulation is presented.

3.3.1 The Tangle graph model

The Tangle data structure is represented as a DAG, whose vertices represent standalone *transactions*, and edges represent *validations*. Let us denote the graph as $\mathcal{G} = (V, E)$, with V the set of vertices, and E the set of directed edges. The structure is arranged so that any transaction validates two of its (possibly equal) predecessors, hence forming a DAG structure. It is then defined that $\forall u, v \in V$, u transaction (directly) *validates* v ($u \rightsquigarrow v$) if $(u, v) \in E$. In this section, the terms *transactions* and *vertices* are then equivalent, as well as the terms *validations* and *edges*. Both terminologies will then be used interchangeably.

Let us further define:

$$\forall u \in V; \mathcal{A}_u = \{v_i \in V : \exists (u, v_i) \in E\} \tag{3.3}$$

as the set of transactions that are *directly validated* by transaction u. For any transaction u, let

us further denote:

$$\forall u \in V; deg_{in}(u) = \#\{e = (v1, v2) \in E : v2 = u\}; \tag{3.4}$$

$$\forall u \in V; deg_{out}(u) = \#\{e = (v1, v2) \in E : v1 = u\}; \tag{3.5}$$

with $\#\{...\}$ denoting the cardinal of a set. Or by equivalence, $deg_{in}(u)$ represents the amount of edges pointing to the vertice u, and $deg_{out}(u)$ the amount of edges originating from the vertice u.

Let then $\mathfrak{G} \subset V$ be the set of *genesis* vertices, such as:

$$\forall v \in \mathfrak{G}; \mathcal{A}_v = \emptyset \tag{3.6}$$

$$\forall v \in \mathfrak{G}; deg_{out}(v) = 0 \tag{3.7}$$

$$\forall v \in V \setminus \mathfrak{G}; deg_{out}(v) = 2 \tag{3.8}$$

Any given transaction is then considered as "*validated*" if the vertice $v \in V$ representing it has at least one incoming edge, that is to say $deg_{in}(v) > 0$. It is further considered that any transaction $u \in V$ *indirectly validates* $v \in V$ if there exists a sequence of vertices $u = x_0, x_1, ..., x_k = v$ such as $x_i \in \mathcal{A}_{x_{i-1}} \forall i \in \{1; ...; k\}$, that is to say there is a directed path from u to v. Let us define $\mathcal{W}_v \subset V$ as the set of all transactions indirectly validating v. The *cumulative weight* of a transaction $v \in V$ is then defined as follows:

$$\forall v \in V; \mathcal{H}_v = 1 + \#\mathcal{W}_v \tag{3.9}$$

Let us then denote $\mathcal{T} \subset V$ the set of *unvalidated transactions* (or *"tips"* of the Tangle), defined as follows:

$$\forall v \in \mathcal{T}; deg_{in}(v) = 0 \tag{3.10}$$

And, by consequence:

$$\forall v \in \mathcal{T}; \mathcal{W}_v = \emptyset \tag{3.11}$$

$$\forall v \in \mathcal{T}; \mathcal{H}_v = 1 \tag{3.12}$$

The Tangle is then modelled as a stochastic process, whose state at time $t \geq 0$ is defined as $\mathcal{G}(t) = (V(t), E(t))$, with $V(t)$ the set of vertices (*"transactions"*) and $E(t)$ the set of edges (*"validations"*) at time t. Regardless of further modelling considerations, the process is then

globally defined as follows:

* The initial state of the process (at $t = 0$) is defined as such:

$$V(0) = \mathfrak{G}, E(0) = \emptyset \tag{3.13}$$

As a consequence:

$$\mathcal{T}(0) = \mathfrak{G} \tag{3.14}$$

* As per its "add-only" nature, the Tangle can only possibly *grow* with time, meaning:

$$\forall (t_1, t_2) \geq 0, t_2 > t_1; V(t_1) \subset V(t_2) \tag{3.15}$$

$$\forall (t_1, t_2) \geq 0, t_2 > t_1; E(t_1) \subset E(t_2) \tag{3.16}$$

As time goes on, any addition of a transaction at time t is modelled as the addition of a vertice u, and of two edges (u, v) and (u, w), with $v, w \in V(t)$ two prior transactions already existing:

$$V(t+) = V(t) \cup \{u\} \tag{3.17}$$

$$E(t+) = E(t) \cup \{(u, v); (u, w)\} \tag{3.18}$$

It should also be noted that any new transaction u is a tip, and that the validated transactions v and w can no longer be tips as they get *validated*:

$$\mathcal{T}(t+) = (\mathcal{T}(t) \cup \{u\}) \setminus \{v; w\} \tag{3.19}$$

This model is then applied to the following sections, where different growths models are considered.

3.3.2 Continuous time model of the Tangle

The initial Tangle White Paper [11] did suggest a model of the growth of the graph as a continuous time stochastic process, behaving as follows:

* Arrival of new transactions is represented as a Poisson point process of rate λ, so that:

$$\forall t > 0; \mathbb{P}\{\#V(t) = n\} = \frac{(\lambda t)^n}{n!} e^{-\lambda t}. \tag{3.20}$$

* It is assumed here that a "random, uniform" Tip Selection Algorithm (TSA) is used for adding new transactions. That is to say, any new transaction will attach itself – and then validate –

to two (possibly equal) unvalidated transactions ("tips") chosen randomly and uniformly in the pool of unvalidated transactions $\mathcal{T}$.

* It is yet also assumed that there is a constant latency h upon arrival of new transactions. As per the author suggestion, this latency should reflect the time necessary to compute a new transaction, as well as the time a transaction needs to propagate through the network. This necessary implies that at the addition of the transaction u at time t, edges are added as follow:

$$E(t+) = E(t) \cup \{(u, v); (u, w)\} : v, w \in \mathcal{T}(t - h) \tag{3.21}$$

Then, an hypothesis has been formulated stating that the amount of tips $L(t) = \#\mathcal{T}(t)$ should have a stationary distribution and then fluctuate around a constant value L_0 without escaping to infinity. L_0 is then estimated in [11] as follows:

$$L_0 = 2\lambda h \tag{3.22}$$

Authors then suggested that this condition allows the Tangle to reach consensus given a blind validation of every transaction regardless of their validity. Assumption is indeed further made that given stationarity, any transaction will at some point be validated by every arriving new transaction, after a time called the "adaptation period", hence making the Tangle reach consensus. This however implies that double-spends and invalid transactions may exist in the graph, and thus need to be detected at application level. Indeed on this model, the TSA is agnostic to the transactions themselves as their actual content is not considered. This adaptation period, representing the time needed for the Tangle to reach consensus over a transaction, is estimated as follows:

$$t_0 \lesssim 2.84 \times h \ln L_0 \tag{3.23}$$

However, this assumption of stability based on the stationarity of $L(t)$ seems to be only the author's intuition, rather that the result of a formal demonstration. This assumption seems based on the fact that whereas the total number of transactions continuously increases due to the Poissonnian arrival, the number of unconfirmed ones stabilises. Yet many studies of the Tangle consider this value and its stationary behaviour as a key characteristic of the Tangle and its stability [26, 73–77].

In this time continuous model, the stationarity of $L(t)$ has not been formally demonstrated, although it has been observed through numerous simulations [74–76]. Furthermore, the analysis have been made assuming a random, uniform TSA algorithm, whereas [11] suggests using a "*Markov Chain Monte Carlo*" TSA in the first version of IoTa, to make the protocol more proof to Byzantine attacks. This TSA consists in a random walk of multiple "*particles*" from "*old*" transactions of the Tangle (the definition of "old" is at the user's discretion, yet he may choose

the genesis as the starting point), and walking their way towards the tips. Tips selected by the algorithm are then the ones on whom the particles land. A particle makes a step by selecting, for each transaction x, a transaction y validating it ($y \rightsquigarrow x$). The transition probability from a transaction x to a transaction y is then given by the following equation:

$$\forall x, y \in V : x \in \mathcal{A}_y; \mathbb{P}_{xy} = \exp\left(-\alpha\left(\mathcal{H}_x - \mathcal{H}_y\right)\right)\left(\sum_{z:z\rightsquigarrow x} \exp\left(-\alpha\left(\mathcal{H}_x - \mathcal{H}_z\right)\right)\right)^{-1} \tag{3.24}$$

With $\mathcal{H}_i$ the cumulative weight of the transaction i, and $\alpha > 0$ a parameter to be chosen. This TSA then tends to select the "heaviest" branch of the Tangle, which needed the highest effort to be constructed as per author suggestion. The system can be then secured assuming an honest majority [11].

Furthermore, works have showed that this TSA proposal leads to a Nash equilibrium as any user should implement it for a successful commitment of his transactions [26, 77]. However, it has been shown through simulation that the graph might loose stability and the number of tips grow linearly over time with some large values of α [75]. This TSA is no longer considered for the second version of the Tangle. Indeed, as described in Section 3.2.2, an alternate, voting-based mechanism is used instead. Following modelling considerations then assume the use of a random, uniform TSA.

3.3.3 Tangle discrete-time model

To address the flaws in the demonstration of the continuous-time model, an alternate, discrete time model has been proposed, where stability is demonstrated with a Markov Chain with a stationary distribution [26]. In this approach, time is discretised to fixed time instants ("rounds"). For each round $t \in \mathbb{N}$, a batch of transactions $\mathcal{N}(t)$ is created:

$$V(t) = V(t-1) \cup \mathcal{N}(t) \tag{3.25}$$

$$E(t) = E(t-1) \cup \{(u, v); (u, w) : u \in \mathcal{N}(t); v, w \in V(t-1)\} \tag{3.26}$$

The amount of created transactions $N(t) = \#\mathcal{N}(t)$ follows a Poisson distribution of parameter λ:

$$\mathbb{P}\{N(t) = n\} = \frac{\lambda^n}{n!}e^{-\lambda} \tag{3.27}$$

The new transactions will then randomly attach at tips as seen on the previous round $t-1$, without considering any further delay:

$$\forall u \in \mathcal{N}(t); \mathcal{A}_u \subset \mathcal{T}(t-1) \tag{3.28}$$

As a result, the evolution of the tips of the Tangle can be estimated as follows:

$$\forall t \in \mathbb{N}; \mathcal{T}(t) = [\mathcal{T}(t-1) \cup \mathcal{N}(t)] \setminus \mathfrak{C}(t) \tag{3.29}$$

With $\mathfrak{C}(t) \subset \mathcal{T}(t-1)$ the set of tips being validated by the transactions arriving at time t. By equivalence, by defining $C(t) = \#\mathfrak{C}(t)$:

$$\forall t \in \mathbb{N}; L(t) = L(t-1) + N(t) - C(t) \tag{3.30}$$

The estimation of $C(t)$ can be modelled as a sum of "balls into bins" problems, with, for each possible value of $N(t)$, $2N(t)$ balls thrown into $L(t-1)$ bins as each new transaction validates two (possibly equal) prior transactions. $C(t)$ is then estimated as the amount of bins that are not empty. This discrete approach allows the number of tips of the Tangle to be modelled with a Markov Chain, with an infinite number of states $N \in [1, +\infty[$ representing the number of tips at a given round (if at the round t the current state is N, $L(t) = N$). The transition probability between two states N and N' is then given by the following equation:

$$\mathbb{P}_{N \to N'} = \sum_{k=|N'-N|}^{N'} \frac{\lambda^k e^{-\lambda} N!}{k! N^{2k} (N'-k)!} \left\{ \begin{matrix} 2k \\ N - N' + k \end{matrix} \right\} \tag{3.31}$$

With $\left\{ \begin{matrix} a \\ b \end{matrix} \right\}$ denoting the Stirling number of the second kind $S(a, b)$.

Then it has been demonstrated that this Markov chain has a positive stationary distribution π verifying $\pi_N = \sum_{i \geq 1} \pi_i P_{i \to N}$, hence making the Tangle stable. An approximation of the stationary distribution is performed in [26], resulting in an average number of tips $L_0 \approx 1.26\lambda$. Such a number has also been found by simulation of the presented discrete time model in [76], where the author suggests this difference with the continuous time models is due to the divergence between the two models. Indeed, the discrete time model does not take into account the delay needed to process a transaction, as it remains fixed at one round.

3.3.4 Tangle sampled time model, and simulations

This divergence between multiple models of the Tangle, as well as the lack of formal demonstrations of the Tangle properties in continuous time has led to the definition of a "sampled" time model of the Tangle in this thesis work. Simulations have further been conducted based on the proposed model, using operational parameters representing the "data layer" use case of this thesis presented in Section 4.3 [12]. These simulations showed similarities with the continuous time model [11], regarding the evolution of the number of tips over time. Results then seem to confirm Popov's initial assumptions of a "stable" Tangle (interpreted as a number of tips following a stationary distribution). They also suggest the Tangle to be a good candidate for

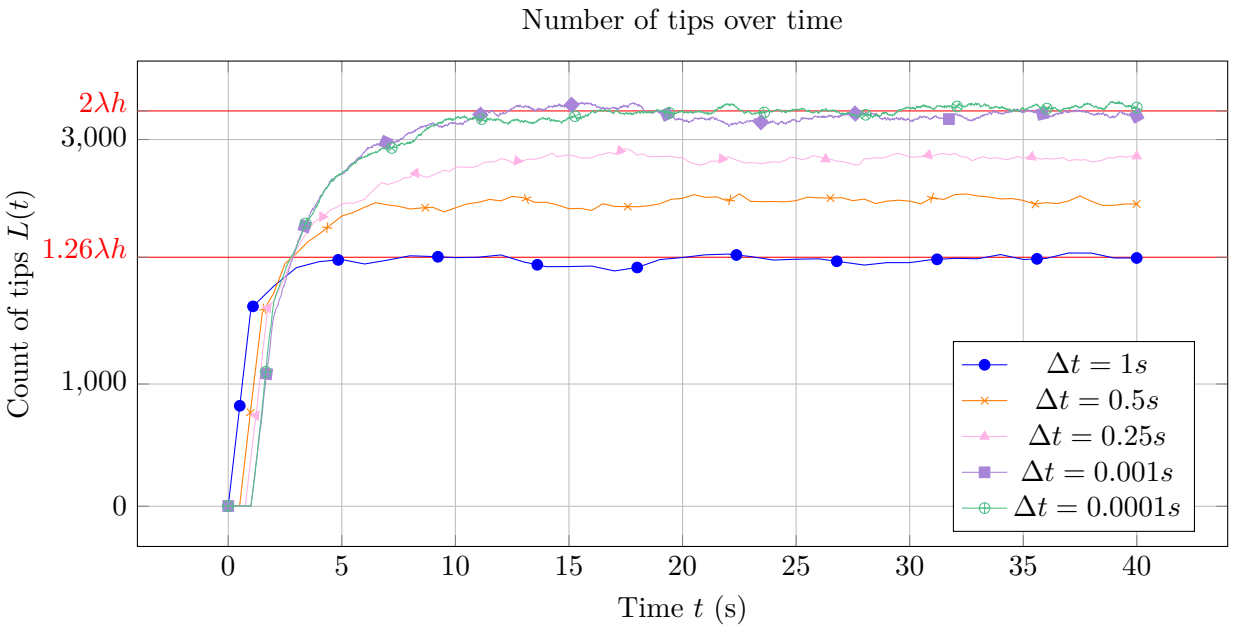


Figure 3.10 – Simulations of the number of tips over time using $\lambda = 1617tx.s^{-1}, h = 1s$ for different values of Δt

the use case proposed in Section 4.3, as stability is observed with the operational parameters that have been estimated on this work [12].

The proposed model takes also place in a discretised setup ($t \in \mathbb{N}$), and works as follows:

* A parameter $\Delta t \in \mathbb{R}^{+*}$ is introduced, as the interval between two time instants.
* For each time instant, a random number of transactions are created, following a Poisson distribution of parameter $\lambda.\Delta t$:

$$\mathbb{P}\{N(t) = n\} = \frac{(\lambda.\Delta t)^n}{n!} e^{-\lambda.\Delta t} \tag{3.32}$$

* The h parameter is reintroduced, so that transactions created at round t select their tips in the list of unvalidated transactions known $h/\Delta t$ rounds ago. This implies that $h/\Delta t \in \mathbb{N}$:

$$\forall u \in \mathcal{N}(t), \frac{h}{\Delta t} \in \mathbb{N}; \mathcal{A}(u) \subset \mathcal{T}\left(t - \frac{h}{\Delta t}\right) \tag{3.33}$$

The growth of the Tangle based on this sampled model is then simulated using the parameters identified in Chapter 4 for this thesis use case ($\lambda = 1617tx.s^{-1}; h = 1s$), and for multiple values of Δ_t ($1s; 0.5s; 0.25s; 0.001s; 0.0001s$). Simulations are run over a time frame of $40s$.

The evolution of $L(t)$ over time is then extracted. Results are displayed on Figure 3.10.

Using $\Delta_t = 1s$, one can observe $L(t)$ to grow and then fluctuate around $1.26\lambda h$. In this

Δt	$1s$	$0.5s$	$0.25s$	$0.125s$	$0.001s$	$0.0001s$
Mean	2015	2497	2848	3040	3205	3232
Std. deviation	37.78	31.73	25.32	53.15	43.66	38.75

Table 3.3 – Analysis of the stationary regime of $L(t)$ for different values of Δt

specific case, as $\Delta t = h = 1$, the simulation is equivalent to the discrete Tangle model [26, 76]. Indeed, in this case, $\lambda.\Delta t = \lambda$, and $h/\Delta t = 1$. The value extracted then matches the values found in the literature, either by former simulations [76] or by a formal analysis [26].

As the Δ_t value is decreased, and thus the sample rate is increased, $L(t)$ is still observed to reach and fluctuate around a constant value. Yet this value tends more and more with $2\lambda h$ as Δt decreases. This value was previously suggested on the continuous model [11], and then shown through simulation afterward [75]. It should be noted that, for values $\lambda.\Delta t << 1$, the Poisson distribution on each round then approximates a Poisson point process, as $\forall t \in \mathbb{N}; \mathbb{P}\{\#V(t) = n\} = \frac{(\lambda.\Delta t.t)^n}{n!} e^{-\lambda.\Delta t.t}$. All of these simulation thus tend to confirm former assumptions of a stationary number of tips, regardless of the model. These results also suggest that the simulations match the continuous model, for small values of Δt.

A analysis of the stationary regime, arbitrary determined as $\forall t > 12s$ is then performed. Table 3.3 gives means and standard deviations for each analysed value of Δt, while Figure 3.11 gives the estimated probability density functions.

Graphically, the value of Δ_t seems to have no incidence on the variance of $L(t)$ during the stationary regime. Furthermore, the calculated standard deviation remains relatively small compared to $L(t)$ ($\sigma < 60$). However, the differences between observed standard deviations calculated for all values of Δ_t look totally random. It should be also noted that although for all values of Δt the distribution seems graphically well-centered around a central value, there seems to be noise on the estimated distributions, especially for high values of Δt. This noise, as well as the observed fluctuations of the standard deviation might be coming from the simulation environment.

Yet, with the proposed input parameters any sample period lower than $\Delta_t \leq 10^{-4}s$ seems to be precise enough to successfully simulate the continuous model, as the observed $L(t) \approx 2\lambda h$.

3.3.5 Discussion & conclusion

In this section, the modelling of the Tangle has been presented. More notably, a contribution of this thesis work on simulation of the Tangle's DAG has been presented. The proposed simulator takes advantage of the state of the art to model the Tangle's graph as a stochastic process. Simulations have been conducted for the sake of evaluating the technology when confronted to the use-case presented in Section 4.3. The work presented on this section then allowed to validate the Tangle for the certification of connectivity-related operational data (performance metrics),

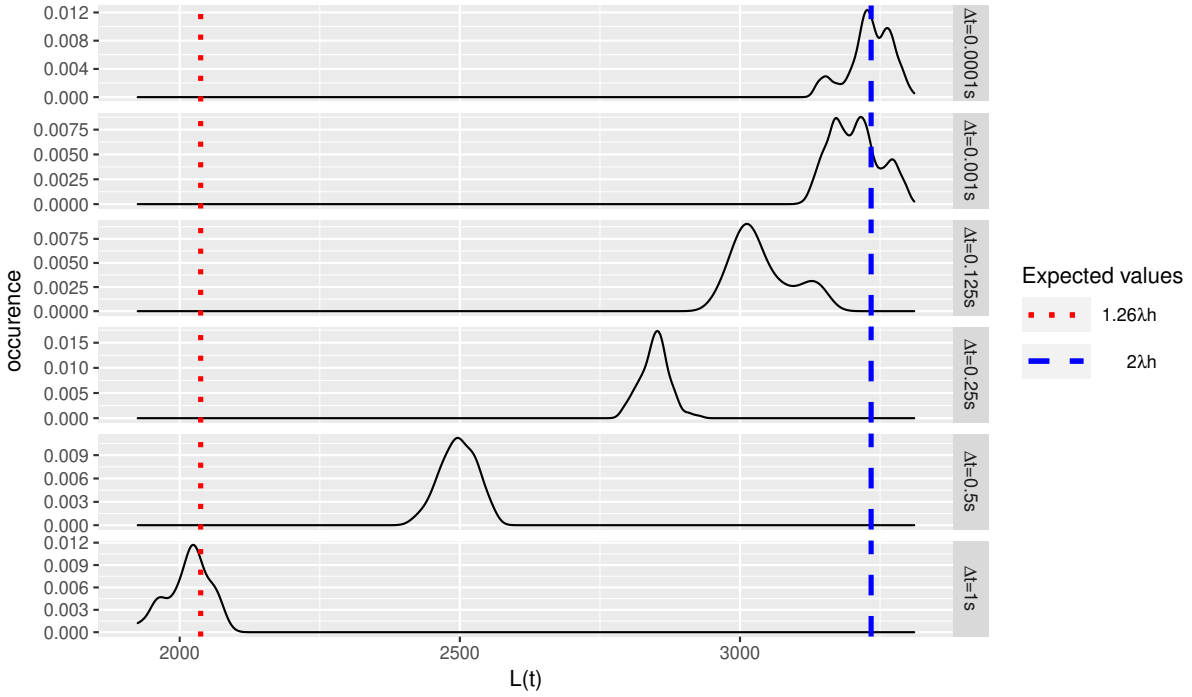


Figure 3.11 – The estimated probability density function of $L(t)$ after stabilisation ($t > 12s$), for each value of Δ_t

coming from a multi-actor infrastructure to its collaborative network management system.

As portrayed in this section, the modelling of the Tangle's DAG is yet complex despite simple hypothesis. The asynchronous nature of the Tangle makes it indeed behave in a stochastic way, rather than being deterministic like a Blockchain with only one possible direction to grow. This characteristic can be detrimental to the adoption of the technology, as any application designer can only approximate the global commitment of a transaction by monitoring its cumulative weight as proposed by [11]. Unlike a Blockchain block's depth, the cumulative weight of a transaction might not be enough to consider a transaction validated with a sufficient level of confidence, for in a realistic, fully public environment it is suggested that Byzantine attacks may occur if a majority of transactions emitted does not come always from honest participants. This flaw implies that honest nodes may need to *continuously emit transactions* to ensure the safe operation of the Tangle. Such a need may actually induce an increased energy consumption, whereas the technology was primarily designed for improved efficiency compared to other Blockchain-based system. As a result, on the current live implementation of the Tangle, a centralised *Coordinator* has been set up, whose purpose is to send trusted transactions named *milestones* [25]. Milestones then currently help to prevent attacks, and are used as signals for application developers as they then consider every transactions indirectly validated by a milestone to be successfully committed [25]. This has then lead the IoTa foundation to propose a second

version of the Tangle, where a voting-based mechanism is introduced for conflict resolution, whereas the MCMC TSA is abandoned [25].

Nonetheless, the current implementation of the Tangle could be implemented as a "layer 2" system, aside an existing DLT. In that specific case, the other DLT could produce the milestones actually necessary for securing the Tangle, while the Tangle could be used to quickly process a huge load of data, thus enhancing performances of the underlying DLT. In that specific case, the Tangle could then serve as a secure, trusted data source for the other DLT, as it provides an easily auditable database whose whole content would get cross validated thanks to the DAG structure, and secured, hence adding extra security compared to systems like lightning channels.

Given this use case, state of the art literature seems to suggest that the stationarity of the number of tips is a sufficient condition for a successful operation of the Tangle, and to further make it stable. The simulations presented in this section tend to support the initial assumptions of the Tangle continuous model, as the number of tips does not escape to infinity, and seems to follow a stationary distribution. Furthermore, the approximation of the continuous time model of the Tangle by sampling the time seems to give accurate results, as the measured average number of tips is close to the value previously determined [11], while initial simulation parameters (λ, h) reflect the operational needs presented later on this work, on Chapter 4. Further work on the modelling/simulation of the Tangle could be performed to fine-tune the result. As an example, further work could be achieved on the delay consideration. Firstly, multiple "classes" of delays could be considered to account for the vast variety of devices producing transactions, with different computing powers and thus different times needed to produce the transactions. Secondly, network propagation delays between the multiple nodes could also be better handled as the distance between nodes is not the same.

3.4 Storage, and transaction archiving

Although many issues and challenges regarding the DLT have been addressed above, one of the main problem of such technologies is their actual transactional, "add-only" nature, as this implies that the amount of storage needed for a distributed ledger is constantly growing, regardless of the technology.

This section first presents existing solutions aiming at mitigating the impact of transaction storage on DLT nodes. Yet these solutions either pose centralisation issues, or do not take into account the operational need to keep a transaction record for some use cases. Then a novel "decentralised archiving" method is presented. The presented contribution aims at enabling partners involved in the operation of a decentralised application to select "*archivists*" of their choice to delegate the storage of old transactions. The proposed method then helps the partners to prune old transactions of their local copy of the ledger, at the reception of trusted signals

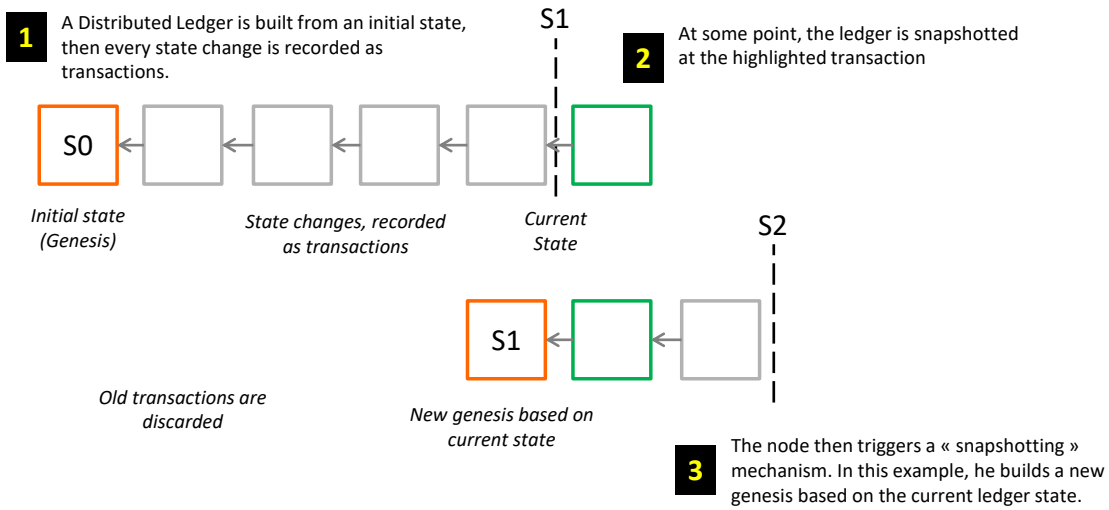


Figure 3.12 – An illustration of snapshotting for any transactional ledger

from the archivists. This contribution has led to a patent [13].

3.4.1 Existing solutions

The impact of Distributed Ledger on storage has been discussed since the dawn of the DLT. On the Bitcoin original paper, Nakamoto suggests that the Merkle tree design allows nodes to free storage as they can keep only the Merkle root, without breaking the chain anyhow. This reduces the growth of the chain to 4.2 MegaBytes per year. As per author's suggestion, this growth is easily overcome by the improvement of storage capabilities as per Moore's Law [8]. On application level, a Distributed Ledger can indeed be seen as an infinite state machine, whose transactions keep a record of every state changes that have happened. In the case of cryptocurrencies, the ledger transactions then represent actual value exchanges, while the state of the ledger then reflects the balance of every account (or the ownership of every UTXOs if applicable). The state of a ledger at any given time instant can then be calculated by sequentially exploring all transactions of the ledger, starting from its genesis to the desired time instant. It is then possible for any node participating in the ledger to save the ledger state in cache at any time ("*snapshot*" the ledger), and discard prior transactions. The resulting snapshotted state can then be used as a new base to participate on the ledger and validate subsequent transactions. Figure 3.12 gives an insight about the process of snapshotting a Distributed Ledger, regardless of the technology used.

Yet the actual behaviour in current Bitcoin implementation is to keep all transactions in storage, which leads to a rapidly increasing amount of space necessary to store the Blockchain.

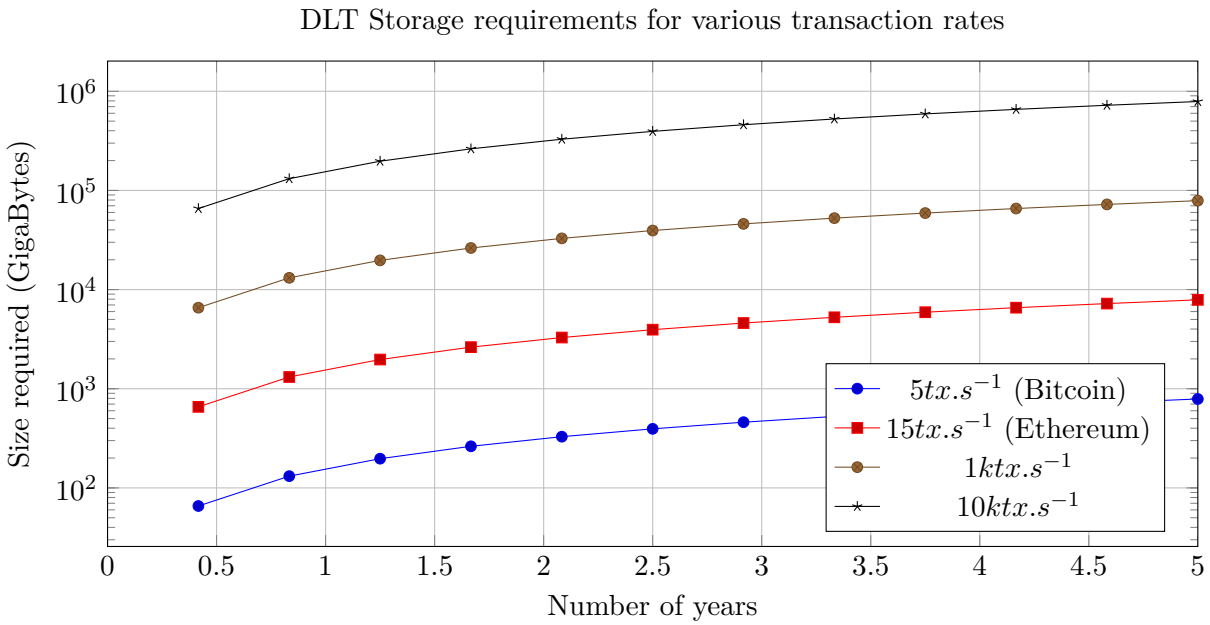


Figure 3.13 – The estimated evolution of DLT storage requirements for different effective transaction throughputs, assuming an average transaction size of 500 Bytes

It is suggested in [78] that nodes participating in Bitcoin consensus lack enough trust to actually discard historical transactions. Yet authors suggest the ever-growing size of the Bitcoin Blockchain is concerning, not only about the amount of storage required to hold the Blockchain, but also for node initial synchronisation as the new node needs to download and check the whole Blockchain, starting from genesis. This then has a negative impact not only on bandwidth, but also on computing power as nodes then have to check and process every single transaction of the chain.

One can expect this issue to become worse and worse as DLTs possible transaction throughput is increasing. Figure 3.13 illustrates the DLT storage requirements for various transaction rates. This study assumes an average transaction size of roughly 500 Bytes as observed in Bitcoin explorers (like `https://tradeblock.com/bitcoin/historical/1h-f-tsize_per_avg-01101`), assumes the ledgers to be continuously operated at full speed, and ignores any other data like Block headers. According to this figure, while systems like Bitcoin would use one TeraByte of data during 5 years, systems scaling to 10k TPS would require almost one PetaByte of storage for the same duration.

In [78], various approaches are also discussed to overcome this drawback. One of the discussed approach is **trust delegation**. Some nodes (e.g. Bitcoin "light nodes") don't hold the whole transaction history, and delegate this feature to nodes with higher capacity (e.g. Bitcoin "full nodes"). Such a solution has also been implemented in the first version of IoTa, as milestones

emitted by the coordinator allow nodes to snapshot ledger state and prune prior transactions. Yet this feature is made available due to the centralised nature of the first Tangle. This approach then poses evident centralisation issues, as per the dependence in fixed trusted third parties (full nodes, coordinator, etc.).

Another approach discussed in this work is **state-based synchronisation**. This mechanism implies that not only transactions are secured into blocks/transaction headers, but also a representation of the ledger state. For that purpose, a *"snapshot"* of the ledger is included in every block/transaction, and then secured into the ledger using regular techniques (such as hash, Merkle tree, ...). This feature has already been implemented into Ethereum as to prevent nodes from computing the ledger state using all historical transactions, and thus save performance. As discussed in the same paper, state-based synchronisation is pretty common in DLT implementations as a way to free space by discarding historical transactions.

The authors in [78] also propose a solution named "CoinPrune" as a state-based synchronisation solution overlayable as a layer 2 application on any DLT. Snapshots of the ledger state are made at a regular time interval, to allow nodes to prune any former transactions. Furthermore, at initial synchronisation a newcomer node implementing CoinPrune can then retrieve a snapshot from another node also implementing CoinPrune, rather than having to retrieve and process the whole transaction history. The node may then only download former block headers to participate in the ledger operation.

Yet historical transactions might also be required for legal and auditing purposes for some DLT use cases. As a result, this approach thus does not solve former centralisation issues as the nodes keeping the ledger history still act as trusted third parties.

3.4.2 Proposal of a decentralised archiving system

This has led to the design of the SeCuRe Archiving of Transactions for distributed ledger Technology (SCRATT) solution on this thesis work. This contribution has led to a patent [13]. This solution allows a group of users sharing a common goal, like a decentralised application, of a supposedly public DLT to designate one or multiple trusted "*archivist*" nodes of their choice to act as trusted transaction storages, to be then able to prune their distributed ledger using either native [25] snapshotting solutions, or solutions overlayed on top of existing DLTs [78]. This proposition then limits centralisation in the DLT, as users are able to choose one or multiple archivists they already trust (e.g. a regulator for connectivity use cases, or one or multiple nodes with higher storage resources, trusted by the users), rather than having to rely on an imposed external trusted third party shared among every user of the DLT.

The whole lifecycle of the archiving process is automated thanks to a smart-contract implemented on the ledger. Figure 3.14 then illustrates how archiving is achieved (with a single archivist). At step **1**, a group of users selects one of multiple "*archivist*" nodes, and instanti-

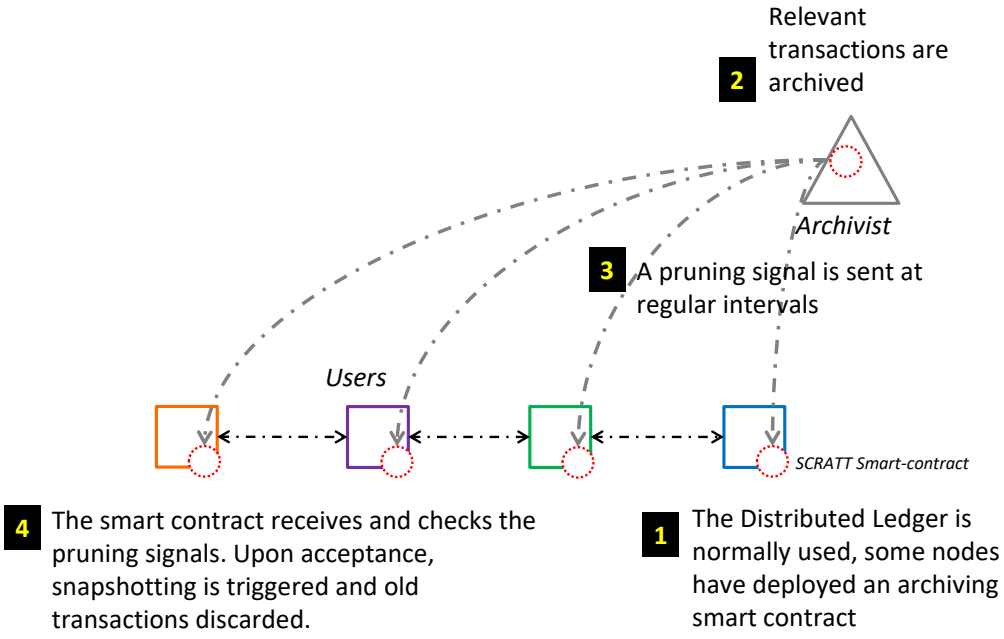


Figure 3.14 – The proposed archiving process

ate the smart-contract by issuing an *"initialisation transaction"*. This transaction contains the following elements:

* One or multiple archivist identifiers;
* And various relevant parameters required by participants' policies (e.g. interval between pruning signals, etc.).

By conception of any DLT, as well as the initial transaction an unique contract identifier (e.g. a hash) is procedurally generated, then saved into the ledger. This identifier can be then transmitted to all of the users involved in the archiving, and is then used to retrieve the data stored about the contract and unique parameters. Archivists then store continuously the distributed ledger transactions (step **2**), and interact with the created smart-contract to issue special signed transactions at regular interval, acting as "*pruning signals*" (step **3**). These pruning signals contain the following informations:

* The associated contract unique identifier;
* The transaction emitter (archivist) identifier, as well as a digital signature authenticating it;
* A sequence number incremented at each pruning signal an archivist sends;
* And any relevant parameters participants may choose to add, regarding their policy (e.g. time of emission of the pruning signal, etc.).

These transactions are then intercepted by the users participating in the smart-contract, using its unique identifier. Users then perform the following checks (step **4**):

* The signature of the transaction must be valid;
* The transaction emitter must be registered as an archivist in the contract;
* The sequence number relative to the transaction emitter must be correctly incremented;
* All parameters initially registered into the smart-contract, related to the participant's policies are respected (e.g. the time difference between two pruning signals matches the time interval set at contract initialisation within a defined tolerance, etc.)

Upon the reception of valid pruning signals from the archivist nodes, users can then snapshot the ledger at the time of the most recent pruning signal, and discard former transactions. Indeed, as per the conception of transactional Distributed Ledgers, a pruning signal transaction undoubtedly validates former transactions of the ledger, that are then kept in the related archivist's memory. If there are multiple archivists in the system, and that users' policies require pruning signals from multiple archivists to be received, users may use the sequence number as a reference. By doing so, upon reception of all the requisite pruning signals, users may account for the oldest pruning signal with the most recent (highest) sequence number for snapshotting. Albeit not illustrated on the figure, the archiving contract can then be stopped at any time by any participant (archivists, or users participating in the contract).

3.4.3 Implementation of the proposed method, and perspectives

The SCRATT method proposes a base framework for multiple partners to safely discard old DLT transactions that remain stored on trusted archivists they trust. The process is further transparent and auditable, thanks to its possible implementation as a smart-contract and related events (instantiation, pruning signals, etc.) registered as transactions. Implementation choices for the proposed solution are then multiple. First, in the case where multiple archivists are designated, partners may choose to only discard old transactions validated by every archivists' pruning signals. In a more trusted environment, partners may only wait for a specific quorum of pruning signals from distinct archivists to discard transactions. The rules about the pruning signals themselves can also be of various nature, depending on the partners' needs and situations. For example, pruning signals could be sent at a regular time interval (10 minutes, 1 hour, ...), predefined at contract establishment. Archivists could also account to the growing size of the ledger, and send pruning signals after the growth of the ledger has reached a specific threshold. For example, pruning signals might be sent after each time 1 TeraByte worth of data is added to the ledger, or after 10,000 blocks/transactions being emitted. Archivists could further send pruning signal on user request, when any user's storage is almost full.

Also, the proposed method can either be implemented "*on-chain*" with all described processes implemented as a smart-contract directly deployed onto the Distributed Ledger itself, so that contract validation can be enforced by all the nodes participating in the Distributed Ledger. This can be the case of smart-contract capable DLTs like Ethereum [79] or Algorand [69]. On DLTs without customised smart-contract capabilities like Bitcoin [8], the proposed method can still be implemented "*off-chain*". In that specific case, the method is overlayed as a layer 2 application on the nodes participating in the archiving contract. These applications then emit transactions with custom data for all events of the archiving process (initialisations, pruning signals, destructions) and validate the emitted transactions according to the process.

Concerning the pruning process itself, the proposed method can take advantage of state-of-the-art existing mechanisms. For DLTs where pruning isn't considered, like Bitcoin, an overlayed solution like CoinPrune can be implemented [78]. Other DLTs have already built-in mechanisms discarding old transactions to free space. In its first version, the Tangle considered in this thesis work has indeed the ability to let nodes discard old transactions, thanks to the trusted "milestones" emitted by the centralised coordinator [25]. This pruning mechanism could be then redirected to account for SCRATT's pruning signals instead.

Yet the proposed method only manages the archiving process itself with the pruning signals of authenticated archivists, and accounts for one or multiple archivists already clearly identified for all participants, and remaining the same through the whole process. Nonetheless participants could implement a more dynamic behaviour, by for example terminating/instantiating new archiving contracts with different archivists at a regular time interval. They may further implement a set of policies and rules to dynamically designate archivists nodes at each turnover, for example using a voting-based or a consensus mechanism-based protocol to select archivists. Such a protocol may also be implemented as a separate smart-contract. Regarding the archiving process itself, partners may also implement policies to penalise any archivist failing to produce pruning signals meeting the constraints eventually pre-established in the archiving contract, or sending faked pruning signals. This is made possible thanks to the easily auditable nature of the Distributed Ledger.

The proposed method is then a simple way to manage a decentralised archiving of any DLT. The method then mitigates the impact of the DLT on the storage space required to operate it, and allows nodes with limited storage capabilities to participate in DLT operations by delegating the storage of old transactions to archivists of their choice. This contribution can futher be implemented on a network-related environment, where DLT nodes deployed onto a telecom infrastructure may have limited storage capabilities. Furthermore the proposed method can serve as a base for a more enhanced archiving system with for example a dynamic selection of archivists, and have thus room for improvement.

3.5 Conclusion

In this chapter, the DLT has been presented more in depth. First, pre-Blockchain and Blockchain-based DLTs have been presented, as well as some of their features and challenges to solve. Then alternate data structures like DAG have also been explored.

In this chapter, two contributions have also been presented:

* First, a simulator of the Tangle, a DAG-based DLT has been presented. This simulator further focuses on the growth of the graph, and placement of new transactions. This simulator helped to validate initial modelling initiatives, and the stability of the technology. It then allowed to further asses how the graph grows when confronted to a high arrival rate of new transactions. The simulator has been made open-source, its code is available at `https://gitlab.com/vmessie/dag-simulator`. It is then used in the "data layer" proposal presented in the next chapter, in Section 4.3 [12].
* And secondly, a decentralised DLT archiving system has been proposed. This system further allows users of a decentralised application deployed on a Distributed Ledger they don't own, to delegate the storage of old transactions to trusted parties, while being able to audit the whole process. Such a solution can be useful in environments where nodes participating in a DLT might have limited storage capabilities, like on a telecom infrastructure. This contribution has led to a patent [13].

The contributions presented in this chapter allowed to better evaluate the relationship between trust mechanisms in telecommunication infrastructures, and the DLTs that may be used to implement such mechanisms in a decentralised way. The next chapter then presents DLT-based architectures allowing the production of trusted, reliable operational data from a multi-actor network infrastructure for its successful management.

Chapter 4

Producing trusted performance reports in collaborative networks with the Distributed Ledger Technology

Contents

This chapter explores the question of securing operational data in a multi-actor telecommunication environment. In multi-actor collaboration scenarii like presented in Chapter 2, having access to trusted, reliable operational data is indeed a requirement for collaboration to be successful. While on this thesis work focus has been made on network-related scenarii, this problematic extends beyond this scope. Indeed, any collaborative systems coping with the outside world may require an access to external reliable data. For that purpose, trusted third parties ("oracles") may be used. The first section presents various initiatives aiming at solving this problem, either in a centralised or a decentralised way. Then contributions of this thesis on the matter are presented. First, a DLT-based system allowing to produce trusted data about the usage of multi-actor network paths is introduced. This contribution considers BALAdIN, a network coverage densification use-case, aiming at inciting non-telco actors to host network capabilities. In a similar way than previous collaboration scenarii presented in Chapter 2, BALAdIN aims at allowing multiple actors not trusting each other to build multi-actor connectivity

services. The developed mechanism thus aims at fostering collaboration on the environment, by providing a trusted, reliable source of informations about the usage of the resources deployed by the participating actors.

Then the main contribution of this thesis is presented, introducing a trusted "*data layer*", allowing the creation of reliable Key Performance Indicators (KPIs) coming from a multi-actor disaggregated network infrastructure. The proposed data layer has been designed taking advantage of the other results of this thesis. Its architecture is inspired on the work on BALAdIN. However, while the contribution on BALAdIN focused on providing usage data, the data layer architecture aims at providing an architecture agnostic to the data itself stored on it. Hence the data layer is able to produce any Key Performance Indicator (KPI) required by the underlying collaboration use-case. The contribution presented in Subsection 2.4.3 is further considered as the driving use-case of the data layer. On this contribution, the use of the DLT is also considered to implement the proposed data layer in a decentralised way.

4.1 On securing network performance data for the operation of multi-actor connectivity services

In Chapter 2, many infrastructure sharing scenarii were discussed. The need for reliable operational data sources has further been identified in the presented use-cases. More particularly, in the contribution presented in Subsection 2.4.2, the presented application relies on an "oracle" reporting reliable data about MNOs' energy consumption. Then in the contribution presented in Subsection 2.4.3, QoS monitoring is necessary for a pro-active management of the collaborative E2E network service chains. In that case the operational data used to calculate the QoS must also be trusted by all involved actors. Many solutions have already been proposed to solve similar problems, on use-cases where secure operational data is needed. These solutions, summarised on Table 4.1, are detailed below.

4.1.1 Centralised initiatives to secure operational data

Centralised trusted third parties, or proprietary hardware are first considered to help securing operational data coming from the outside world. As an example, in the Ammbr project [37], routers are secured thanks to proprietary hardware. They indeed need to solve a "*Proof of Velocity*" algorithm of Ammbr's conception, which, according to Ammbr's claims is only doable on routers' custom hardware [37]. The result of the Proof of Velocity is then a guarantee that a given transaction has originated from a secured Ammbr router. Also, the IDSA architecture, presented in Subsection 2.3.1, is primarily designed to securely exchange data. This architecture yet relies on multiple trusted, certified intermediaries to operate. Among these intermediaries,

Name	Purpose	Description of protocol	Security	Deployment devices	Status
Ammbr [37]	Collaborative mesh connectivity	"Proof of Velocity" bound to specific hardware	Hardware-based, closed	On Ammbr's proprietary harware only	Stalled
IDSA [21]	Architecture for generic data centric securing	Architecture with multiple trusted entities securing data	Trusted components	High-level architecture, non applicable	In development
Torcoin [60]	TOR connectivity relay rewarding	"Proof of Bandwidth" consensus mechanism	Hybrid (assignment servers bring centralisation), consensus-based	On any TOR device	Academic research project
Helium [36]	Decentralised connectivity	"Proof of Coverage" consensus mechanism	Decentralised , consensus-based	On approved hardware	Worldwide deployed

Table 4.1 – Operational data certification mechanisms

the **data clearing house** is notably necessary to keep a record of every event concerning the usage of data happening on the architecture.

4.1.2 On oracles in the Distributed Ledger Technology

The problem of securing "external" performance data extends beyond connectivity-related use-cases, as decentralised applications running on DLTs may require trusted *"oracles"* to properly operate if they need access to data coming from outside the DLT boundaries. As defined in [23], an oracle is a function whose purpose is to provide "external" data to decentralised applications running on DLTs. An oracle can operate in the two directions (from or towards the external environment), as they may not only provide trusted operational data to the Blockchain, but also execute tasks triggered by the DLT in the real world. Such tasks include per example the operation of a connected lock for a rental application, or the onboarding of connectivity resources in collaborative telecommunication use-cases. Tasks being executed in the outside world must be monitored by the DLT, with the help of observations of the environment (e.g. acknowledgements, sensors, etc.). Such observations are thus also provided by an oracle. Yet from DLT side, this function acts as a single point of failure as any flaw in their design may thus compromise the decentralised application relying on them. As a result, oracles must provide a sufficient level of security in relation to their underlying DLT. Many works have thus been undergone, whose some are summarised in [23]. It is worth noting that many oracle solutions then tend to be *decentralised*, involving multiple actors in the process. This thus makes such oracles more

resilient, and increase their reliability. For that purpose, decentralised oracles are often designed in a *consensus-based* way. Among partners participating to the oracle, truthfulness mechanisms, such as consensus mechanisms (PoS, etc.), or voting mechanisms are set up. Also, some decentralised oracle solutions are designed such as truthful playing is a Nash equilibrium, thus incentivising players to act honestly [23]. The next section covers some telecom-related oracle solutions, helping partners involved in collaborative connectivity services to secure operational data coming from the network infrastructure.

4.1.3 Connectivity-related decentralised data certification mechanisms

In collaborative connectivity use-cases, one of the earliest proposal of a mechanism allowing to secure network operational data in a decentralised way is the Torcoin's Proof of Bandwidth (PoB) [60], operated on TOR Onion Router (TOR) network. The TOR protocol [80] is a network protocol designed to allow its users to communicate anonymously. For that purpose, TOR users use a chain of 3 "*relay*" nodes to communicate, that each add a layer of ciphering to the packets ("*onion*" ciphering). However, as discussed in [60], the network grows poorly as all of its participants remain anonymous, making the remuneration of relays impossible. Torcoin is a Blockchain-based proposal, aiming at allowing the fair retribution of TOR relays in an anonymous, decentralised way. For that purpose, a Blockchain is implemented with the PoB, a novel consensus mechanism. A slightly modified version of this mechanism is detailed more in-depth in Section 4.2.1. This mechanism consists in a frame regularly performing round-trips on any TOR path, initiated by the user and hopping through the 3 relays. This protocol is initiated after m packets are exchanged on the TOR path. Cryptographic primitives are then exchanged during the process, requiring every user's approval. Upon completion of the round-trip, the user then saves the complete frame in the Torcoin Blockchain, which in return triggers the remuneration of relays, in a similar way than Bitcoin mining [8]. However, as such this protocol lacks security, for any path with all participants colluding can compromise the protocol and generate faked PoBs as an attempt to generate a greater income. Furthermore, the protocol is initiated by the user that might lack incentives to initiate PoBs, as unlike relays the user isn't rewarded in return. To address the collusion issue, it is proposed to implement trusted "*assignment servers*" to shuffle and enforce the composition of the TOR paths. As per author suggestion, this vastly decreases the probability of collusion, as with 50% of participants colluding, only 6.25% of the then-created TOR paths would be composed of colluding users [60]. Moreover, this might also prevent users to be lazy, as relays may terminate their services and shutdown any TOR path with a user not willing to collaborate. Yet this approach causes evident centralisation issues, as assignment servers then act as trusted third parties. To mitigate centralisation, it is suggested that multiple distinct assignment servers could collaborate in a consortium, whilst the list of created TOR paths would require every assignment server's

approval. Although the Torcoin proposal never overcame the state of academic research project, its proposal is promising as the PoB allows the retrieval of operational data from the TOR network infrastructure in a trusted way. Indeed, one just needs to track down PoBs stored on the Torcoin Blockchain to get an accurate view of TOR paths' usage.

A similar proposal is implemented in the Helium solution developed by the Helium foundation [36], named the "Proof of Coverage". This mechanism consists in multiple radio exchanges among Helium routers providing network coverage. These exchanges are then used to assess the coverage provided by any router, and retribute them accordingly. At a regular interval, a router called the **challenger** "challenges" another router, named the **transmitter**, to prove his location and provided coverage. Upon challenge, the transmitter broadcasts a frame to **witnesses**, that answer by testifying on the existence of the transmitter's frames. The challenger then packs all exchanges performed in a transaction, and adds it to the Blockchain. Upon addition of the transaction, all participants in the Proof of Coverage get rewarded with Helium Network Tokens (HNTs). This protocol then helps the growth of the Helium network, for it incitates its participants to host routers and provide network coverage. Yet unlike Torcoin, the Proof of Coverage isn't used as a consensus mechanism powering the Blockchain, but rather helps providing reliable data about the state of the network. It is worth noting that Helium routers' hardware must be approved by the Helium foundation to operate. As a result, the Helium foundation suggests that the collusion risk in the Proof of Coverage remains low, as approved routers lack enough hardware resources to produce faked Proofs of Coverage [36]. This protocol is now widely adopted, as according to the Helium foundation, there have been more that tens of millions of Proofs of Coverage produced on the Helium network since the beginning [40]. The elements cited above show the growing interest of providing reliable operational data from connectivity infrastructures sustained by the DLT. They further show how a decentralised oracle can be operated on a network infrastructure.

4.2 BAndwidth Ledger AccountIng Network, truthfulness in path usage data in a consensual way

The growing interest on collaborative connectivity, as well as recent work on decentralised connectivity oracles has led us to propose BAndwidth Ledger AccountIng Network (BALAdIN), a collaborative solution powering distributed wireless networks (e.g. 5G, collaborative Wi-Fi, etc.) during this thesis work [15]. This contribution allows the densification of network coverage by incitating a crowd of "local actors", non-telco partners like shop tenants or railway stations, to deploy small cells, usable by any customer of the telcos involved in collaboration. It has been suggested to use the PoB to achieve truthfulness on the collaborative network paths then created between the end user, local actor and used telco's infrastructures, and to provide to the system

reliable information about the usage of the network in a decentralised way. The contribution has further been expanded in this thesis work, to evaluate the implementation of the PoB in the proposed use-case and assess its possible deployment on a Blockchain-based DLT. This work notably showed the limitations of Blockchain-based DLTs to sustain PoBs in such a use-case. Then, a deployment architecture of the layer handling PoBs and providing reliable usage reports is proposed. A sharding approach is notably proposed to mitigate the impact of multiple PoBs coming from simultaneous active paths on multiple ledgers. This extended contribution has led to an article [18].

4.2.1 The BALAdIN solution, in detail

The BALAdIN solution allows a crowd of "local actors" such as shop tenants, railway stations, supermarkets, etc. to deploy small cells managed by their connectivity provider and get a reward based on the use of these cells. Users of such a system could then benefit from better coverage, thus improving their experience for greatly reduced costs.

In the example presented Figure 4.1, three telcos (Green, Yellow and Blue) form a consortium. Thanks to agreements between Green and Yellow operators, Bob, a customer of Yellow operator can get broadband connectivity from Alice, a local actor customer of Green operator. A network path is then established between Bob's device, Alice's small cell, and Green & Yellow respective infrastructures. Similar network paths can simultaneously coexist in the system, involving a different combination of telcos, for example Blue and Yellow. A Blockchain is set up between all actors to track network-related events in a decentralised way thanks to PoB transactions sent at regular interval. From the business side, the design of such a collaborative network has already been discussed in Chapter 2. Promising state-of-the-art projects already introduce similar collaboration scenarii [36, 37, 81], as well as contributions of this thesis work [2, 7]. In this section, focus is thus made on the technical side of the solution. Focus of the work is further made on the Blockchain keeping track on the usage of the network with the help of PoB transactions stored on the Blockchain.

The PoB mechanism is then used to provide a decentralised way of measuring the traffic, with the help of agents deployed on each actor's own infrastructure along the path. While the presence of telecom operators in the access network is no longer ensured for Helium [36] and Ammbr [37], it is still required in the proposed solution. Indeed, in this way, the users and local actors of the platform will then be authenticated by their respective home operators in a way that would preserve their privacy.

A permissionned ("*hybrid*") approach is then considered for the Blockchain holding the PoB transactions, as only participants authenticated by their telco are able to use it.

At step **1**, PoB frames are exchanged within the path, then stored and secured onto the ledger (step **2**), thanks to agents deployed on the network nodes. If the PoB process fails for

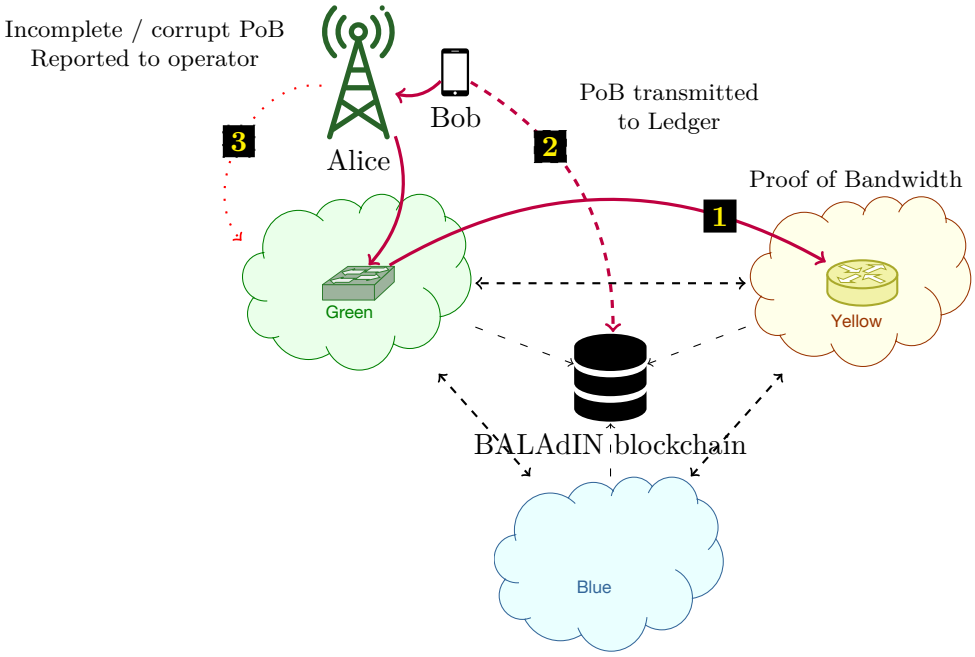


Figure 4.1 – The BALAdIN design

whatever reason this should be reported to telcos (step **3**). Alice, a local actor, has deployed a small cell with the help of Green, her operator. The deployed cell will grant Bob, a traveller customer of Yellow, some mobile connectivity to reach its home operator network.

From the Blockchain side, the authentication system is very simple: each operator generates a Ledger identifier for their customers/local actors and devices. Within the Blockchain, these identifiers are certified with a simple Public Key Infrastructure (PKI) mechanism where each operator would act as a trusted root authority.

In the proposed solution, unlike in Torcoin [60] the PoB is not used as a consensus mechanism. PoB frames are simply stored into the Blockchain using PoB transactions, as a way to keep track of each network path and their use in a decentralised way. The whole PoB process shall allow to set up a bandwidth allocation and billing system similar to PayFlow [81].

Also, there are no assignment servers as the paths are self-regulated. Indeed, unlike Torcoin, the path nodes are physical devices (smartphones, base stations, etc.). Therefore they cannot be shuffled. However:

* Operators want their resources to be used and optimised; therefore on such a multi-actor collaborative platform they should provide a way to measure traffic in a decentralised way. In the case of the Visited operator (Green), this system will ensure a fair measure of the traffic used. The home operator (Yellow) will then be able to use this decentralised mechanism to testify of its usage of the collaborative path. This protocol shall then strengthen inter-telco agreements thanks to the induced trust.

* The Local Actor (Alice) needs incentives to deploy cells; therefore a fair remuneration based on the use of them shall be set up for her. A trusted traffic measurement mechanism will also give her the necessary trust.
* And finally, the customer (Bob) is also part of the PoB process. While it is still possible for him to block the frame and cheat on the measure, the path could in return be killed by the local actor or the operator.

In the proposed case, the collusion risk is evaluated as pretty low, for it is not in the interest of the involved actors. While these assumptions are not sufficient in the case of a fully public Blockchain, in the proposed case the use of a permissioned Blockchain may prevent any risks. Indeed, the telcos act as trusted third parties, and certify the identity of users and local actors. While fraud detection mechanisms themselves are outside the scope of this work, the telcos may use such mechanisms to blacklist cheating and colluding users. Let us now see in details how the proposed solution works.

Path Creation

The path creation mechanism is described in Figure 4.2. Initially, Bob and Alice's identifiers are both registered on the ledger by their respective operator's PKIs. Bob will at first physically attach itself to Alice's small cell, and then initiates an *initial PoB* process by sending the first frame, serving as an attachment request (step **1**). The frame is first fed by this request and will be forwarded to Green network, then to Yellow network. Given that:

* Alice and Bob are both registered by their operators on the Blockchain;
* cooperation is possible from Green to Yellow operator, according to their respective policies;
* and each operator agrees on the creation of such a path;

the PoB frame will then come back, performing the return trip from Yellow network to Bob. Bob will then generate a *path creation request* containing the full PoB that will be submitted as a transaction on the Blockchain (step **2**). This transaction then triggers the activation of the path, via a smart-contract deployed on the Blockchain. This is step **3**: the transaction has been processed, and as a result, every implied actor receives a "positive feedback" from the Blockchain. Upon reception of this feedback, Yellow and Green may trigger the opening of traffic flow.

The path creation transaction will then be validated and its identifier stored on the Blockchain. The transaction should then be accepted on the Blockchain only if Bob and Alice's agents have been registered, the identifiers used by Yellow network's and Green network's agents are valid, and the initial PoB cryptographic variables are correct. Once the path is considered as active, Bob may begin to use it.

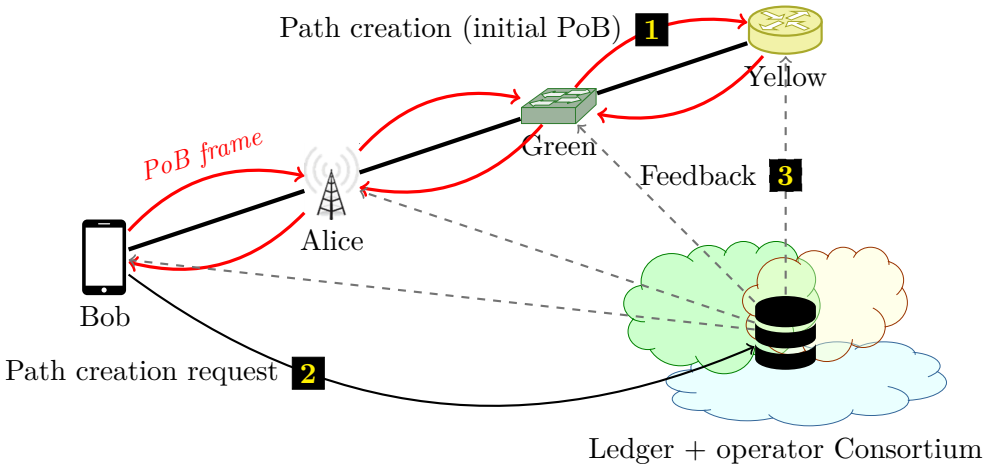


Figure 4.2 – BALAdIN path creation

Path flow

Once the path is created and active, all path users count the amount of traffic exchanged on their own interfaces. At a specific rate, a PoB frame will be initiated and sent by the client (Bob), containing information about the traffic exchanged. This frame will then perform a round trip on the path. During this round trip, the other path users will thus compare their measurements to Bob's, and validate it using the process described in Section 4.2.1.

In the proposed implementation, choice have been made to emit PoB frames at a fixed time rate instead of a fixed data rate. This assumption is motivated as to allow the measurement of any network path at a constant rate, regardless of their activity. Then a simple study has been conducted to estimate at which rate PoB frames should be emitted on a network path. First, at the time of the study, it has been assumed that mobile network peak rates should be around the Gigabit [82]. Furthermore, taking as an example Google Fi [83], one can observe that telcos usually bill each GigaByte of data for "on-usage" plans.

As a result, assumption has been made that a precision of a GigaByte should be efficient enough for traffic measurement, for in case of a failed PoB process the financial loss would be minimised. Therefore, the time between two PoB frames should be of $8s$, as the resulting precision will be of at least 1 GigaByte.

The packet overhead is then given by the following equation, plotted in Figure 4.3:

$$O_h = \frac{L_{PoB}}{\delta t B_{ps}}, \tag{4.1}$$

where O_h represents the overhead, B_{ps} the effective path bitrate, δt the time between two PoB frames (8s) and L_{PoB} the size of a PoB ($3.128kB$ in the case described in Section 4.2.1, using regular ciphering suites).

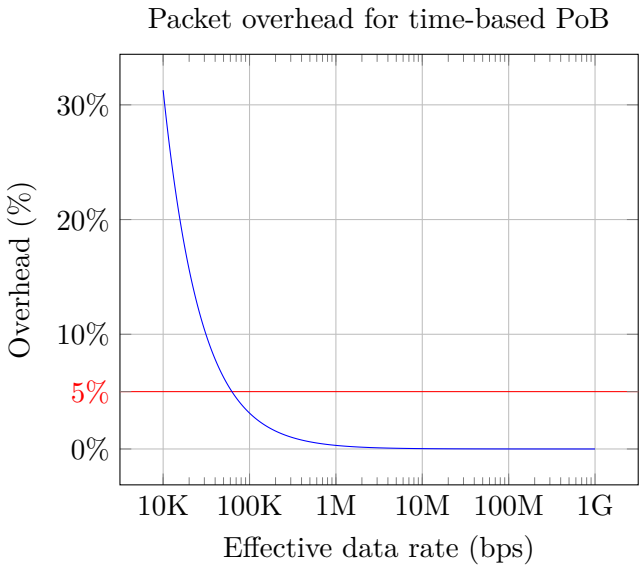


Figure 4.3 – BALAdIN PoB packet overhead

Although the lesser the data rate, the greater the packet overhead, this choice should not be an issue, as for a maximum data rate of 100 Kbps the overhead is less than 5% (Figure 4.3). As nowaday's network peak rates are way above 100 Kbps, the overhead introduced by PoB frames should not be an issue.

The method used by agents involved in the PoB process to count the traffic is still a question to address. Indeed, multiple factors can be considered, as to discriminate uplink and downlink traffic, as to consider also at which protocol layer counting should occur. Furthermore, due to the conception of LTE networks and the multiple protocol layers modifying the frames along the path, the measurements may diverge between the different actors of the BALAdIN path.

As a result, PoB agents should use a set of simple rules such as setting up maximum thresholds of divergence of measures, etc. Such rules should be carefully established beforehand depending on the path user's needs, so that their measures can converge and the PoB process be successful, while still providing a sufficient level of security for fulfilling path user's accuracy needs.

Then, similarly to the path creation (Figure 4.2), a PoB transaction will be sent to the ledger, and path users will monitor its successful commitment.

Path destruction

Any user of the path that has detected a fault or an abuse in the operation of the path should shut the path down and send the incomplete frame to its telcos for investigation.

Path users should also implement a "timeout" mechanism, to terminate the path if PoBs are

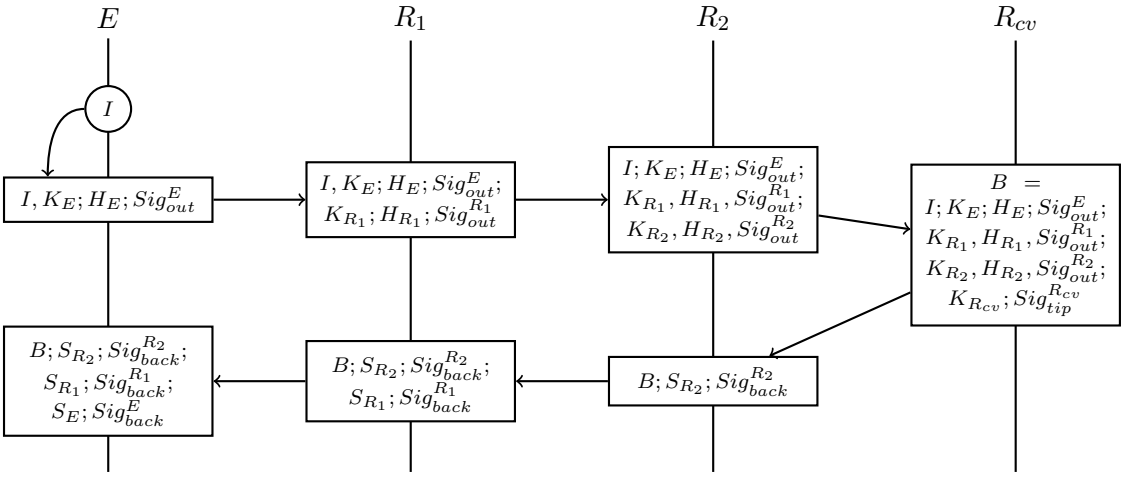


Figure 4.4 – The PoB process

not validated and committed onto the ledger within the limits, or if a PoB transaction is found to be invalid.

The network path is at first directly terminated, then the event should be logged onto the Blockchain using a Path Destruction transaction. This transaction may be sent at any time, by any path actor (user, local actor, or any operator).

A closer look to the proposed Proof of Bandwidth Implementation

The process is described on Figure 4.4. In the following section, Bob is considered as the **Emitter** (E), as it starts the process, Yellow Router the **Receiver** (R_{cv}) at it is the target that Bob wants to reach; and Alice and Green agents are considered as **Relays** (R_1 and R_2), as they relay traffic and **provide network resources** so that Bob and Yellow may communicate.

At first, the Emitter creates the PoB initial frame I containing:

* The PoB number $P\#$. For the initial PoB, $P\#$ is set to zero.
* The timestamp in POSIX64 format t,
* the Path identifier P_{id}. This identifier is calculated thanks to a hash function performed on the path users' identifiers.
* The padding (information for path users about the next PoB time), Π (default to 8s)
* The amount of data exchanged since the last PoB frame, Λ.
* Hence, Let $I = \{P\#; t; P_{id}; \Pi; \Lambda\}$.

This frame will then hop through the network path, performing a round trip. On each step (relay 1, ...), the other path users (relays, receiver) will, upon approval of I, perform cryptographic calculations as defined in [60] and digitally sign the frame. These calculations

imply the generation of a random *nonce* value S_x prior to the PoB process as well as its hash H_x, the addition of the hashes H_x at each hop toward the receiver, and of the actual values S_x on the return trip. As suggested in [60], this further secures the PoB frame, as the nonces are only revealed during the return trip, hence preventing any modification of initial informations. These signatures and variables will then carry a "proof of approval" of the PoB frame, and shall then guarantee the integrity of the data contained into I.

On the contrary, each actor within the path should ignore any PoB frame that they find corrupt, and consider to end the path, following the process described in Section 4.2.1.

Compared to the state of the art PoB process [60], the implementation of the protocol in the proposed use-case is slightly modified. Indeed some additions have then been made on the PoB:

* The frame is signed in both ways by users;
* The initial tuple (I) is improved, by the addition of extra fields:
  * The amount of data exchanged on the path since the last PoB frame;
  * The timestamp corresponding to the creation of the initial tuple;
  * A field indicating after which amount of data exchanged on the path the next PoB frame should occur;
  * And a path identifier (set to zero for the first one, as it is not known).

This choice is motivated by the different nature of the usage of the protocol. Indeed, whereas Torcoin uses the PoB as a consensus mechanism, the mechanism is rather used as a proof of the usage of a path in BALAdIN. As such:

* The double signature will allow everyone on a path to report abuse to its operator, and the incomplete PoB will help to identify where the failure has occurred;
* Logging the data amount and the timestamp of the initiation of the PoB will allow more precise metrics as a given path throughput could then be measured.

The complexity added to the frame shouldn't be an issue, neither for resources consumption or byte overhead. Indeed, current cryptographic algorithms such as Rivest–Shamir–Adleman (RSA) for digital signatures/Secure Hash Algorithm (SHA) for hashes remain relatively simple compared to nowadays's devices capabilities.

At the time of the study, to confirm this assumption, benchmarks of the SHA256 and RSA2048 performances have been performed on "basic" virtual machines (2GB RAM / 2 virtual CPUs), using the `openssl` tool. Results are displayed on Figure 4.5.

Results indicate that the cryptographic operations performed to create a PoB can be realistically made on every kind of devices without consuming too much resources, as nowadays even low-end smartphones have now at least the tested configuration (2GB ram, dual-core CPU).

```
### SHA256 ###
Doing sha256 for 3s on 16 size blocks: 5130762 sha256's in 2.99s
Doing sha256 for 3s on 64 size blocks: 2770388 sha256's in 3.00s
Doing sha256 for 3s on 256 size blocks: 1220497 sha256's in 2.99s
Doing sha256 for 3s on 1024 size blocks: 362161 sha256's in 3.00s
Doing sha256 for 3s on 8192 size blocks: 47681 sha256's in 2.99s

### RSA 2048 ###
                  sign    verify    sign/s verify/s
rsa 2048 bits 0.002936s 0.000088s    340.6  11400.8
```

Figure 4.5 – RSA 2048 and SHA256 performances

Transaction validation at Blockchain side

For any received PoB transaction, it should be carefully checked on the Blockchain that:

* The path users are registered on the Blockchain and active;
* The path is active (has been properly enabled with initial PoB, and has not been destroyed by any user), except in the case of the initial PoB;
* All cryptographic variables (hashes, signatures) are valid;
* The data contained in I is realistic (the $P\#$ is incremented properly).

As the actual traffic exchanged on the path is opaque from Blockchain side, the integrity of this metric should then be validated by the path users themselves, which implies methods to ensure reconciliation. Yet the result of the process should be trustworthy enough, for a PoB frame requires everyone's approval in a consensual way.

4.2.2 BALAdIN performance evaluation

In this section, the performances of BALAdIN and its underlying Blockchain are explored. The maximum transaction throughput (TPS) is the main parameter considered is this study, as it has a direct influence on the capability of the Blockchain to support many simultaneous BALAdIN paths all constantly emitting PoB transactions. Simulations of the BALAdIN Blockchain are then performed on a virtualised testbed. The observed transaction processing time is then extracted as a metric indicating how well does the Blockchain process PoB transactions. This section thus shows the limitation of Blockchain-based DLTs for supporting the proposed use-case.

Limitations of Blockchain for the proposed use-case

As explored in Section 3.2, state-of-the-art Blockchains present limitations. Blockchains indeed do not scale well, and have a limited transaction throughput (TPS). In the proposed

use-case, the required transaction throughput can be evaluated as such:

$$T_{PS} = x/\Delta_t \tag{4.2}$$

with x the number of simultaneously active paths, and Δ_t the time between two PoB frames, as with each PoB frame is associated a PoB transaction. Mutually, if the maximum TPS of a Blockchain is known, one can evaluate the maximum number of simultaneous active path:

$$x_{max} = T_{PS_{max}} \times \Delta_t \tag{4.3}$$

As discussed in Chapter 3, the maximum throughput of a Blockchain $T_{PS_{max}}$ varies from implementation to implementation. At the time when BALAdIN was developed, the most scalable Blockchain project open to experimentation was Hyperledger's Sawtooth project [84]. This solution indeed implements a novel consensus mechanism called the Proof of Elapsed Time (PoET) [85]. The PoET consists in a cryptographic challenge to solve in a limited timeframe, taking advantage of hardware-provided features like the Intel Software Guard eXtensions (SGX) environment. This implementation allows a theoretical maximum TPS of 1000. On a Sawtooth implementation of the BALAdIN PoB, x_{max} is then evaluated as $1000 \times 8 = 8000$ simultaneously active paths at maximum.

Assessing the total simultaneous amount of users is however difficult, for this metric is dependant on the scenario considered. It further depends on the level of coverage (regional? national? worldwide?) of a BALAdIN network. To approximate this number, a study has then been performed in this work taking statistics of public Wi-Fi hotspots coverage. Public Wi-Fi is indeed considered to be close to the proposed use-case. Some projects like World WiFi [86] have estimated that there are more than 4.5 million hotspots worldwide. Therefore, given that World WiFi estimates simultaneously 2-3 users per hotspot on average, the excepted global transaction rate would be of $1,125,000$ - $1,687,500$ transactions per second. This number is well beyond the previously assessed max transaction throughput of a Blockchain like Sawtooth. This thus makes a worldwide implementation of BALAdIN unrealistic. On the other hand, a regional deployment of BALAdIN might be feasible. At the time of the study, the O2 service had $15,000$ hotspots through UK [16]. The expected number of simultaneously active connections through the entire UK can be estimated to $45,000$. The UK could then be split into 6 separate regions with their own BALAdIN networks and Blockchains.

Beside these throughput considerations, issues might also arise on the commitment of PoB transactions. Indeed, as discussed in Section 3.1.2, block/transaction commitment is never guaranteed, due to the decentralised P2P nature of a Blockchain. As the persistence of a block – and its transactions – is only probabilistic, users of BALAdIN may need to set up their own rules regarding block validation. Nonetheless, like state-of-the-art applications of the DLT, a

block can be considered as committed only after his depth has reached a threshold value to be chosen [55].

Evaluating Proof of Bandwidth integration onto BALAdIN

To further evaluate the proposed architecture, a simulation testbed of the peer-to-peer network hosting the Blockchain supporting and handling the PoB has been made. The purpose of this testbed is to assess the Blockchain-side performances regarding the maximum transaction throughput, and also the propagation and commitment of the transactions within a simulated environment. Results then allowed to validate former assumptions on the behaviour of the Blockchain, as well as fine-tuning the proposed architecture to propose an initial deployment scheme. More particularly, experimentations allowed to determinate a "timeout" value for PoB transaction commitment, as this value depends on the observed transaction propagation delay.

At the time of the study, the Hyperledger Sawtooth project was the most basic Blockchain engine [84].

The Sawtooth API indeed provides only the necessary abstraction to allow application deployed on it to interact with the Blockchain like with a regular transactional database.

For that purpose, applications developers can deploy "*transaction processors*" applications on the nodes running Sawtooth. These applications can then handle and decode customised transactions, choose to validate them or not, and update a "*state database*", an abstract representation of the application data that can be modified at will. As a result, all Blockchain-specific operations (peering, consensus mechanism, block structure/scheduling, etc.) are managed by Sawtooth, whereas the actual data stored in the ledger are fully customisable, and application specific. Furthermore, as said above the Sawtooth project uses an efficient PoET consensus mechanism [85]. Sawtooth has then been chosen to implement the proposed use-case, as the technology is the most customisable whilst promising good performances.

As depicted on Figure 4.6, a transaction processor has been created to handle PoB transactions. For testing purpose, the transaction processor is designed to handle not only *path creation*, *path destruction* and regular PoB transactions, but also user ("client" and "relay") registrations in the Blockchain, a process that would be normally achieved by the operators. All the data necessary to handle BALAdIN transactions (registered user keys, active paths and their parameters) are thus stored into the state database. Using it then allows the transaction processor to fully check for the integrity of the PoBs like described in Section 4.2.1. While the core of the validator, the consensus mechanism and the REpresentational State Transfer (REST) API are provided by Sawtooth, the Transaction Processor and Tx Submitter Agent deployed on the customer's UE are designed for BALAdIN. The validator schedules the new blocks; manages the State Database and manages the interconnection with other nodes. Then, with Sawtooth, the consensus mechanism is actually implemented in a separate component to easily replace

it. The REST API acts as a proxy for incoming transactions, by providing convenient ways of submitting transactions and monitoring the chain via a WebSocket server. In the proposed case, the local actor, visited operator and home operator subscribe to the WebSocket channel to monitor the commitment of the PoBs.

Some programs have also been created to send mock PoBs transactions to the validators. These programs simulate every user of a mock BALAdIN path, in order to test the transaction processor and the Blockchain as a whole. Mock transaction generators work as follows:

* As the first stage, for each simulated simultaneous path, they generate the necessary keypairs (for the two clients, and the two relays).
* Then, using previously generated keypairs, they submit the initial PoB transactions described in Section 4.2.1 to register the paths.
* To then, at a specific rate, submit the PoB transactions for each simulated path.

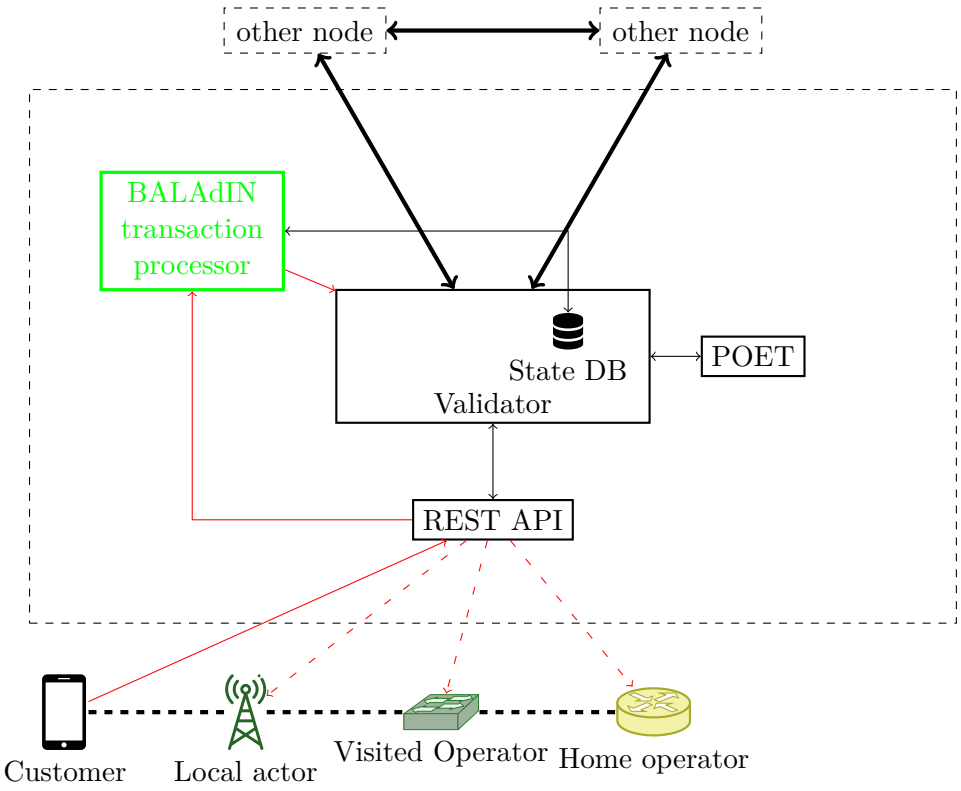


Figure 4.6 – Overview of a BALAdIN node implementation using Sawtooth

Two distinct networks of nodes are then created, separated by a router, as depicted on Figure 4.7. The goal of this router is to emulate realistic network propagation delays between two distinct areas, by adding latency with the help of the `tc` Program [1].

1. https://linux.die.net/man/8/tc

Then, for each subnet:

* The three nodes are monitored:
* On each side, two nodes, highlighted in dashed green are "*active*", as they receive transactions to commit from the test programs, and the third node (in solid grey) is only "*passive*" (does not receive any submitted transactions).

The presented testbed is then deployed on a single-node Openstack suite (Devstack). This choice indeed allowed to get the simplest installation, with the use of basic and low-level tools such as openvswitch for networking, qemu and KVM for virtualisation, etc.

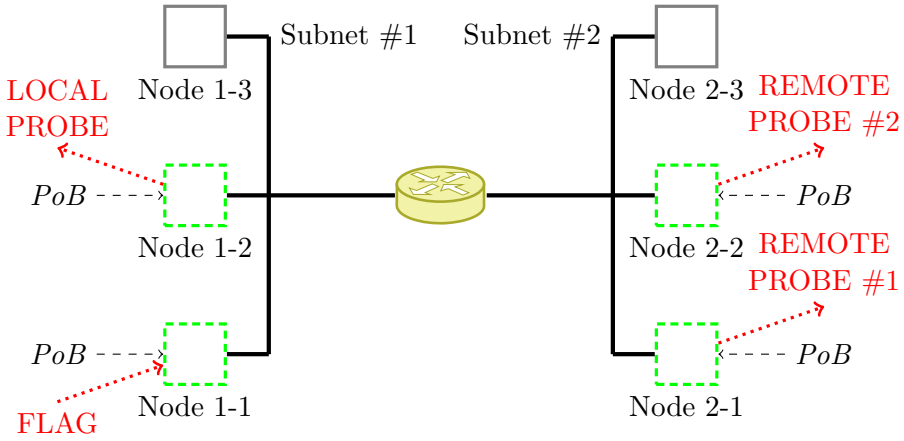


Figure 4.7 – BALAdIN network testbed

To assess the performance of the Blockchain, mock PoB transactions are generated and submitted to nodes, at a specific rate. The central router then simulates a network latency, following a random normal distribution centered on 50ms. This value has been arbitrarily chosen as being the mean observed latency in mobile networks at the time when this work was performed.

Sawtooth nodes were then deployed on virtual machines with a dual-core CPU (2 vCPUs) and with 2GB of RAM, which was roughly the hardware of a nano-computer at the time of experiments.

Then, the propagation delay of PoB transactions was also assessed by using "*probes*":

* At first, the nodes are synchronised on the same clock using an external NTP server;
* One active node of the testbed generates and submit PoB transactions with a special "*flag*", and log the current time of submission;
* And *probes* are set up on the three other active nodes. These pieces of software watch the Blockchain's incoming transactions, identify the *flagged* ones, and log the time of reception.

* The measured transaction propagation delay here is then, for each probe, determined as the difference between the time of submission, and the time of reception of a "*flagged*" PoB transaction. This transaction propagation delay is then extracted for each probe.

Figure 4.8 shows the results of the conducted experimentation. At first (from $t = 0$ to $t = 14$ minutes), the experiment is run with an idle Blockchain (no other PoB submitted). Then, from $t = 14$ to $t = 32$ minutes, each active node sends 10 PoB transactions per second to the Blockchain. Finally, from $t = 32$ minutes, the Blockchain is idle again.

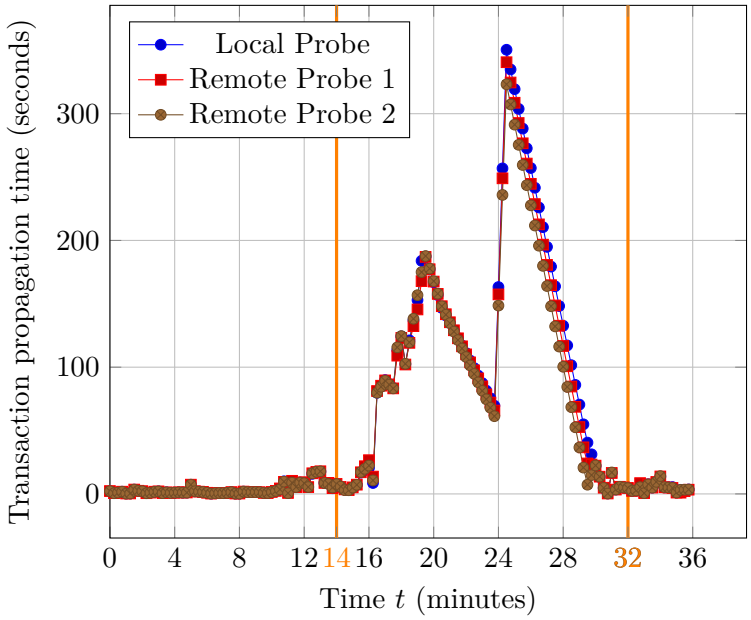


Figure 4.8 – Measurement of the PoB transactions propagation delay on a network testbed

Following observations can be made:

* When the Blockchain is idle, the transaction propagation delay remains low (bounded to 2-3 seconds), and doesn't fluctuate. This seems to indicate that the processing of PoB transactions is nominal as they propagate within finite time on the network.
* However, when every active node emits 10 PoB frame per second, the propagation delay dramatically increases to up to 5 minutes ($300s$). It also seems to fluctuate at random. This behaviour indicates that the processing of PoB transactions isn't consistent anymore while under load.
* There isn't any significant differences between the three probes.

The big increase of the transaction propagation delay during the "active" phase tends to indicate that the Blockchain reached congestion, as a huge increase in the transaction propagation delay is observed, and the latter fluctuates without any recognisable pattern. On the

other hand, the network infrastructure does not seem to have any significant impact on the transaction propagation delay, as little to no divergence was observed between the three probes, regardless of their location in the testbed. These results yet show that the global transaction throughput of BALAdIN has a strong impact on the transaction propagation delay, and then the processing time of PoB transactions. It is also noticeable that this impact can be observed at a relatively low throughput (40 TPS, for a theoretical maximum of 1000 TPS). Yet this observation can be the result of the testbed itself, as nodes are deployed as VMs on a single physical host. This setup does not reflect what a real deployment would look like, that might enable a higher throughput. However, it has shown that the proposed technology might be inappropriate to sustain BALAdIN's PoB transactions, as from application side the Blockchain gives erratic results while reaching its congestion point. For a real deployment of BALAdIN, special care will need to be taken as to limit the effective TPS, or the amount of simultaneously active BALAdIN paths.

4.2.3 Discussions on the implementation of BALAdIN

As depicted in Section 4.2.2, a globalised, general deployment of the Blockchain handling PoBs is not something doable, due to the relatively high transaction rate required by the use-case. An alternate approach can however be considered, by deploying multiple "*shard*" Blockchains dedicated to PoB handling. These shards could then be interconnected in a same way as explored in Ethereum 2.0, whose architecture is presented in Section 3.2.1 [62]. This approach is feasible as two BALAdIN paths managed on different shards should not interfere, thus preventing any conflicts between shards. However, some mechanisms need to be set up to prevent any user to simultaneously use paths from two separate shards, for this may lead to double spends afterwards. Indeed, if any user is simultaneously active on two separate shards, he might be remunerated/charged twice with the PoB transactions on both sides.

Possible deployment

For a better reactivity, handling PoBs right at the edge of the network, directly on the local actors' small cells could be a good idea. However, such devices might not have the required hardware to process transactions, for the observed transaction propagation delay increases dramatically when confronted to a high global transaction throughput when the Blockchain is deployed on VMs with low allocated resources.

Some pre-processing of the PoBs could still be achieved at the edge, as cells managing multiple paths simultaneously could pack multiple PoBs onto a single batch for faster processing for the nodes running the Blockchain. The integrity of the PoBs may also be checked at this step, as from application side this step does not consume too many resources. The Blockchain nodes could then be deployed by operators higher in the network, at a regional level.

Such a hybrid approach is not fully decentralised. However, it will allow a better user experience for customers and local actors, while preserving some levels of transparency for them. Furthermore, a reduced number of nodes will reduce the probability of a desynchronisation and then divergence of the Blockchain. It should also be noted that the processing of PoBs remains transparent, and that users and local actors of the platform would still be able to deploy validator nodes themselves, by providing the necessary hardware. In that specific case, the latency induced by the transaction submission itself to a node in the region should not be a problem as it will remain low compared to the time needed to process a transaction.

Bearing this deployment in mind, whereas the users remain able to wait for a given number of confirmations depending on their own policy, a transaction may be considered as committed with zero confirmations. Indeed, the separation between paths and the limited number of nodes, as well as their identity certification should already provide a sufficient level of truthfulness. However, after m failed submissions of PoB transactions, users of the path should consider to destroy the path following the process described in Section 4.2.1. The value of m remains an open topic, for this should be determined by the underlying contract and the valuable assets implied. All submitted PoBs should, however, be cached by the emitter after submission. Indeed, as within a path, PoBs are sequential (each PoB transaction is dependant of the previous), they will need to be resubmitted if the frame is dropped anywhere in the network during the process. In that case, the failure can be detected by monitoring the Blockchain and the commitment of the PoB transactions. This study then allows to consider a deployment as illustrated on Figure 4.9. The platform as a whole then has the following elements:

* First, an "*overlay*" function shared only between telcos is deployed. It would be responsible to manage the user authentication, and remuneration model. The structure this system would take remains an open topic as on this specific work focus has been made on PoB integration. Yet a DLT could be considered to implement this function. This has indeed already been considered in state-of-the-art work [36, 37], as well as been considered in this thesis work, on the contribution presented in Subsection 2.4.3 [7].

* Then, multiple "*shard*" Blockchains are deployed. Shards store PoB to keep track on the access network and users interactions. They interact with the "*overlay*" function in two ways:

  * The overlay manages the authentication of every BALAdIN user with a PKI-like mechanism. Shards then interrogate the overlay to validate transactions coming only from registered users.

  * Then; a remuneration engine deployed on the *overlay* function would read the committed PoBs as a proof of the usage of given network paths; so that a fair retribution/billing could then be ensured for every user.

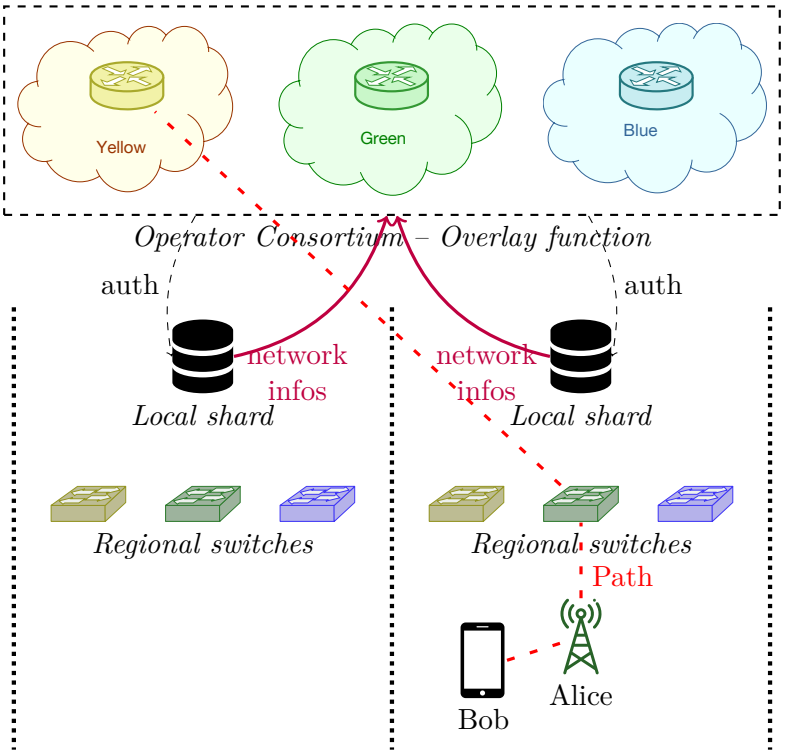


Figure 4.9 – A proposition of deployment of BALAdIN

Open issues

In this section, BALAdIN, a Blockchain-based solution enabling collaborative coverage has been presented. Focus has notably been made on the implementation of the PoB to monitor the usage of the network in a decentralised way, hence enhancing truthfulness among partners. The behaviour of the Blockchain processing PoBs has been particularly assessed, which allowed to estimate some of the key requirements of BALAdIN and its PoB integration. Yet the proposed testbed suffered limitations that may be considered for future experimentation:

* At first, it is hard to efficiently simulate a Blockchain network using only virtual resources. Indeed, physical hosts still have limitations to successfully emulate a big number of virtual machines, such as disk/net IO.

* Furthermore, for the Sawtooth case, the PoET mechanism should use the Intel's SGX hardware feature [85]. As it requires close interaction with the CPU hardware, it is impossible to exploit such a feature on a Virtual Machine. On this work's experiments, the "PoET simulator" consensus model provided with Sawtooth has rather been used. The resulting computing requirements of this component may also throttle down the overall performances.

Also, many components/functionalities of the proposed solution have not been covered yet:

* The authentication of BALAdIN users needs to be carefully designed, for such a system would need to provide safe authentication and identity management, whilst preserving the privacy of each telco's customer.

* The overlay function, as well as the business model of this proposal also needs to be designed, in a way to trade connectivity on the most efficient way. Yet such business models have already be proposed on previous works. Indeed, many DLT-based collaborative systems have already been imagined, as a way to sustain collaboration between multiple connectivity-related actors [36, 37]. Such collaborative systems have also been studied as part of this thesis work, in Chapter 2 [2, 7]. In all the covered cases, the purpose of such systems is to foster collaboration between telcos and eventually non-telco partners to build better network infrastructures, while being able to automate the life-cycle of connectivity services. A solution like BALAdIN can then interact with such collaborative systems, by providing trusted informations about the usage of the network.

* The remuneration model, as well as the possible penalty mechanisms can be more detailed as well. Whereas this question was outside the scope of this thesis, the definition of the retribution/penalty mechanisms may take advantage of game theory, in a similar way than the contribution presented in Subsection 2.4.2. The remuneration model will need to establish a Nash equilibrium so that each actor has an individual interest in acting honestly during the PoB process.

* The necessary safety mechanisms to provide the necessary QoS/QoE for end users also need to be addressed. Indeed, as no telco holds the infrastructure as a whole, this topic should be particularly taken care of. This problematic is somewhat shared with the contributions presented in Section 2.4, for in these contributions telcos also share their infrastructures to achieve collaboration. More particularly, in Subsection 2.4.3 a "*data layer*" is made necessary to provide reliable and trusted KPIs to provide service assurance. It should however be noted that in BALAdIN case, the PoB already provides some trusted information about the usage of a shared network path. Yet this information remains limited by the nature of the PoB frame, as the initial frame containing the information to secure is produced only by the customer. The PoB process could then be improved to account for extra parameters like latency/jitter, etc. so that full KPIs could be calculated.

* The actual integration of the PoB frames into the network needs to be discussed as well. Furthermore, due to the wide variety of protocols used into modern mobile networks, the question of which counting/reconciliation mechanisms to use in the PoB process need to be considered as well.

This work also indicated that a Blockchain approach to store PoBs, or more generally performance reports produced at a regular time interval isn't optimal. Limitations of Blockchain-based

DLTs might indeed induce a bottleneck for processing such transactions, for the presented experimentation showed a huge increase of transaction propagation delay while the Blockchain was under load. Furthermore, the amount of simultaneous BALAdIN path can be expected to fluctuate, for they can be instantiated and terminated on the fly. This then results in a fluctuation of the PoB transaction throughput, which cannot be accommodated by a fixed number of shards.

In the next section, a similar contribution is presented, allowing partners sharing their infrastructure like described in Subsection 2.4.3 to implement a "*data layer*" to produce and exchange trusted performance reports. Compared to BALAdIN's PoB, the trusted information produced by the data layer isn't bound to network paths' usage, as the proposed architecture is agnostic to the data secured on it. It can then support any KPI needed by its users, either customised or standardised [87, 88]. Yet in the following section, a DAG-based DLT implementation is considered rather than a Blockchain to process performance reports, as to enable better performances and efficiency while avoiding complex sharding like proposed with BALAdIN.

4.3 A generic data layer for end-to-end agnostic service assurance

This section presents a contribution of this thesis work, on the design of a decentralised "*data layer*" providing service assurance to telcos sharing their infrastructures to build multi-actor End-to-End (E2E) connectivity service chains. To achieve service assurance, the data layer allows the production of trusted, reliable Key Performance Indicators (KPIs) produced with performance data coming from the infrastructures of every telco involved in the service chains. These trusted KPIs can then be used as a trusted input in the network's management system. As an example, on implementing the marketplace with the DLT like presented on Subsection 2.4.3, from marketplace side the data layer has then the role of a trusted oracle providing operational data, or as a "layer 2" system built on top of it.

It is notably proposed to implement the data layer architectures with the help of a dedicated Distributed Ledger to build trust and avoid centralisation. The use of DAG-based DLTs is further considered to achieve efficiency. This section then presents the proposed data layer architecture as well as its key components, then motivates the choice of a DAG-based DLT to sustain it, taking the Tangle [11] as a driving example. This contribution has led to a conference paper [12].

4.3.1 On Cloud-RAN sharing

In Subsection 2.4.3, a federated Communication Service Provider (CSP) marketplace scenario has been presented, allowing any CSP to exchange connectivity resources. This scenario takes

advantage on both the emergence of Network Function Virtualisation (NFV) making network infrastructures elastic and easily shareable between multiple users, and multiplexing techniques such as Wavelength Division Multiplexing (WDM) allowing to easily share network links between multiple users. In this work, CSPs providing infrastructure are referred to as "*providers*", users deploying virtual resource on providers' premises are referred to as "*consumers*". Hence, these two categories of actors are independently referred to as "*prosumers*" in this work. In this section, Centralised-RAN (C-RAN) architectures are further considered. On future Radio Access Networks (RANs), the radio signal base band processing Virtualised Network Functions (VNFs) can indeed be deployed higher in the network. Such functions are spread in Radio Units (RUs), Distributed Units (DUs), and Central Units (CUs). RUs are placed on the antennas themselves (or close to); DUs can be placed only a few kilometers from RUs in MEC infrastructures while CUs can be deployed higher in the network in central cloud infrastructures [49, 89]. Each of these elements are interconnected by dedicated optical networks, namely the "*fronthaul*" between antennas (RUs) and MEC (DUs); and the "*backhaul*" between MEC and core network (where CUs might be deployed).

Then, as illustrated on Figure 4.10, a given MNO aiming to deploy a mobile network can use (i) a third party towerco infrastructure to place RUs, (ii) a MEC provider close to the chosen antennas to host DUs, and (iii) a third-party central cloud infrastructure to deploy both CUs and core network functions. Another MNO could further use the same infrastructures to deploy his own VNFs. The marketplace facilitates relationships and exchanges between prosumers. When a consumer requests an asset which matches offers from providers, SLAs are settled and resources are deployed. A SLA describes the obligations, constraints and commercial arrangements (between a provider and a consumer) attached to a service. Each provider needs then to manage his infrastructure according to the negotiated SLA.

Indeed, the VNF distribution between the RU, the DU and the CU may have a high impact on the QoS as network delays are increased. The authors in [89, 90] address the resource sharing principles and scheduling requirements for running RAN functions on cloud environments while meeting the RAN real-time requirements. For the purpose of E2E assurance across prosumers, a trusted "*data layer*" is required to allow players to share necessary information required to guarantee E2E services [88]. Prosumers can then check and audit the delivered services according to SLAs, without relying on any third party.

SLAs can then further identify the required performance metrics and how they need to be processed to validate the operation of the agreed E2E services. This validation can then be used as an input in the marketplace to manage multi-party settlement transactions and reconciliation.

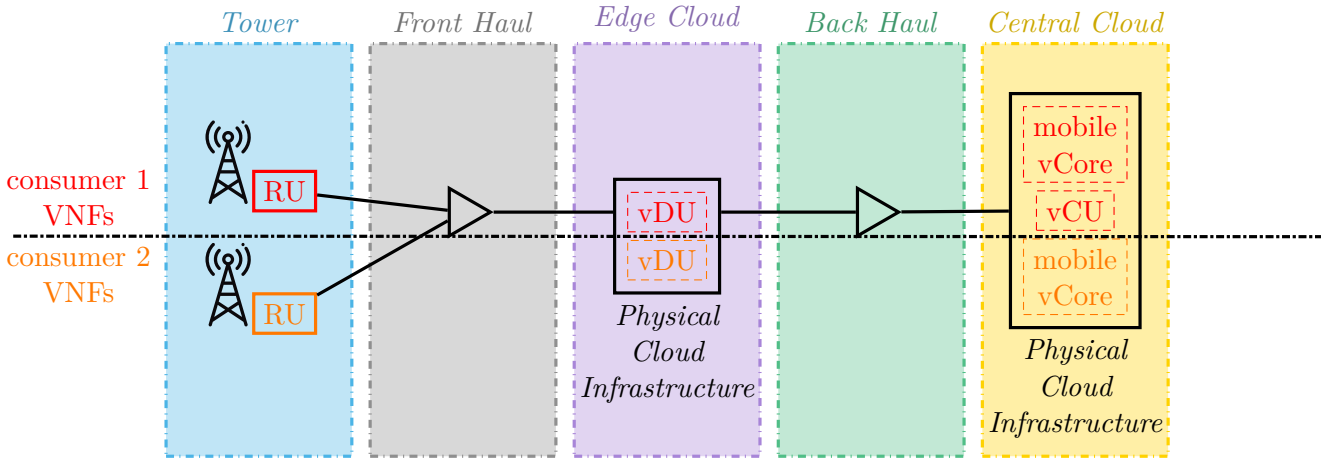


Figure 4.10 – Resource sharing scenario in convergent mobile networks

4.3.2 The data layer proposal

In this work, it is considered to implement a "data layer" to feed a marketplace system with reliable operational reports coming from the infrastructure. This data layer can then be considered as a decentralised oracle from marketplace side, or even as a "layer 2" system if the marketplace is implemented using the DLT. It is further proposed to implement the data layer with the help of a dedicated Distributed Ledger, as to avoid costly trusted third parties. With the proposed model, the collected and shared performance metrics are then processed by decentralised applications in the data layer. This scenario takes advantage of the DLT and its capabilities to achieve truthfulness easily and to reduce reconciliation time. The smart-contracts of the decentralised application are then approved by each party involved in a given SLA within a trusted environment [91]. As the users of the data layer are already known, the data layer's DLT is made permissionned by allowing only authenticated agents to interact with it, as to provide extra security.

Service Assurance

To provide service assurance, various data are collected from the different network domains of the E2E service. These pieces of information can be either, basic data like Bytes flowing through a given network interface, or performance metrics used to compute Key Performance Indicators (KPIs), such as E2E latency. Customised performance requirements can be defined in SLAs while generic performance constraints and KPI formulas are standardised [87, 88] (for example, Ultra-Reliable Low-Latency Communications (uRLLC) services need to guarantee E2E latency under a given threshold). In both cases, each involved player has to comply with the negotiated SLA constraints.

The actual collection methods of the requisite data shall also be determined and defined in the SLAs taking into consideration the required level of service, the slice type, etc. for there exist various methods to achieve operational data collection. The most common method is to regularly retrieve performance metrics from network equipments and send a bundle of data at a regular time interval (for example every 15 minutes). Another possibility is to gather data in real-time: once the metrics are collected by performance management services of network equipments, they will be packed and sent to the management application. This approach is considered in 5G, as some critical applications need to gather metrics in almost real-time. Furthermore, this discharges network equipments from heavy computations as no pre-processing will need to be achieved by them. However, such a method comes with a huge network cost for raw, uncompressed data flow is resource consuming to transmit. The data collection mechanisms in place shall then be defined and agreed among all partners involved in the service as well.

Nonetheless, some specific "*Usage Report (UR) agents*" will need to be deployed as close as possible to the measurement sources, a.k.a "*probes*" to reduce network overhead and take quick actions (e.g onboard more resources) when necessary. These equipments will gather raw measurements (e.g byte flow on a given interface), store them into the data layer's Distributed Ledger, and process them locally with the help of smart-contacts:

* The data will need to be authenticated, analysed and cleaned for detecting/correcting measurement errors. Their timestamping needs also to be reliable, which can be achieved with the help of the COCOS method presented in Subsection 2.3.3. To validate the data itself, a simple threshold rule can be established for two metrics that should be similar. More complex methods based on Cognitive Modules (CMs) might also be considered.
* The compliance with the SLA shall be verified too, as if the specific constraints are not met the players should be notified with alarm to take proper action.

Authentication is necessary as the conformity of probes should also be certified in some way. Indeed, using only approved services will make produced measurements trustworthy for every prosumer. For that purpose, a Software Asset Management (SAM) framework can be used to provide trusted certificates for data produced by probes [92]. Also, data stored into the Distributed Ledger will allow every partner involved in the service to keep an accurate, reliable record of all performance metrics reported by probes. The content of the ledger is then non-repudiable, and can be easily audited to detect any flaws in the system. The ledger may also be audited by the regulator, as regulation aspect is mandatory for the then-created E2E services. Indeed, the auditor/regulator will need to verify that the deployed infrastructure meets the regulation in force. Another aspect that needs to be investigated in some specific use cases is the compliance of the infrastructure with the United Nations (UNs)'s sustainable "*Goal 9: Build resilient infrastructure, promote sustainable industrialisation and foster innovation*" [44].

Then the pre-processed indicators are processed more widely at a higher level to generate

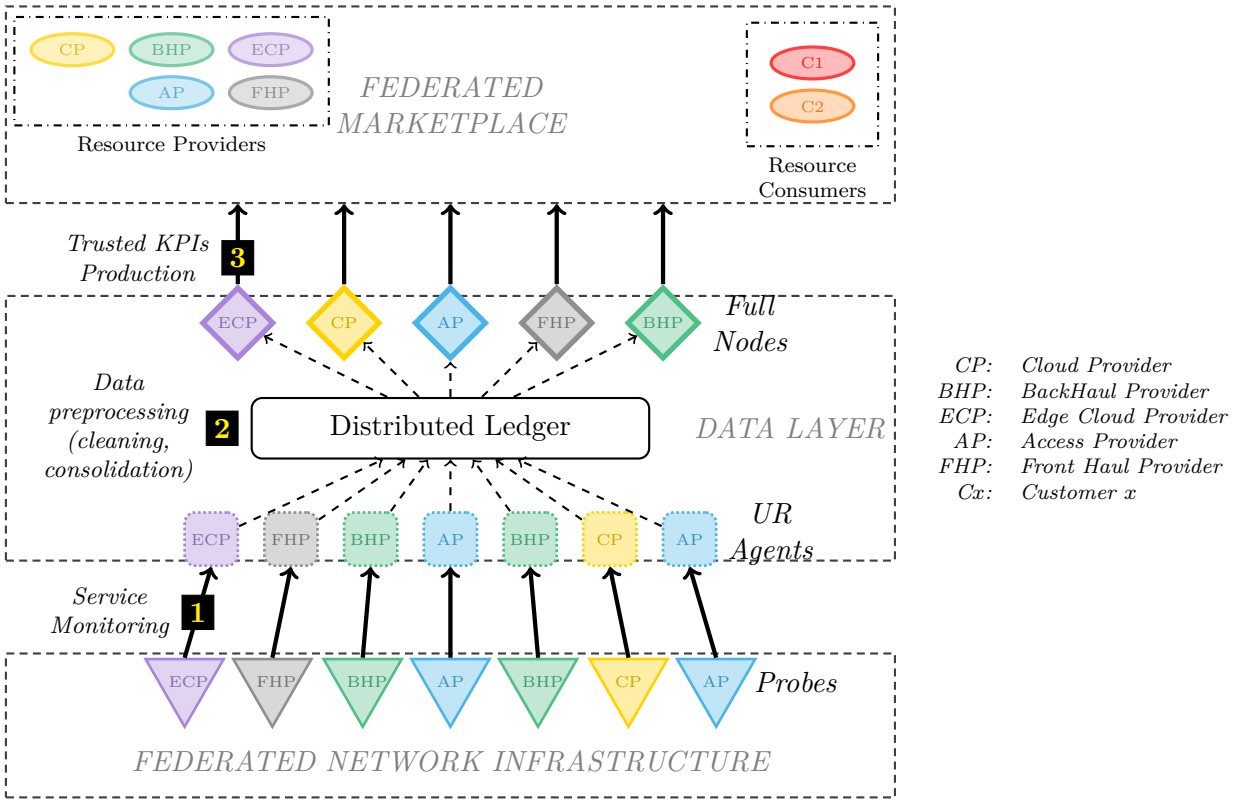


Figure 4.11 – The different nodes implied in the data layer

reliable KPIs, using "*full nodes*" with more resources available. Figure 4.11 then illustrates the processes: metrics are created by probes deployed on each E2E service, then transmitted directly to UR agents. These agents pre-process, clean up the indicators that are then forwarded to the higher level full nodes. Theses nodes will be able to produce the trusted indicators necessary for the marketplace to operate. Depending on the SLA and the measured KPIs, a given prosumer may deploy multiple probes/UR agents on a given E2E service. Yet a single full node per provider for the entire platform would be enough.

Description of the Distributed Ledger processes

This section describes more in depth the different components of the data layer. They are further described on Figure 4.12.

Once submitted, each metric is packed into a transaction readable by the Distributed Ledger and its smart-contracts, with all the necessary identifiers to authenticate its origins (issuer and management service).

The first action taken on the ledger is to check the validity of the signatures (*Data Validation*),

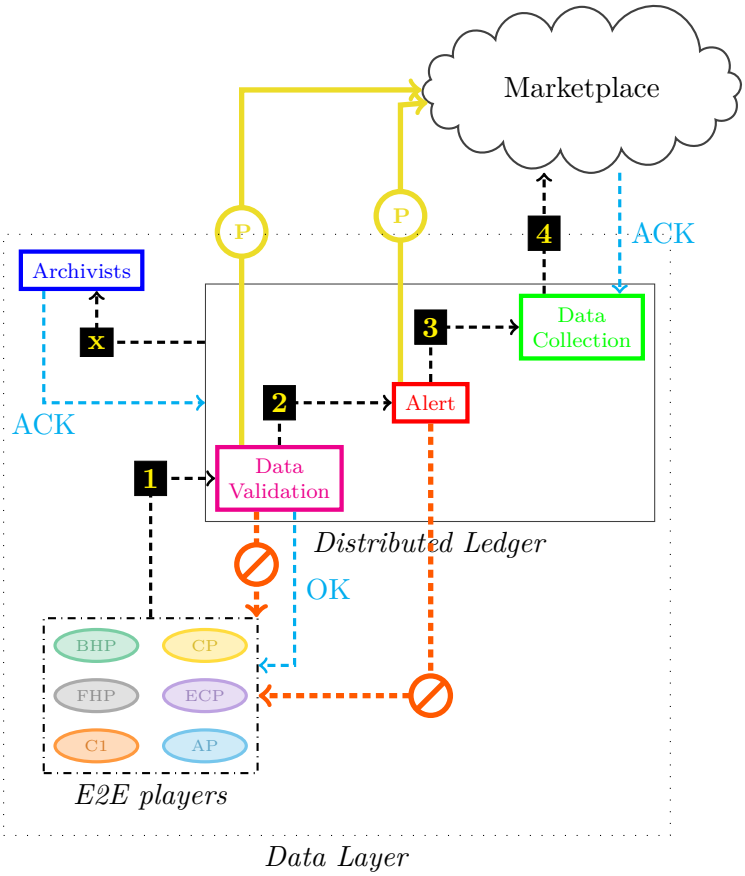


Figure 4.12 – Components to deploy on a Distributed Ledger for the Data Layer

to verify the identity of the issuer/management service.

If any of the signatures are invalid, the transaction will be rejected by the participating nodes as instructed by the smart-contract (step **1**). The participating nodes should also notify the marketplace if too many invalid transactions are submitted so that proper actions could be taken like penalty mechanisms. These notifications are represented as the yellow "P" arrows.

The second step is to check the coherence of gathered metrics, and their compliance with the SLA constraints. As the data collected from the different players need to be aggregated, their overall coherence should indeed be checked (step **2**).

The collection and aggregation mechanisms themselves are outside the scope of this work, and thus remain open questions, although some simple threshold rules could be established taking into consideration the accuracy needed by the SLA, and the possible noise that can be introduced in data collection. It is yet worth noting that such mechanisms depend on the considered metrics, and the users' needs. As the proposed data layer is agnostic to both of these parameters, the exact implementation of these mechanisms should then let be up to partners involved in each deployed E2E service.

When the validation/aggregation mechanisms fail or the resulting metrics do not match the SLA constraints, the event is notified. Thus, both players and marketplace can take proper actions (*Alarm*) which are usually defined within the SLA. These actions can consist of re-allocating resources, triggering penalty mechanisms in the Marketplace, etc.

Once these steps are performed, the validated data can be collected. KPIs are then generated using specific formulas (step **3**) to be used by the marketplace as a reliable input of the connectivity service operation (step **4**).

For legal reasons, a trace of all these processes needs to be archived during a certain amount of time. The specifications of this archive can be specified by regulators and/or within the SLAs (step **x**). On the other hand, the DLT nodes involved in the data layer might have limited storage capabilities. As a result, it is proposed to let *archivist* nodes managed by various players to store and secure the data linked with the E2E services. For that purpose, every participating node can implement the archiving process using the SCRATT method described in Subsection 3.4.2 [13].

The nature of the archivists remains an open topic, as various choices can be made. Actors involved in a E2E service chain may for example designate their full nodes as archivists, or may choose to use the services of a trusted third party like the regulator. Yet the nature of the SCRATT proposal allows multiple possible deployments as the method can be implemented as a smart-contract on the data layer's ledger. Multiple archiving scenarii may then coexist on the data layer, to accommodate every need.

Each inbound/outbound data transfer of the Distributed Ledger (steps **1, 4 and x** on Figure 4.12) needs to be acknowledged, so that the players get assurance about the registration of the transactions. These acknowledgements are however of various nature. At step **1**, the light

node submitting a performance report must monitor the ledger to assess its good commitment. As covered in Chapter 3, this step not only depends on the DLT used to sustain the data layer, but also on the user's policies, as the commitment of a transaction on a Distributed Ledger is a probabilistic statement. As a result, parameters like block's depth or transaction's cumulative weight are to be monitored by light nodes as to consider a given transaction as committed according to their own policies. At step **4**, the acknowledgement from the marketplace may only be implemented as a signed transaction sent by the marketplace system, whose commitment on the data layer is to be monitored. At step **x**, the acknowledgement corresponds to the "pruning signals" described by the SCRATT process [13]. Yet it is worth noting that archivists may be implemented on the full nodes on the marketplace. In that specific case, the acknowledgements of step **2** and step **x** are equal. Nonetheless, light nodes can use these acknowledgements as a proof of the successful commitment of any transaction they submit. Indeed, thanks to the nature of DLTs, such a trusted transaction will also contain a non-repudiable proof of all former transactions.

Multiple choices are further possible for the implementation of the data layer components as smart-contracts. The simplest implementation would be to run these components *off-chain*, only between the prosumers involved in a specific E2E service chain. Using this approach, the ledger would only store data regardless of its accuracy. The validity of transactions would then be assessed only by the prosumers involved in any specific E2E service chain. This approach allows to respect the privacy of the services, as prosumers may encrypt performance reports prior to its storage in the ledger. As such, data associated with a specific E2E service chain would be deciphered only by the prosumers involved in the service.

Yet components of the data layer may also be implemented as *on-chain* smart-contracts, to increase their security (non-repudiation). Using this approach, each single participant in the marketplace processes data from every active E2E service chain. The extra cross-validation would then further strengthen the ledger as a whole, as it will not contain any invalid transactions. Moreover, confidentiality of the data could be still ensured by using Homomorphic encryption, thus allowing any partner to process any encrypted data without even knowing it. Such "private" smart-contracts already exist in the literature [93]. Nevertheless whereas this approach would be more secure than off-chain smart-contracts, it might require more computing resources to handle the extra cryptographic operations needed to preserve privacy in such a case.

Whereas the proposed architecture is also agnostic to the DLT sustaining it, appropriate technology must be selected for optimum operation. Indeed, as detailed in Chapter 3, there are multiple DLTs existing to accommodate multiple needs. Next section is about choosing an appropriate DLT for the proposed data layer.

Requirements for the Data Layer DLT

The main challenge for the data layer is to find an effective DLT solution to accommodate its needs. Indeed, on a global scale such a system is unique for the different asset providers, per the amount of data source, and the amount of data exchanged. The exact performance requirements depend on the services and their negotiated quality. For instance, Industry 4.0 non-public network slice agreements [94], roaming agreements [95], miscellaneous 5G slices of various types (enhanced Mobile BroadBand (eMBB), Ultra-Reliable Low-Latency Communications (uRLLC), etc.) will not realistically have the same requirements on the matter. As it is considered to propose a data layer agnostic to the services deployed on it, it must embrace as much as possible networking use-cases. The worst case scenario (e.g. uRLLC slice with strong QoS constraints requiring a huge load of operational data to be enforced) also needs to be considered, whereas the chosen DLT must remain elastic enough to accommodate to a fluctuating demand, as to achieve a good efficiency [44].

To further evaluate the performance requirements of the data layer, the expected transaction throughput (or *TPS* value) is used as a driving parameter. As per conception, this parameter is directly related to the amount of raw performance reports emitted onto the data layer, considering that each performance report results in a transaction. A simple study has thus been conducted to estimate the volume of performance reports emitted on the data layer. In this study, only raw transactions that would be emitted by mobile base stations in France are considered. Statistics about 4G networks at the time of the study are used as an estimation. The French Agence Nationale des FRéquences (ANFR) has listed about $48,500$ active 4G sites in 2020 [96]. It is considered that each of these sites hosts an active E2E service chain emitting performance reports. Example of such performance reports are the number of connected users, the average DownLink/UpLink throughput, the Signal-to-Noise Ratio Interference (SNRI), etc.) [87]. Operational teams may also use KPIs identified in [88] for assessing accessibility, utilisation, retainability, mobility and energy efficiency. Such performance data already exist today, and are used by operational teams for daily monitoring and troubleshooting of a network infrastructure's performances. Yet the collected metrics/KPIs vary from telco to telco, and are never disclosed for strategic reasons. In this study, it is considered as a rough estimation that 20-30 raw performance metrics are collected every 15 minutes for a given E2E service.

The estimated TPS of the proposed use case can then be estimated as follows:

$$T_{PS} = 48,500 \times 30 \times \frac{1}{900} \approx 1617 tx.s^{-1} \tag{4.4}$$

It should be noted that this calculated value remains a simple estimation, as it can be expected to fluctuate with time. This value depends also on the coverage of the solution. Indeed, in the similar study conducted in Section 4.2.2, higher values were estimated, taking

public Wi-Fi coverage on a worldwide scale as a use-case. Furthermore, some sensitive services like uRLLC slices may also produce more data temporary, for testing purposes.

Nonetheless the estimated value of $1617 tx.s^{-1}$ remains well above the processing capabilities of state-of-the-art DLTs that only scale to a few transactions per second, as per explained in Subsection 3.1.4 and in Section 3.2. It is further assumed in this work that every single raw performance report needs to be secured into the data layer's DLT, for reconciliation purposes if a KPI fails to be produced. As a result, a "simple" layer-2 architecture like lighting [57] can't be used as is, as the data stored into any payment channel isn't secured while the channel remains open. Also, one would except such a use-case to be dynamic with E2E services of different nature being deployed and terminated on-the-fly, thanks to the automated marketplace and autonomous network. Furthermore, any prosumer may join or leave the marketplace at any time, and the network itself will be made elastic as per the use of VNFs instantiated in. Energy and resource consumption need to be also considered when implementing DLT systems since they have a strong impact in some areas of the network [44, 97]. An efficient and scalable DLT is then needed to sustain the development of the proposed solution.

To accommodate these needs, a similar thesis contribution has been presented in Section 4.2. In this work, an architecture with multiple Blockchains (that may be considered as "shards") interconnected thanks to a marketplace system has been considered. This approach might however cause issues due to its relative complexity. Furthermore, this approach might not be efficient enough, for a Blockchain capacity is fixed by nature and won't possibly accommodate for a fluctuating transaction arrival. On the other hand, implementing a general Distributed Ledger instance enhances the security of the transactions of all service chains. This approach also prevents the issues associated with sharded (parallel) Distributed Ledgers instances and therefore prevents conflicting operations between prosumers. This will also enforce the trust in the Distributed Ledgers instance regarding its adoption. As a result, novel DLT technologies going beyond Blockchain are further considered in this work.

The advantages of DAG-based DLTs

As explained in Section 3.2, scalability can be enhanced by making a DLT asynchronous using various techniques. These techniques include sharding [62], protocols above a main Blockchain [57] or more ambitious projects based on other data structures [11, 71, 72]. As further developed in Subsection 3.2.2, DAG-based DLTs may also achieve a better efficiency as these technologies typically adapt themselves to a fluctuating arrival of transactions.

In this section a DAG-based ledger is considered for the proposed use-case. This choice is motivated both by the state of the art, and by former contributions on this thesis. Indeed, DAG-based technologies are actually well studied, and have already some existing implementations such as IoTA [11], Nano [71] or Hashgraph [72]. Moreover, DAG-based DLTs are known to be

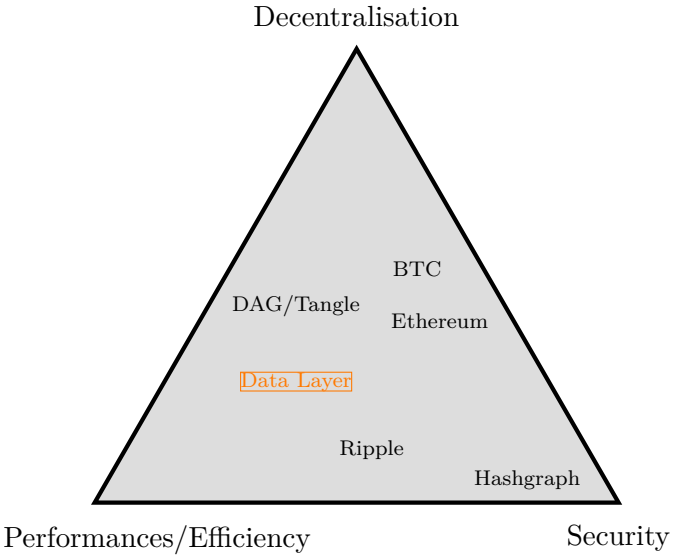


Figure 4.13 – The Trilemma of Distributed Ledgers

very efficient and scalable due to their asynchronous nature [10, 98]. Unlike sharded Blockchain, using an unique DAG instance shared among every E2E service enables each player to confirm the transactions of anyone, and thus to strengthen the overall ledger. Furthermore, a similar use-case has also been assessed in this thesis work in Section 4.2, and the Blockchain scenario presented mitigated performance results.

The Tangle implementation is further taken as a reference, as it is the oldest, strongest and most open DAG-based project, and has been well studied through time [11, 12, 25, 26, 73–77, 99]. While public Tangle seems to present security issues allowing the presence of "lazy nodes" or attackers [11], here the authentication and certification of participating nodes solve these issues. Furthermore, the marketplace system may help participants to assess the commitment of KPIs as from data layer side it behaves like a trusted third party.

Figure 4.13 then summarises the above exposed claims about this technical choice: DAG-based ledgers seem to be the closest to the performances/security/decentralisation balance needed by the data layer.

4.3.3 Assessing the Tangle behaviour with the proposed use-case

In this section the behaviour of the Tangle when confronted to the proposed use-case is assessed with the simplified parameters presented in the previous sections. The evaluation is notably comforted with simulations conducted for this work [12], and presented in Subsection 3.3.4.

As per explained in Section 4.3.2, the expected TPS of the proposed use case would be then

of roughly $1,617 tx.s^{-1}$.

Such transactions can be expected to be generated on conditions similar to low end hardware, with limited resources. Work cited in [99] has shown that as few as 687 ms are needed to prepare and sign a transaction on embedded systems; while this time grows to approximately 300 seconds when the emitter is required to solve a PoW. However, in the proposed case this outcome won't be first covered as such a PoW should not be needed. Indeed, as the transaction emitters (a.k.a UR agents) of the use-cases are identified thanks to the system managing the collaboration environment, one could assume the transaction rate of any specific agent to be controllable. Indeed, if transactions are emitted by identified softwares trusted by everyone [92], it can be assumed that they won't perform any DDoS attacks or equivalent. As a result, a PoW as a rate control mechanism wouldn't be needed in the proposed use-case.

Let us define arbitrary the time needed to produce a transaction $h = 1s$. The remaining 313 milliseconds would then represent the time to compute the data of the transaction and for the then-signed transaction to propagate on the network.

Taking into consideration the continuous time model of the Tangle, validated by simulation in Subsection 3.3.4, the average number of tips L_0 would be approximately of $2 \times 1617 \times 1 = 3234$ tips. In a similar way, the "adaptation period" of the Tangle can be approximated as 23 seconds. According to IoTa foundation's claims, this period corresponds to the time after which any given transaction is "globally" validated, as every new arriving transaction will validate it [11].

Reciprocally, if an archivist or a marketplace full node looks the Tangle at time t, then there is a high probability that all transactions committed before $t - 23s$ will be visible to this actor. As a result, archivists could send pruning signals every 30 seconds, as there is a high probability that they will get the ledger as a whole.

Using the same formula with $h = 300s$, if a PoW is used as described in [99], this time would grow to 11,745 seconds, which is approximately 3 hours and 15 minutes.

While rate control could be achieved by alternate means than PoWs in the proposed case, this indicates that the time needed to process a transaction has a strong impact on the capacity of the Tangle to converge in a limited time.

4.3.4 Discussion & perspectives

In this section, a decentralised "data layer" architecture has been presented. This architecture allows the collection of trusted, reliable operational data from a multi-actor connectivity infrastructure, in order to provide service assurance. The data layer indeed produces trusted KPIs, usable as trusted input for the network management system. The proposed architecture thus solves the need for trusted, reliable operational data on such collaborative environments, discussed in Chapter 2.

The proposed architecture remains agnostic to the underlying infrastructure, as well as being

agnostic to the marketplace system. As a result, it may be overlayed on various types of marketplace systems, either centralised or decentralised like the use-case presented in Appendix A [2], or the use-case presented on Subsection 2.4.3 [7]. Furthermore, while only mobile, C-RAN based network infrastructures were considered on this study, the proposed data layer may support various other types of network, either fixed or mobile. As an example, it may also give host to the BALAdIN proposal presented on Section 4.2 by facilitating the production and processing of PoB transactions.

The Tangle, a DAG-based DLT [11] has also been selected as the driving DLT capable of powering the proposed data layer. This choice accounts for the literature review of the DLT, and simulations of the Tangle, presented in Chapter 3. Indeed, unlike the Blockchain, such a DLT is indeed elastic enough to accommodate to a varying and possibly high amount of transactions to process, while remaining resource efficient [44, 97].

Work performed on this proposal indicates that a Tangle or Tangle-like DLT would accommodate the needs of the data layer, accounting for a TPS of 1617. Such DLTs should also be able to handle higher TPSs values, thanks to their asynchronous nature allowing them to scale.

The proposed data layer further takes advantage of other contributions of this thesis work. The COCOS method presented in Subsection 2.3.3 can indeed help actors produce data correctly timestamped prior to their storage in the DLT, while the SCRATT method presented in Subsection 3.4.2 may help with the deployment of the data layer's DLT, by reducing the impact of the technology on storage capabilities.

It is worth noting that due to the wide range of possible networking use-cases, the implementation of many components of the data layer such as smart-contracts processing performance reports remains up to the actors involved in the decentralised E2E service chains. As a result, operational data can also be of various nature, either standardised [87, 88] or customised. Furthermore, the QoS needs for the delivered connectivity services may vary a lot, as well as the required volume of KPIs needed to assure them. This fluctuating, almost-unpredictable volume of data to process further motivates the choice of a DAG-based DLT.

Yet the DLT is a rapidly evolving technology, for many new innovations emerge each year. As an example, whereas the first version of the Tangle was considered in this study, the protocol has further been updated for performance improvements, as well as avoiding too much centralisation [25]. While the proposed study seems to indicate that the Tangle is appropriate to sustain the data layer, future work may consider other DLTs to sustain the proposed data layer.

Nonetheless, the work presented in this section seems to indicate that in the proposed scenario, a Blockchain-based DLT isn't fit to support the proposed data layer on its own, as per the high, varying and almost unpredictable amount of data to process. This claim is further motivated by the study of the BALAdIN proposal presented in Section 4.2. Indeed, on this

work, a Blockchain approach is considered for a similar use-case, and its implementation adds complexity [18].

While a generic view of the data layer has been proposed in this section, future work can then focus on proposing a more detailed architecture of the data layer by considering its implementation for more specific use-cases.

4.4 Conclusion

In this chapter, the question of providing trusted, reliable operational data from a multi-actor connectivity infrastructure to its management system has been explored. On collaborative connectivity use-cases like explored in Chapter 2, having access to reliable operational data from the network infrastructure is indeed a requirement. This problem is further generalised as the "oracle" problem in decentralised applications needing data outside of their distributed ledgers.

During this thesis work, decentralised solutions have further been explored, running thanks to the DLT. As first, a Blockchain approach is considered with BALAdIN to provide operational data. Yet work showed that a Blockchain-based DLT barely sustains such a use-case, for this technology misses the requisite elasticity.

Then, a "data layer" architecture has been proposed to provide agnostic service assurance to multiple actors building E2E virtualised connectivity service chains. On this work, DAG-based DLTs are rather considered to achieve a better efficiency of the ledger. It should yet be worth noting that whereas this contributions introduces the framework allowing telcos to share operational data, the nature of the performance data themselves, as well as the applications required to process them are not defined. This question is hard to anticipate, as while there is ongoing work on standardisation about performance data [87, 88], the performance monitoring mechanisms of network infrastructures are usually never disclosed for strategic reasons. Furthermore, while the DLT provides a trusted, easily auditable database, further work is required on securing performance report themselves. This particular topic actually depends on the nature of the performance reports themselves, as well as the underlying use-cases. The implementation of the data layer is also dependant on the definition of decentralised collaborative connectivity initiatives like explored in Chapter 2 supported by the data layer, as well as the work performed on standardising KPIs [87, 88] and truthfulness components necessary besides the data layer's DLT. These requisite components include but are not limited to software certification [92], and truthfulness architecture enabling data securisation [21].

To summarise the points listed above, a successful data layer must meet the following conditions:

* The DLT supporting the data layer must have enough performances, especially on the maximum transaction throughtput (TPS) it can handle. The exact required TPS then depends

on the connectivity services using the data layer, and their own special requirements.

* Furthermore, the DLT nodes' resource requirements must also match the constraints of the environment their are deployed into, for a successful operation of the data layer. It should be noted that DAG-based DLTs can meet these first two conditions, as they achieve a better elasticity on the TPS than Blockchain-based ones, while being more resource-efficient.

* Finally, truthfulness components will need to be deployed onto the data layer for securing and cleaning the raw performances indicators. Depending on the connectivity services, such components can be of various nature (software certification, clearing rules, etc.). Is is worth noting that such truthfulness components may be deployed as smart-contracts on the data-layer's Distributed Ledger, depending on the underlying DLT .

While enhanced collaboration between telcos remains an emerging topic nowadays, the works presented on this chapter set the root of processes allowing the production of the required trusted operational data needed to operate such collaborative networks.

Chapter 5

Conclusions and perspectives

Contents

5.1 Main contributions

In this thesis work, the question of trust mechanisms securing collaborative network infrastructures has been explored. It is worth noting that although there already exist multiple use-cases of cooperation among telcos such as international roaming, this topic remains rapidly evolving. New technologies like cloud computing indeed allow communication networks to become more elastic as they are migrated from dedicated hardware to VNFs deployed on commodity hardware. The DLT can further foster collaboration by providing a database whom content can be trusted to involved partners, without any trusted third parties.

The results of this thesis shall then facilitate the deployment and operation of trusted, reliable E2E connectivity service chains built by multiple actors. To build such multi-actor connectivity services, first a trusted collaboration is needed among actors delivering such disaggregated E2E services. This collaboration relies on reliable operational performance data needing to be trusted by every partners involved in the multi-actor services. Such data is indeed necessary to enable the pro-active management of the multi-actor connectivity services, as to guarantee a QoS and provide service assurance to the users.

The main contribution of this thesis work is the definition of a decentralised, DLT-based *"data layer"* architecture enabling the production of trusted, reliable E2E KPI from an E2E connectivity service delivered by multiple distinct telco and/or non-telco partners.

Then multiple other contributions are presented to further explore the future collaboration architectures, and to facilitate a future deployment of the proposed data layer architecture. These contributions are detailed below.

5.1.1 On infrastructure optimisation for collaboration

At first, the improvement of the infrastructure for sustaining collaboration has been explored in this thesis work. This thesis work has been the opportunity to contribute in a method aiming at facilitating interaction between a collaboration architecture, taking the IDSA architecture as a use-case, and the telcos providing the underlying infrastructure [4]. The proposed architecture takes both advantage on telcos' ability to adapt their infrastructure depending on the needs, but also on Multi-access Edge Computing (MEC) capabilities enabling the deployment of any application processing data in infrastructures close to the data sources, hence limiting network overhead.

Furthermore, work has been done in this thesis to propose a trusted time synchronisation mechanism between partners collaborating on a trustless environment, so that data exchanged on a collaboration environment can be timestamped with a time source trusted by everyone. The proposed method takes advantage of consensus mechanisms like the PoW or the PoS for that purpose, so that partners can reach a consensus about the time reference to follow, and that in a decentralised way [13].

5.1.2 On collaborative architectures

This thesis work also led us to participate in contributions on novel collaborative network architectures. At first, a mobile infrastructure sharing scenario is considered, aiming at helping MNOs to share their infrastructures [2]. For that purpose, energy saving is considered as the driving argument incentiving telcos to collaborate. MARL algorithms are then considered to evaluate the scenario. Results showed that collaboration must be enforced by an environment driving collaboration and eventually blacklisting non-collaborative MNOs. This environment itself requires trusted, reliable readings of the MNOs' energy consumption to successfully operate.

Then another contribution on a federated connectivity marketplace has been proposed [7]. This contribution was the result of international collaboration with multiple telco and non-telco partners. It introduces a marketplace architecture allowing multiple actors related to the connectivity ecosystem to automatically trade connectivity-related digital resources. With the proposed architecture, "asset providers" can publish various types of offer (connectivity, cloud, MEC, etc.). These resources may then be used by "service providers", deploying virtualised network elements on the asset providers' infrastructures in order to serve their final customers. In this work, it is evidenced that QoS must be enforced for successful collaboration. Some connectivity services, like C-RAN, or uRLLC slices are indeed very sensitive to the delivered

QoS, considering parameters like latency, jitter or max throughput. To achieve QoS enforcement, trusted, reliable performance metrics (KPIs) must also be provided to the collaboration environment.

The problem of providing such trusted, reliable operational data was the main problem of this thesis work. Indeed, both state of the art and contributions of this thesis on collaboration between telcos showed that having access to trusted data is mandatory for successful collaboration. This problem extends beyond connectivity-related use-cases, as some decentralised applications running on the DLT require trusted "oracles" to interact with elements outside the boundaries of the ledger when it is required by the underlying use-case. As a result, this thesis work strives to provide architectures enabling the production of trusted, reliable operational data for telecommunication use-cases. The usage of the DLT is further considered as a major component of the architectures considered, to avoid to rely on too many trusted third parties.

5.1.3 On DLT research

The DLT was then studied as part of this thesis work, to consider its usage for allowing the collection of trustworthy performance reports from a multi-actor connectivity infrastructure. This study is made necessary as the usage of the DLT for the production of trusted performance data is quite unique, and afar from more regular DLT use-cases like cryptocurrencies. Indeed, one needs to account for a high and fluctuating throughput of data to process, whom exact amount cannot be predicted, both due to the opacity of each telco's practices on performance monitoring, and to the multiple variations of possible use-cases needing performance monitoring. Resource & energy efficiency should also be considered, as for this use-case the DLT will be deployed in a resource-sensitive environment.

This thesis's work concluded that Blockchain-based DLTs aren't fit to sustain such a use-case, for they scale poorly. DAG-based ledgers have been rather studied, as they characteristics seemed more fit to the proposed use-case. The Tangle, one of the earliest DAG-based ledger proposal has more particularly been studied.

This allowed to propose a contribution on the simulation of the technology, as to evaluate its pertinence for producing trusted operational data [12].

Another issue investigated in this thesis is the storage required for operating a DLT. Indeed, the "add-only" nature of the technology implies that the amount of required storage increases continuously. To address this issue, a decentralised archiving method has been proposed in this thesis work. The proposed contribution allows multiple collaborating partners using any DLT to delegate the storage of historical ledger data to trusted "archivist" of their choice. This contribution is important in the context of this work, as users of collaboration environments like presented in this work might not have the same policies regarding data archiving, as well as their storage.

5.1.4 On truthfulness protocols

Finally, in this thesis work multiple DLT-based architectures were proposed to enable the production of trusted operational data. At first, the "Proof of Bandwidth" mechanism has been studied in our BALAdIN proposal. This mechanism allows to monitor the usage of a multi-actor network path by its users, in a decentralised way. The PoB takes advantage of cryptographic computations for providing operational data validated by every partners involved in such mutli-actor network paths. These operational data can be further used for accounting. The deployment of the solution, and more particularly of the PoB providing trusted operational data is more particularly considered. The study showed that such a solution might be complex to deploy using a regular Blockchain, for it thus requires sharding mechanisms.

Then, a decentralised "data layer" architecture has been proposed as a more generic trusted source of performance report. The goal of the data layer is to propose an architecture enabling any telcos collaborating to build multi-actor network infrastructures to produce trusted Key Performance Indicators (KPIs). The use-case considered is the "federated connectivity marketplace" also presented in this thesis work. The proposed architecture particularly uses the DLT as to provide a trusted, easily auditable record of the operation of monitored networks, through their KPIs. In this thesis work, DAG-based DLTs are rather considered, thanks to their increased efficiency compared to Blockchain-based ones. The work further allowed to propose an architecture for implementing the proposed data layer, taking C-RAN based shared networks, and the Tangle technology as the driving DLT. This work takes advantage of other contributions of this thesis, on Tangle modelling and DLT transactions archiving [12].

The work of this thesis focused on the question of trust mechanisms enabling collaboration. The major contributions of this work are summarised on Figure 5.1.

5.2 Research perspectives

In this thesis work, a generic "*data layer*" enabling multiple collaborating connectivity-related actors to produce trusted operational data has been proposed. Then various contributions on specific truthfulness mechanisms of such a framework have been explored. It is worth noting that the proposed data layer remains agnostic of the collaboration scenarii themselves. As a result, also considering the rapidly evolving telecommunication ecosystem and the vast variety of collaboration use-cases in telecoms, this thesis opens the door to many research opportunities :

* At first, new collaboration opportunities between telcos, as well as their requirements and challenges can be better explored, for new needs in telecommunication networks are to be taken into account. It is worth noting that such studies can take advantage of MARL algorithms to model actors' interaction, like explored in this thesis.

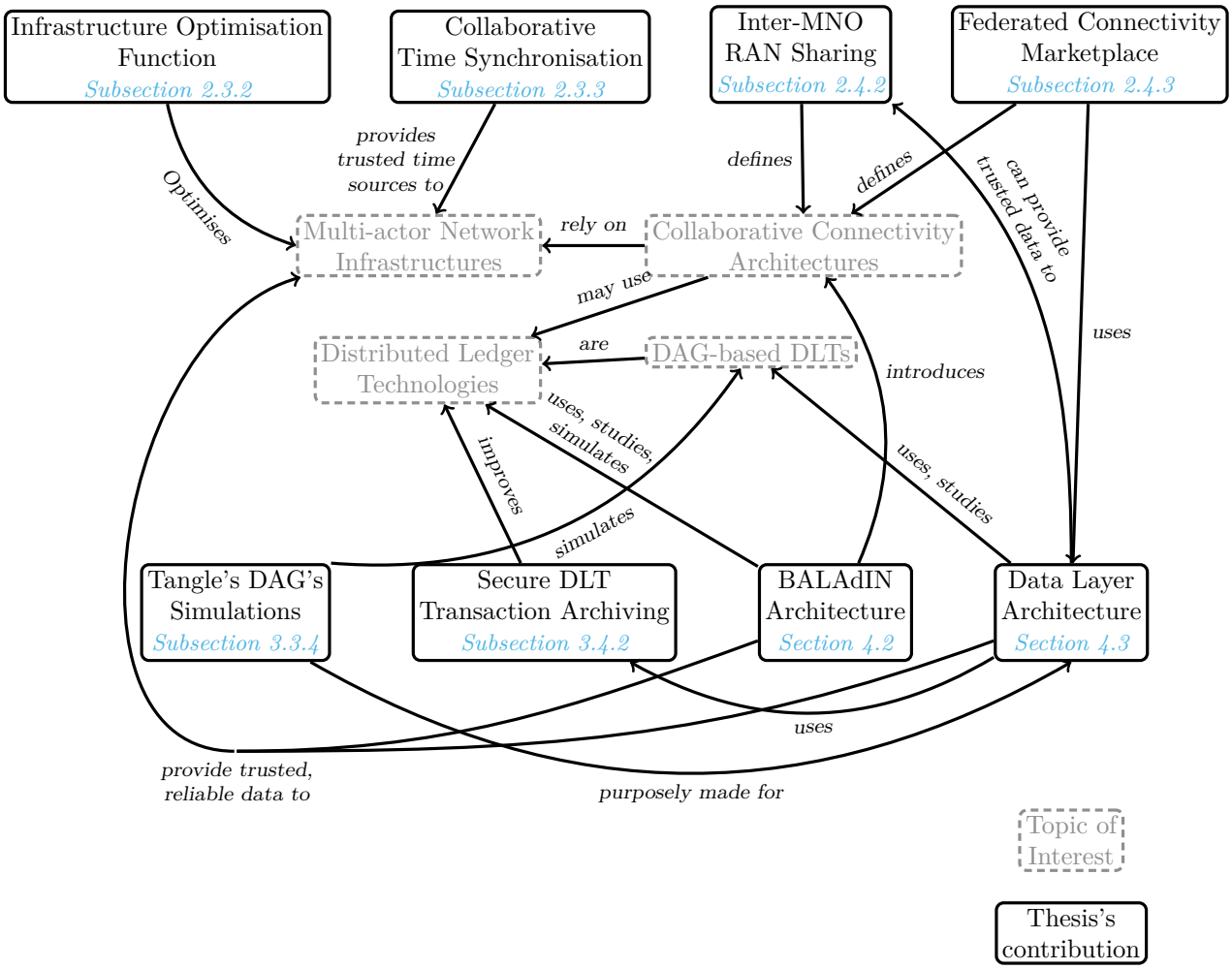


Figure 5.1 – Major outcomes of this thesis work

* Concerning service assurance, work can be made to further evaluate the required metrics and necessary KPIs, as well as the frequency of their collection. It should be noted that these parameters remain use-case specific, and the required frequency of collection might also vary a lot. Furthermore, such parameters might currently remain opaque for strategic reasons. For future investigation of the question, the current work on KPI standardisation may be useful [87, 88].
* To ensure the necessary confidentiality of KPIs produced through a shared environment like a DLT, Zero Knowledge Proofs (ZKPs)-based algorithms can be explored. These algorithms allow the operation of an *on-chain* smart-contract in a public ledger while keeping its data ciphered. Yet it should be noted that zero-knowledge operations involve extra cryptographic operations that need to be performed for each usage of the underlying contract. These extra operations are heavily dependent on the underlying contract computations (addition, substraction, comparison, etc.) and their complexity might vary at lot. The implementation of ZKP algorithms is then KPI-specific, and thus also use-case specific.
* Similarly, the reliability of the performance metrics prior to their storage/KPI computation may also be investigated. This topic remains wide, as many mechanisms can be set up to achieve this primary truthfulness. First, certification of the performance data sources, either software [92] or hardware [37] can be imagined. However such a work will require a tight collaboration with the software/hardware equipments' vendors since their policy in certification might vary. Also, a redundancy of the data sources can further secure specific measurements that can profit from redundancy. As an example, the PoB mechanism explored in this work requires every implied actors' approval on the amount of traffic exchanged, a metric whose divergence between each measurement point should be minimal. Furthermore, depending on the collaboration environment, assumptions of the honest behaviours of the agents producing the performance metrics can be made. Depending on the collaboration use-case, game theory can indeed be used to assess the personal interest for each actor to act honestly.
* Then, work can be sustained on the DLT regarding the use-cases proposed in this work. The proposed infrastructure monitoring use-cases indeed account of a big, varying amount of micro-transaction whose atomic value remains low compared to crypto-currencies. The DLT can indeed still be considered as "experimental", for many evolutions of the technology are still explored nowadays to improve the technology, being on security, performances, scalability, and also energy/resource efficiency.
* Finally, the question of the resource/energy overhead of such trust mechanisms should also be explored. This topic may be challenging, for the real impact of such mechanisms will heavily depend on the collaboration use-cases, their required implementations, as well as on the DLT used if applicable, and the hardware used to support the network infrastructure.

5.3 Publication list

Conference papers

* X. Marjou, T. Le Gléau, V. Messié, B. Radier, T. Lemlouma, and G. Fromentoux, "Evaluating Inter-Operator Cooperation Scenarios to Save Radio Access Network Energy", in *2022 1st International Conference on 6G Networking (6GNet)*, Jul. 2022, pp. 1–5. DOI: 10.1109/6GNet54646.2022.9830283
* V. Messié, B. Radier, V. K. Quintuna Rodriguez, G. Fromentoux, S. Vaton, and I. Amigo, "A decentralised data layer for collaborative End-to-End service assurance", in *2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, Mar. 2022, pp. 81–85. DOI: 10.1109/ICIN53892.2022.9758094
* V. Messié, G. Fromentoux, X. Marjou, and N. Labidurie, "BALAdIN for blockchain-based 5G networks", in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, IEEE, 2019, pp. 201–205. DOI: 10.1109/ICIN.2019.8685867

Journal articles

* V. Messié, G. Fromentoux, N. Labidurie, B. Radier, S. Vaton, and I. Amigo, "BALAdIN: truthfulness in collaborative access networks with distributed ledgers", *Annals of Telecommunications*, Jun. 6, 2021, ISSN: 1958-9395. DOI: 10.1007/s12243-021-00855-x

Patents

* V. Messié, B. Radier, G. Fromentoux, and A. Braud, "Procédé de gestion d'un registre local d'un noeud appartenant à un ensemble de noeuds contribuant à un registre distribué", French pat. 2 105 671, Filed, 2022
* V. Messie, B. Radier, A. Braud, and G. Fromentoux, "Procédé de synchronisation d'une pluralité de serveurs de communications, dispositifs et programmes d'ordinateurs correspondants", French pat. 3114712A1, Apr. 1, 2022. [Online]. Available: https://patents.google.com/patent/FR3114712A1/fr?inventor=messi%C3%A9&oq=inventor:messi%C3%A9 (visited on 05/17/2022)
* B. Radier, G. Fromentoux, A. Braud, and V. Messié, "Procédé de traitement d'un service de transport de données", pat. WO2022034273A1, Feb. 17, 2022. [Online]. Available: https://patents.google.com/patent/WO2022034273A1/en?inventor=messi%C3%A9&oq=inventor:messi%C3%A9 (visited on 05/17/2022)
* V. Messié, G. Fromentoux, and N. Omnes, "Method for preparing usage data for relays used during a communication between two devices and for searching for the data and associated devices", U.S. Patent 20210092110A1, Mar. 25, 2021. [Online]. Available: https://patents.

google.com/patent/US20210092110A1/en?inventor=messi%C3%A9&oq=inventor:messi%C3%A9 (visited on 05/17/2022)

Miscellaneous

* A. Adhiappan, A. Chernetsov, M. Fenomenov, U. Karabudak, A. Korabanova, S. Kislyakov, L. Le Beller, M. Nati, B. Radier, A. Sushkov, A. Ustimenko, A. Vedin, O. Yurlov, T. Ben Meriem, V. Messié, and N. Omnes, "Federated CSPs Marketplace : A DLT-based Data Trust enabling Business Assurance for CSPs Platforms Federation", TM Forum, White Paper 1.0, Nov. 13, 2020. [Online]. Available: https://www.tmforum.org/vertical-industry-telcos-federated-dlt-based-marketplace/ (visited on 09/20/2022)

Bibliography

[1] J. Vincent. (Jul. 4, 2016). "UN condemns internet access disruption as a human rights violation", The Verge, [Online]. Available: https://www.theverge.com/2016/7/4/12092740/un-resolution-condemns-disrupting-internet-access (visited on 08/30/2022).

[2] X. Marjou, T. Le Gléau, V. Messié, B. Radier, T. Lemlouma, and G. Fromentoux, "Evaluating Inter-Operator Cooperation Scenarios to Save Radio Access Network Energy", in *2022 1st International Conference on 6G Networking (6GNet)*, Jul. 2022, pp. 1–5. DOI: 10.1109/6GNet54646.2022.9830283.

[3] E. Debeau and V. Quintuna-Rodriguez, "ONAP: an open source toolkit for zero touch automation", in *Design Innovation and Network Architecture for the Future Internet*, IGI Global, 2021, pp. 212–249, ISBN: 978-1-79987-646-5. DOI: 10.4018/978-1-7998-7646-5.ch008.

[4] B. Radier, G. Fromentoux, A. Braud, and V. Messié, "Procédé de traitement d'un service de transport de données", pat. WO2022034273A1, Feb. 17, 2022. [Online]. Available: https://patents.google.com/patent/WO2022034273A1/en?inventor=messi%C3%A9&oq=inventor:messi%C3%A9 (visited on 05/17/2022).

[5] G. Chidambaranathan, T. Ben Meriem, B. Radier, P. Genestier, V. Purohit, Gnanapriya Chidambaranathan, Seshadri, Luxman, Nixon,Mark, Nati, Michele, Adhiappan, Anand, Karabudak,Umut, and Spencer,Thomas, "CSP Use Cases Utilizing Blockchain", TM Forum, TR 279, Aug. 8, 2019. [Online]. Available: https://www.tmforum.org/resources/technical-report/tr279-csp-use-cases-utilizing-blockchain-v3-1/.

[6] V. Messie, B. Radier, A. Braud, and G. Fromentoux, "Procédé de synchronisation d'une pluralité de serveurs de communications, dispositifs et programmes d'ordinateurs correspondants", French pat. 3114712A1, Apr. 1, 2022. [Online]. Available: https://patents.

google.com/patent/FR3114712A1/fr?inventor=messi%C3%A9&oq=inventor:messi%C3%A9 (visited on 05/17/2022).

[7] A. Adhiappan, A. Chernetsov, M. Fenomenov, U. Karabudak, A. Korabanova, S. Kislyakov, L. Le Beller, M. Nati, B. Radier, A. Sushkov, A. Ustimenko, A. Vedin, O. Yurlov, T. Ben Meriem, V. Messié, and N. Omnes, "Federated CSPs Marketplace : A DLT-based Data Trust enabling Business Assurance for CSPs Platforms Federation", TM Forum, White Paper 1.0, Nov. 13, 2020. [Online]. Available: https://www.tmforum.org/vertical-industry-telcos-federated-dlt-based-marketplace/ (visited on 09/20/2022).

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf (visited on 09/20/2022).

[9] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain", *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 101–128, 2018. DOI: 10.3745/JIPS.01.0024.

[10] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks", *IEEE Communications Surveys & Tutorials*, 2020, Publisher: IEEE, ISSN: 2373-745X. DOI: 10.1109/COMST.2020.2969706.

[11] S. Popov, "The tangle", the IoTa Foundation, 1.4.3, 2016. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf (visited on 09/20/2022).

[12] V. Messié, B. Radier, V. K. Quintuna Rodriguez, G. Fromentoux, S. Vaton, and I. Amigo, "A decentralised data layer for collaborative End-to-End service assurance", in *2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, Mar. 2022, pp. 81–85. DOI: 10.1109/ICIN53892.2022.9758094.

[13] V. Messié, B. Radier, G. Fromentoux, and A. Braud, "Procédé de gestion d'un registre local d'un noeud appartenant à un ensemble de noeuds contribuant à un registre distribué", French pat. 2 105 671, Filed, 2022.

[14] V. Messié, G. Fromentoux, and N. Omnes, "Method for preparing usage data for relays used during a communication between two devices and for searching for the data and associated devices", U.S. Patent 20210092110A1, Mar. 25, 2021. [Online]. Available: https://patents.google.com/patent/US20210092110A1/en?inventor=messi%C3%A9&oq=inventor:messi%C3%A9 (visited on 05/17/2022).

[15] V. Messié, G. Fromentoux, X. Marjou, and N. Labidurie, "BALAdIN for blockchain-based 5G networks", in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, IEEE, 2019, pp. 201–205. DOI: 10.1109/ICIN.2019.8685867.

[16] *Free Wi-Fi Anywhere | How to get O2 Wifi | O2*. [Online]. Available: https://www.o2.co.uk/connectivity/free-wifi (visited on 05/19/2020).

[17] K. Wang, "Ethereum: Turing-Completeness and Rich Statefulness Explained", *Hacker Noon*, Jul. 2017. [Online]. Available: https://hackernoon.com/ethereum-turing-completeness-and-rich-statefulness-explained-e650db7fc1fb (visited on 09/20/2022).

[18] V. Messié, G. Fromentoux, N. Labidurie, B. Radier, S. Vaton, and I. Amigo, "BALAdIN: truthfulness in collaborative access networks with distributed ledgers", *Annals of Telecommunications*, Jun. 6, 2021, ISSN: 1958-9395. DOI: 10.1007/s12243-021-00855-x.

[19] Z. Yan, P. Zhang, and A. Vasilakos, "A survey on trust management for Internet of Things", *Journal of Network and Computer Applications*, vol. 42, pp. 120–134, 2014, Publisher: Academic Press. DOI: 10.1016/j.jnca.2014.01.014. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84901427258&doi=10.1016%2fj.jnca.2014.01.014&partnerID=40&md5=1dc0f5eb0728b4f27ec7bdc184a5792d.

[20] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A Flexible Platform for 5G Research", *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, Oct. 10, 2014, ISSN: 0146-4833. DOI: 10.1145/2677046.2677053.

[21] B. Otto, S. SteinbuSS, A. Teusher, S. Lohmann, S. Auer, S. Bader, H. Bastiaansen, H. Bauer, P. Birnstil, M. Böhmer, J. Bohn, G. Böge, U. Brettner, G. Brost, J. Ceballos, J. Cirullies, C. Ciureanu, E. Corsi, S. Dalmolen, S. Danielsen, A. Duisberg, A. Eitel, T. Ernst, F. Fournier, M. Franz, S. Geisler, J. Gelhaar, R. Gude, C. Haas, J. Heiles, B. Heisen, J. Hierro, J. Hoernle, M. Huber, C. Jung, J. Jürjens, A. Kasprzik, M. Ketterl, J. Koetzsch, P. Sorowka, G. Spiegelberg, M. Spiekermann, C. Spohn, G. Stöhr, E. Tanger, M. Theß, S. Tramp, M. Wappler, A.-C. Weiergräber, S. Wenzel, O. Wolff, and H. Wörner, "International Dataspace Association - Reference Architecture Model", International Data Spaces Association, V3.0, 2019. [Online]. Available: https://internationaldataspaces.org//wp-content/uploads/IDS-Reference-Architecture-Model-3.0-2019.pdf (visited on 06/01/2022).

[22] N. Szabo, "Smart Contracts", 1994. [Online]. Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html (visited on 08/13/2020).

[23] A. Pasdar, Z. Dong, and Y. C. Lee, *Blockchain Oracle Design Patterns*, Jun. 17, 2021. arXiv: 2106.09349.

[24] V. Dhillon, D. Metcalf, and M. Hooper, "The DAO hacked", in *Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You*, V. Dhillon, D. Metcalf, and M. Hooper, Eds., Berkeley, CA: Apress, 2017, pp. 67–78, ISBN: 978-1-4842-3081-7. DOI: 10.1007/978-1-4842-3081-7_6.

[25] S. Popov, H. Moog, D. Camargo, A. Capossele, V. Dimitrov, A. Gal, A. Greve, B. Kusmierz, S. Mueller, A. Penzkofer, *et al.*, "The Coordicide", 2020. [Online]. Available: https://files.iota.org/papers/20200120_Coordicide_WP.pdf (visited on 09/20/2022).

[26] Q. Bramas, "The Stability and the Security of the Tangle", in *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019)*, V. Danos, M. Herlihy, M. Potop-Butucaru, J. Prat, and S. Tucci-Piergiovanni, Eds., ser. OpenAccess Series in Informatics (OASIcs), vol. 71, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, 8:1–8:15, ISBN: 978-3-95977-108-5. DOI: 10.4230/OASIcs.Tokenomics.2019.8. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2020/11972 (visited on 05/25/2021).

[27] Y. Rekhter, S. Hares, and T. Li, "A Border Gateway Protocol 4 (BGP-4)", Internet Engineering Task Force, Request for Comments RFC 4271, Jan. 2006, Num Pages: 104. DOI: 10.17487/RFC4271. [Online]. Available: https://datatracker.ietf.org/doc/rfc4271 (visited on 08/12/2022).

[28] G. Canzian, G. Mazzarella, L. Ranchail, F. Verboven, and S. Verzillo, "Evaluating the impact of price caps - evidence from the european roam-like-at-home regulation", Social Science Research Network, Rochester, NY, SSRN Scholarly Paper 3928867, Sep. 1, 2021. [Online]. Available: https://papers.ssrn.com/abstract=3928867 (visited on 06/13/2022).

[29] A. Banerjee, Y. Park, F. Clarke, H. Song, S. Yang, G. Kramer, K. Kim, and B. Mukherjee, "Wavelength-division-multiplexed passive optical network (WDM-PON) technologies for broadband access: a review [Invited]", *Journal of Optical Networking*, vol. 4, no. 11, pp. 737–758, Nov. 1, 2005, Publisher: Optica Publishing Group, ISSN: 1536-5379. DOI: 10.1364/JON.4.000737. [Online]. Available: https://opg.optica.org/jocn/abstract.cfm?uri=jon-4-11-737 (visited on 06/03/2022).

[30] K. Martiny, T. Ben Meriem, A. Buschmann, J. M. Cornily, M. Geipl, M. Mackert, K. Martiny, P. Olli, and B. Zeuner, "NGCOR Introduction, generic requirements, modeling & tooling requirements", NGMN Alliance, 2013. [Online]. Available: https://www.ngmn.org/wp-content/uploads/NGMN_Next_Generation_Converged_Operations_Requirements.pdf (visited on 09/20/2022).

[31] E. Anton, "Performance analysis of redundancy and mobility in multi-server systems", Ph.D. dissertation, Jun. 2, 2021. [Online]. Available: https://oatao.univ-toulouse.fr/28527/ (visited on 06/03/2022).

[32] S. Goyal, "Software as a service, platform as a service, infrastructure as a service – a review", *International journal of Computer Science & Network Solutions*, vol. 1, no. 3, pp. 53–67, Nov. 2013, ISSN: 2345-3397. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.9470&rep=rep1&type=pdf (visited on 08/16/2022).

[33] (2020). "5G for Industry 4.0", [Online]. Available: https://www.3gpp.org/news-events/2122-tsn_v_lan (visited on 08/12/2022).

[34] P. Kivimäki. (2020). "X-Road Implementation Models", [Online]. Available: https://www.niis.org/blog/2020/3/30/x-road-implementation-models (visited on 08/16/2022).

[35] *KSI Blockchain Timestamping — Guardtime*. [Online]. Available: https://guardtime.com/timestamping (visited on 08/16/2022).

[36] A. Haleem, A. Allen, A. Thompson, M. Nijdam, and R. Garg, "Helium : A Decentralized Wireless Network", 0.4.2, 2018. [Online]. Available: http://whitepaper.helium.com/ (visited on 09/20/2022).

[37] "Ammbr Whitepaper", Ammbr foundation, 1.1_15, Aug. 15, 2017. [Online]. Available: http://ammbr.com/docs/201708/Ammbr_Whitepaper_v1.1_15Aug2017.pdf (visited on 10/08/2018), Still available on the Wayback machine at https://web.archive.org/web/20181008145101/http://ammbr.com:80/docs/201708/Ammbr_Whitepaper_v1.1_15Aug2017.pdf.

[38] Y. Morozov, "Blockchain Telecom: Bubbletone Blockchain", Bubbletone, 2017. [Online]. Available: https://blockchaintele.com/BlockChain_Telecom_Platform_EN.pdf (visited on 02/16/2022), Still available on the Wayback machine at https://web.archive.org/web/20220216090725/https://blockchaintele.com/BlockChain_Telecom_Platform_EN.pdf.

[39] H. Zhu, C. Huang, and J. Zhou, "EdgeChain: Blockchain-based Multi-vendor Mobile Edge Application Placement", in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, Jun. 2018, pp. 222–226. DOI: 10.1109/NETSOFT.2018.8460035.

[40] *Proof-of-coverage | helium documentation*. [Online]. Available: https://docs.helium.com/blockchain/proof-of-coverage (visited on 07/20/2022).

[41] (Aug. 12, 2022). "The Wait Is Over. MOBILE Rewards Are Live.", [Online]. Available: https://blog.helium.com/the-wait-is-over-mobile-rewards-are-live-7d21cb014e22?source=collection_home---4------2---------------------- (visited on 09/16/2022).

[42] "DLT-Based Commercial and Operational Services Framework – Billing", Metro Ethernet Forum, MEF 114 draft 0.2, 2020. [Online]. Available: https://wiki.mef.net/display/LSO/DLT-based+Commercial+and+Operational+Service+Framework+-+Contributions (visited on 03/23/2021).

[43] "Smart Contracts in Permissioned Distributed Ledgers System - Architecture and Functional Specification", ETSI, DGR/PDL-004. [Online]. Available: https://portal.etsi.org/ngppapp/ContributionCreation.aspx?primarykeys=204429 (visited on 08/12/2020).

[44] *Infrastructure and Industrialization – United Nations Sustainable Development*, Publisher: United Nations. [Online]. Available: https://www.un.org/sustainabledevelopment/infrastructure-industrialization/ (visited on 09/09/2022).

[45] (Dec. 15, 2021). "Green future networks: network energy efficiency", NGMN, [Online]. Available: https://www.ngmn.org/publications/green-future-networks-network-energy-efficiency.html (visited on 06/09/2022).

[46] F. E. Salem, T. Chahed, E. Altman, A. Gati, and Z. Altman, "Optimal Policies of Advanced Sleep Modes for Energy-Efficient 5G networks", in *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, USA: IEEE, Sep. 2019, pp. 1–7, ISBN: 978-1-72812-522-0. DOI: 10.1109/NCA.2019.8935062.

[47] V. K. Quintuna Rodriguez, "New Network / IT Command: Virtualized Function Performance for a Programmable Infrastructure", Ph.D. dissertation, Sorbonne Université, 2018. [Online]. Available: https://hal.inria.fr/tel-02612498 (visited on 09/20/2022).

[48] A. Holt and S. Rivett, "Unlocking the benefits of 5g for enterprise customers", KPMG International, 2019, p. 4. [Online]. Available: https://assets.kpmg/content/dam/kpmg/xx/pdf/2019/04/unlocking-the-benefits-of-5g-for-enterprise-customers.pdf (visited on 07/04/2022).

[49] *ORAN alliance*, ORAN Alliance, Library Catalog: www.o-ran.org. [Online]. Available: https://www.o-ran.org (visited on 05/27/2020).

[50] N. Atzei, M. Bartoletti, S. Lande, and R. Zunino, "A formal model of bitcoin transactions", in *Financial Cryptography and Data Security*, S. Meiklejohn and K. Sako, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2018, pp. 541–560, ISBN: 978-3-662-58387-6. DOI: 10.1007/978-3-662-58387-6_29.

[51] B. Cohen, "The BitTorrent protocol specification", 0e08ddf84d8d3bf101cdf897fc312f2774588c9e, 2008. [Online]. Available: https://www.bittorrent.org/beps/bep_0003.html (visited on 05/05/2022).

[52] H. Zhang, Y. Wen, H. Xie, and N. Yu, *Distributed Hash Table: Theory, Platforms and Applications*, ser. SpringerBriefs in Computer Science. New York, NY: Springer New York, 2013, ISBN: 978-1-4614-9007-4. DOI: 10.1007/978-1-4614-9008-1.

[53] N. Szabo, "Bit Gold proposal", *Decentralized Business Review*, p. 21 449, 2008. [Online]. Available: https://www.debr.io/article/21449.pdf (visited on 09/20/2022).

[54] F. Brockners, S. Bhandari, S. Dara, C. Pignataro, J. Leddy, S. Youell, D. Mozes, and T. Mizrahi, "Proof of Transit", Internet Engineering Task Force, Internet-Draft draft-brockners-proof-of-transit-05, 2018. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-brockners-proof-of-transit-05 (visited on 09/20/2022).

[55] M. Rosenfeld, *Analysis of Hashrate-Based Double Spending*, Feb. 9, 2014. arXiv: 1402.2009.

[56] W. Li, M. Cao, Y. Wang, C. Tang, and F. Lin, "Mining Pool Game Model and Nash Equilibrium Analysis for PoW-Based Blockchain Networks", *IEEE Access*, vol. 8, pp. 101 049–101 060, 2020, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2997996.

[57] J. Poon and T. Dryja, "The bitcoin lightning network:" 2014. [Online]. Available: https://lightning.network/lightning-network-paper.pdf (visited on 09/20/2022).

[58] F. Vogelsteller and V. Buterin, *EIP-20: token standard*, 2015. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20 (visited on 05/18/2022).

[59] W. Entriken, S. Dieter, J. Evans, and N. Sachs, *EIP-721: non-fungible token standard*, 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721 (visited on 05/18/2022).

[60] M. Ghosh, M. Richardson, B. Ford, and R. Jansen, "A TorPath to TorCoin: Proof-of-bandwidth altcoins for compensating relays", NAVAL RESEARCH LAB WASHINGTON DC, 2014. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA621867 (visited on 09/20/2022).

[61] Y. Shahsavari, K. Zhang, and C. Talhi, "A Theoretical Model for Fork Analysis in the Bitcoin Network", in *2019 IEEE International Conference on Blockchain (Blockchain)*, Jul. 2019, pp. 237–244. DOI: 10.1109/Blockchain.2019.00038.

[62] V. Buterin, "Ethereum 2.0 Mauve paper", 2016. [Online]. Available: https://cdn.hackaday.io/files/10879465447136/Mauve%20Paper%20Vitalik.pdf (visited on 08/18/2020).

[63] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains", in *Proceedings of the Thirteenth EuroSys Conference*, ACM, 2018, p. 30. DOI: 10.1145/3190508.3190538.

[64] R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: an introduction", *R3 CEV, August*, vol. 1, no. 15, p. 14, 2016. [Online]. Available: https://docs.huihoo.com/corda/release-V2.0/_static/corda-introductory-whitepaper.pdf (visited on 09/20/2022).

[65] F. Armknecht, G. O. Karame, A. Mandal, F. Youssef, and E. Zenner, "Ripple: Overview and outlook", in *International Conference on Trust and Trustworthy Computing*, Springer, 2015, pp. 163–180. DOI: 10.1007/978-3-319-22846-4_10.

[66] D. Mazières, *The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus*, 2015. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.696.93&amp=&rep=rep1&amp=&type=pdf (visited on 09/20/2022).

[67] W. Song, W. Zhang, L. Zhai, L. Liu, J. Wang, S. Huang, and B. Li, "EOS.IO blockchain data analysis", *The Journal of Supercomputing*, vol. 78, no. 4, pp. 5974–6005, Mar. 1, 2022, ISSN: 1573-0484. DOI: 10.1007/s11227-021-04090-y.

[68] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol", 889, 2016. [Online]. Available: http://eprint.iacr.org/2016/889 (visited on 05/19/2022).

[69] *Algorand consensus - algorand developer portal*. [Online]. Available: https://developer.algorand.org/docs/get-details/algorand_consensus/ (visited on 05/19/2022).

[70] T. Rocket, M. Yin, K. Sekniqi, R. van Renesse, and E. G. Sirer, "Scalable and probabilistic leaderless BFT consensus through metastability", Aug. 24, 2020. arXiv: 1906.08936.

[71] C. LeMahieu, "Nano: A feeless distributed cryptocurrency network", *Nano*, 2018. [Online]. Available: https://content.nano.org/whitepaper/Nano_Whitepaper_en.pdf (visited on 09/20/2022).

[72] D. L. Baird, M. Harmon, and P. Madsen, "Hedera: a public hashgraph network & governing council", Hedera, Whitepaper v2.1, 2020, p. 97. [Online]. Available: https://hedera.com/hh-whitepaper (visited on 09/20/2022).

[73] M. Jay, A. Mollard, Y. Sun, R. Zheng, I. Amigo, A. Reiffers-Masson, and S. Rincón, "Utility maximisation in the coordinator-less IOTA tangle", presented at the International Symposium on Ubiquitous Networking (UNET2021), Marrakesh, Morocco: IEEE, 2021, p. 13. DOI: 10.1007/978-3-030-86356-2_8.

[74] B. Kusmierz, W. Sanders, A. Penzkofer, A. Capossele, and A. Gal, "Properties of the Tangle for Uniform Random and Random Walk Tip Selection", in *2019 IEEE International Conference on Blockchain (Blockchain)*, Jul. 2019, pp. 228–236. DOI: 10.1109/Blockchain.2019.00037.

[75] B. Kusmierz, P. Staupe, and A. Gal, "Extracting tangle properties in continuous time via large-scale simulations", p. 21, May 2018. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/4T4IAlxk9ym0eWco0UoQIQ/90094e746745b89253eb3636b4ad1597/Extracting_Tangle_Properties_in_Continuous_Time_via_Large_Scale_Simulations_V2.pdf (visited on 09/20/2022).

[76] B. Kusmierz, "The first glance at the simulation of the tangle: discrete model", p. 10, 2017. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2ZO5XxwehymSMsgusUE6YG/f15f4571500a64b7741963df5312c7e7/The_First_Glance_of_the_Simulation_Tangle_-_Discrete_Model_v0.1.pdf (visited on 09/20/2022).

[77] S. Popov, O. Saa, and P. Finardi, "Equilibria in the tangle", *Computers and Industrial Engineering*, vol. 136, pp. 160–172, C Oct. 1, 2019, ISSN: 0360-8352. DOI: 10.1016/j.cie.2019.07.025.

[78] R. Matzutt, B. Kalde, J. Pennekamp, A. Drichel, M. Henze, and K. Wehrle, "How to Securely Prune Bitcoin's Blockchain", in *2020 IFIP Networking Conference (Networking)*, Jun. 2020, pp. 298–306, ISBN: 978-3-903176-28-7.

[79] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger", 2014. [Online]. Available: http://gavwood.com/Paper.pdf (visited on 09/20/2022).

[80] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router", Naval Research Lab Washington DC, 2004. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA465464 (visited on 09/20/2022).

[81] D. Chen, Z. Zhang, A. Krishnan, and B. Krishnamachari, "PayFlow: Micropayments for Bandwidth Reservations in Software Defined Networks", presented at the Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2019, pp. 26–31. DOI: 10.1109/INFCOMW.2019.8845319.

[82] J. Wannstom, "LTE-Advanced", 3GPP, 2013. [Online]. Available: https://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced (visited on 05/19/2020).

[83] *Google fi - a different kind of phone plan*, Meet Google Fi, a different kind of phone plan. Library Catalog: fi.google.com. [Online]. Available: https://fi.google.com/about/ (visited on 05/19/2020).

[84] *Hyperledger sawtooth*, Hyperledger. [Online]. Available: https://www.hyperledger.org/use/sawtooth (visited on 01/14/2021).

[85] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, "On security analysis of proof-of-elapsed-time (poet)", in *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, Springer, 2017, pp. 282–297. DOI: 10.1007/978-3-319-69084-1_19.

[86] *World Wi-Fi – decentralized free Wi-Fi network powered by blockchain.* [Online]. Available: https://en.worldwifi.io/ (visited on 05/19/2020).

[87] 3GPP TS 28.552 V16.04.0, *Management and orchestration; Concepts, use cases and requirements (Release 15)*. 2019. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3413 (visited on 05/06/2020).

[88] 3GPP TS 28.554 V16.03.0, *Management and orchestration; Concepts, use cases and requirements (Release 15)*. 2019. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3415 (visited on 05/06/2020).

[89] V. Quintuna Rodriguez and F. Guillemin, "Cloud-RAN Modeling Based on Parallel Processing", *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 457–468, Mar. 2018, Conference Name: IEEE Journal on Selected Areas in Communications, ISSN: 1558-0008. DOI: 10.1109/JSAC.2018.2815378.

[90] V. K. Quintuna Rodriguez and F. Guillemin, "Higher aggregation of gNodeBs in Cloud-RAN architectures via parallel computing", in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Feb. 2019, pp. 151–158. DOI: 10.1109/ICIN.2019.8685900.

[91] "CSP use cases utilizing blockchain", TMForum, TM Forum Technical Repor TR279, Aug. 8, 2019. [Online]. Available: https://www.tmforum.org/resources/technical-report/tr279-csp-use-cases-utilizing-blockchain-v3-1/ (visited on 01/21/2021).

[92] A.-L. Vion, "Software asset management and cloud computing", Ph.D. dissertation, Université Grenoble Alpes, Mar. 29, 2018. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01901991 (visited on 09/20/2022).

[93] S. Steffen, B. Bichsel, R. Baumgartner, and M. Vechev, "ZeeStar: Private Smart Contracts by Homomorphic Encryption and Zero-knowledge Proofs", presented at the 2022 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, Apr. 20, 2022, pp. 1543–1543, ISBN: 978-1-66541-316-9. DOI: 10.1109/SP46214.2022.00114. [Online]. Available: https://www.computer.org/csdl/proceedings-article/sp/2022/131600b543/1CIO7PhuocU (visited on 07/21/2022).

[94] *5g-ACIA*, ZVEI, Library Catalog: www.5g-acia.org. [Online]. Available: https://www.5g-acia.org/ (visited on 05/27/2020).

[95] *Blockchain unleashed - TM forum*, TM Forum, Library Catalog: www.tmforum.org. [Online]. Available: https://www.tmforum.org/catalysts/blockchain-unleashed/ (visited on 05/27/2020).

[96] (Oct. 7, 2020). "Observatoire ANFR : près de 48 500 sites 4G en service en France au 1er octobre", [Online]. Available: https://www.anfr.fr/toutes-les-actualites/actualites/observatoire-anfr-pres-de-48-500-sites-4g-en-service-en-france-au-1er-octobre/ (visited on 10/13/2020).

[97] X. Ling, J. Wang, T. Bouchoucha, B. C. Levy, and Z. Ding, "Blockchain Radio Access Network (B-RAN): Towards Decentralized Secure Radio Access Paradigm", *IEEE Access*, vol. 7, pp. 9714–9723, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2890557.

[98] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, "When Internet of Things Meets Blockchain: Challenges in Distributed Consensus", *IEEE Network*, vol. 33, no. 6, pp. 133–139, Nov. 2019, ISSN: 0890-8044, 1558-156X. DOI: 10.1109/MNET.2019.1900002. arXiv: 1905.06022.

[99] D. Stucchi, R. Susella, P. Fragneto, and B. Rossi, "Secure and Effective Implementation of an IOTA Light Node using STM32", in *Proceedings of the 2nd Workshop on Blockchain-enabled Networked Sensor*, ser. BlockSys'19, New York, NY, USA: Association for Computing Machinery, Nov. 10, 2019, pp. 28–29, ISBN: 978-1-4503-7012-7. DOI: 10.1145/3362744.3363344.

Appendix A

Making Mobile Network Operators cooperate thanks to Multi-Agent Reinforcement Learning

Contents

A.1 Presentation of the proposed model

The problem is first modelled as a social dilemma, about the will for MNOs to remain *on-guard* and serve every users during low-activity periods. Preliminary modellings show that without cooperation model, the system leads to a prisoner's dilemma, as MNOs won't find personal interest in cooperating. A MARL environment is then set up to model MNOs' interactions. On the environment, the MNOs are represented by *agents*, interacting with each other through *negotiations*, and each negotiation lasts multiple *timesteps*. A decentralised partially observable Markov decision process is then used to model agents' negotiations. On each negotiation, the agents try and decide by themselves This process allocates individual rewards to each agent at each negotiation. For each negotiation, the rewards are defined as follows:

* -1.00 when an agent negotiated to be *on-guard*;
* -0.90 when an agent failed at finding an *on-guard* agent;
* -0.01 when an agent succeeded at finding an *on-guard* agent;

* -0.01 for each time step to incite agents to negotiate quickly (with as few timesteps as possible).

Each agent i tries and learns to find a policy $\hat{\pi}_i$ in order to maximise its reward.

A.2 Simulation & Results

The proposed model is then simulated on a MARL environment, in the three modes discussed before (Free, Recommanded and Imposed), and with different amount of agents (MNOs): $N = 3, 4, 8, 10$. After a training step, each agent i then behaves according to its learned policy $\hat{\pi}_i$, that is compared with an optimal cooperating policy $\pi_{i,C}$ and a worst cooperating policy $\pi_{i,D}$. Let $G(\hat{\pi}_i)$ be the expected return, that is to say the sum of all rewards received by the agent i. Following values are then extracted :

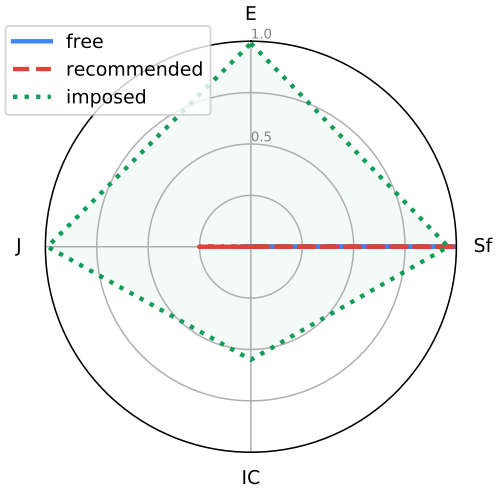
* *Efficiency* E represents how close to the optimum the social welfare is, that is to say how close MNOs are for perfect cooperation. This metric is calculated by dividing the sums of every *rewards* received by each actor with the best case scenario.
* *Safety* $Sf(i)$ measures the risk taken by any agent i, and is defined as the difference, if all other agents $j \neq i$ do not cooperate, between the expected return received when the agent i cooperate and when he isn't.
* *Incentive-compatitivity* IC measures the capacity to incentivise cooperation. It is defined as the difference for a given agent i, if all other agents cooperate, between the expected return received when the agent i cooperate and when he isn't.
* *Fairness* J represents the total number of KWh saved after negotiation.

The following metrics are displayed on radar chart, on Figure A.1.

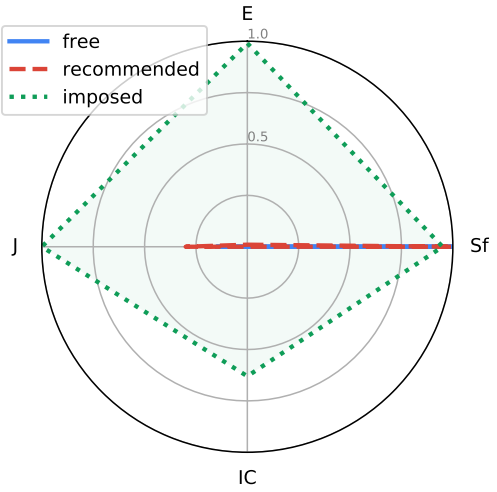
In scenarii with *free* modes, efficiency remained close to 0.0. The IC Value also remained close to 0, meaning that any MNO trying to cooperate with other cooperating MNOs did not get a good return. Yes the Safety value remained close to 1 in most cases, meaning that MNOs did not try to cooperate. These resulted to low fairness scores.

On the other hand, on all scenarii with *imposed* mode Efficiency is almost at its maximum (1.0). This means that the policy adopted by MNOs is close to optimum, and that energy is actually saved. This can be futher seen with the IC value : if all but one MNO cooperate, the more agents they are, the bigger the regret for the defecting MNO. Furthermore, these scenarii showed as well good Safety and Fairness scores. This means that control from a regulator can be beneficial for all agents, given convincing rewards for the agents.

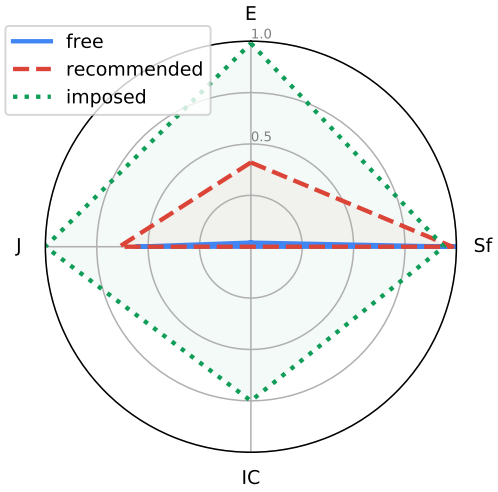
Results in *recommanded* mode are more mitigated. While agents did not cooperate well with $N = 3$ and $N = 4$ agents, results observed showed that the more agents, the more willing to collaborate they become.



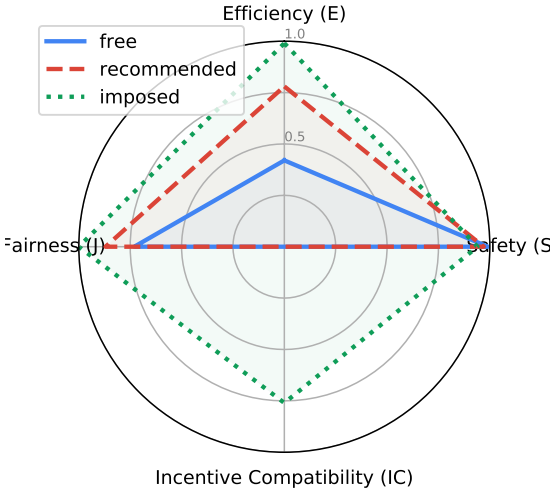
(a) 3-agent game



(b) 4-agent game



(c) 8-agent game



(d) 10-agent game

Figure A.1 – Radar chart of the cooperation metrics

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Mécanismes de confiance pour l'assurance de services de connectivités sur infrastructures multi-acteurs

Mot clés : Confiance, Blockchain, Registres Distribués, Graphes Acycliques Orientés, Réseaux Collaboratifs

Résumé : L'évolution des réseaux de communication vers des infrastructures virtualisées a permis l'émergence de nouvelles architectures de réseaux. En effet, les équipements réseaux qui étaient jusqu'alors complètement physiques, et dédiés à une fonction particulière peuvent être maintenant exploités sous la forme de briques logicielles, elles-mêmes déployées sur des équipements physiques banalisés. Cela permet maintenant de déployer et gérer une infrastructure réseau virtuelle de bout en bout complète de manière purement automatique. De plus, l'infrastructure physique peut être également partagée entre plusieurs réseaux virtuels, grâce au principe de l'architecture "cloud" pour mutualiser les fonctions de réseau virtuelles, et des techniques comme le multiplexage par longueur d'onde pour les liens optiques.

Cela permet donc l'émergence de nouveaux modèles collaboratifs et de places de marché numériques permettant aux opérateurs de créer des services réseaux sans en posséder l'infrastructure physique, ou de partager la leur. Des technologies comme les Registres Distribués peuvent permettre de consolider les échanges entre opérateurs sur de telles places de marché, grâce à des données prouvables et facilement auditables stockées sur des systèmes distribués. Ainsi, des réseaux collaboratifs complètement décentralisés et automatisés sont maintenant possibles sur des infrastructures virtuelles.

Cependant, de tels réseaux nécessitent une visibilité sur l'exploitation du réseau, pour assurer la qualité du service rendu. Pour cela, des indicateurs de performance de confiance doivent être collectés sur l'infrastructure réseau partagée. Sécuriser et consolider de tels indicateurs est difficile de manière traditionnelle, car cela nécessite une architecture complexe, et des intermédiaires de confiance.

Cette thèse propose une *"data layer"*, ou "couche de données" pour permettre la production de tels indicateurs de performance d'une manière décentralisée. L'utilisation de la technologie du Tangle, une technologie de registre distribué novatrice et conçue pour être performante et efficace est considérée pour permettre la création, sécurisation et conservation des indicateurs de performance. Plusieurs contributions sont explorées dans cette thèse pour évaluer et valider l'architecture proposée.

Tout d'abord, plusieurs scénarii de partage de ressources entre opérateurs de télécommunication sont explorés. Un simulateur de la structure du Tangle s'inspirant des modèles existant dans la littérature est également proposé pour évaluer la pertinence de la technologie pour le cas d'usage proposé, par rapport à ses besoins. Des méthodes permettant la gestion de tâches basiques attrayant à la collaboration comme la sécurisation du transport de la donnée, la synchronisation d'horloge pour assurer l'horodatage des données, et l'archivage sécurisé de données sont également explorées. Une contribution permettant à des acteurs distincts impliqués dans un lien réseau collaboratif de partager des informations d'usage à l'aide d'un mécanisme de consensus et d'un registre distribué est également explorée. Cette contribution permet ainsi la création d'indicateurs de confiance. Finalement, une étude plus détaillée de la "data layer" proposée plus haut et de ses différents composants est proposée, prenant en considération les réseaux mobiles.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Title: Trust mechanisms for connectivity service assurance on multi-actor infrastructures

Keywords: Trust, Blockchain, Distributed Ledgers, Directed Acyclic Graphs, Cooperative Networks

Abstract: The evolution of telecommunication networks toward virtualised infrastructures has enabled new architectures to emerge. Indeed, then fully physical, dedicated network equipments can be now virtualised into pieces of software, deployable on commodity hardwares. As a result, a full end-to-end virtualised network infrastructure can be deployed on the fly, and its lifecycle fully automated. Furthermore, sharing principles can apply thanks to cloud principles on virtualised network functions, or features like Wavelength Division Multiplexing (WDM) on optical links.

This thus enables new collaborative models to emerge as marketplaces can help telcos to build services without owning the physical infrastructure, or by sharing theirs. Truthfulness technologies like the Distributed Ledger Technology (DLT) can strengthen said exchanges thanks to provable, easily auditable data stored on distributed systems. As a result, fully decentralised and fully automated collaborative networks are now possible on virtualised infrastructures.

However, such systems require trusted knowledge on the network operation as to ensure the quality of the delivered service. For that purpose, trusted Key Performance Indicators need to be collected on the shared infrastructure. Securing such data is a burden in a traditional way, as it requires a complex framework and trusted third parties like Data Clearing Houses.

This thesis proposes a “data layer” as an alternative approach to provide trusted knowledge about a shared network infrastructure to a collaborative framework. More precisely, the use of the Tangle, a novel DLT primarily designed for efficiency is considered to process, hold and secure performance indicators in a decentralised way. Then various contributions are explored in this thesis to validate and assess the proposed framework.

First, various efforts are devoted in exploring collaborative resource sharing scenarii and opportunities for telcos. Simulations of the Tangle structure based on state-of-the-art models are also explored to evaluate this DLT pertinence for the proposed framework, in regard of the performance requirements. Novel methods to handle basic tasks such as data transport, clock synchronisation and data archiving are also presented, to improve the operation of such a multi-actor environment. A contribution allowing actors involved in a network path to share information about the usage of the path, using a consensual mechanism and a DLT is also explored to generate trusted indicators on a multi-actor environment. Finally, a more detailed study of the proposed data layer and of its different components is proposed, taking mobile networks as a use-case.