A novel blockchain-based architecture for mobile network operators : Beyond 5G

Fariba Ghaffari

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAS009

Thèse de doctorat

A novel Blockchain-based Architecture for Mobile Network Operators: Beyond 5G

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

École doctorale n°626 Ecole Doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à CNAM, Paris, le 13/10/2023, par

FARIBA GHAFFARI

Composition du Jury :

Axel Kupper Service-centric Networking, TU Berlin, Berlin, 10587 Germany	Rapporteur
Mika Ylianttila Centre for Wireless Communications, University of Oulu, Finland	Rapporteur
Emmanuelle Anceaume CNRS, University of Rennes, Irisa, France	Examinateur
Abdelkader Lahmadi Universite de Lorraine, CNRS, Inria, LORIA, Nancy, France	Examinateur
Cigdem Sengul Brunel University London, Uxbridge, UK	Examinateur
Noel Crespi SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, France	Directeur de thèse
Emmanuel Bertin Orange Innovation, Caen, France	Co-directeur de thèse

Doctor of Philosophy (PhD) Thesis
Institut-Mines Télécom, Télécom SudParis & Institut Polytechnique de Paris (IP Paris)

Specialization

Computer Science

presented by

Fariba GHAFFARI

A novel Blockchain-based Architecture for Mobile Network Operators: Beyond 5G

Committee:

Axel Kupper	Reviewer	Service-centric Networking, TU Berlin, Berlin, 10587 Germany
Mika Ylianttila	Reviewer	Centre for Wireless Communications, University of Oulu, Finland
Emmanuelle Anceaume	Examiner	CNRS, University of Rennes, Irisa, France
Abdelkader Lahmadi	Examiner	Universite de Lorraine, CNRS, Inria, LORIA, Nancy, France
Cigdem Sengul	Examiner	Brunel University London, Uxbridge, UK
Noel Crespi	Supervisor	SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, France
Emmanuel Bertin	Advisor	Orange Innovation, Caen, France

Thèse de Doctorat (PhD) de Institut-Mines Télécom, Télécom SudParis et l'Institut Polytechnique de Paris (IP Paris)

Spécialité

Informatique

présentée par

Fariba GHAFFARI

Une nouvelle architecture basée sur la Blockchain pour les opérateurs de réseaux mobiles : au-delà de la 5G

Jury composé de :

Axel Kupper	Rapporteur	Service-centric Networking, TU Berlin, Berlin, 10587 Germany
Mika Ylianttila	Rapporteur	Centre for Wireless Communications, University of Oulu, Finland
Emmanuelle Anceaume	Examinateur	CNRS, University of Rennes, Irisa, France
Abdelkader Lahmadi	Examinateur	Universite de Lorraine, CNRS, Inria, LORIA, Nancy, France
Cigdem Sengul	Examinateur	Brunel University London, Uxbridge, UK
Noel Crespi	Supervisor	SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, France
Emmanuel Bertin	Advisor	Orange Innovation, Caen, France

Dedication

To
all people who accepted to live in the discomfort of unknowing,
rather than the comfort zone of believing without thinking.
and ; To
all my Iranian friends who choose to fight for freedom,
to sacrifice their future for the prosperity of our homeland,
who have big hearts regardless of their youthfulness,
and to those who could not defend their thesis...

Acknowledgements

My Mom ;
Thank you for your unrequited love that I feel in the deepest part of my soul. This love empowered me to break my limits and experience life freely and fearlessly.
My Dad ;
Thank you for all your unrequited love that I used to feel in the deepest part of my soul which was my wings to explore this new life. I started this journey in the shadows of your endless support ; Although I lost you in the middle of this career, from the deep of my heart I believe you would have been the happiest person seeing me now.
My true friend Amin ;
When it comes to goodness, you are the first to come to my mind. You are proof of true love for me. I am very lucky to be your sister.
Prof. Emmanuel Bertin ;
The lessons I learned from you brightened my way and helped me to overcome all the obstacles. Thank you for your confidence and mentorship over the years and for sharing your extensive knowledge with me.
Prof. Noel Crespi ;
Thank you for believing in me at the beginning of this journey and for providing me with valuable opportunities. Being a part of your team, was an honor for me.
Dear Marc Mazoué ;
Words cannot even express my thanks for your support and friendship. I wish life to unfold you beautiful and valuable people just like you.
My best friends ;
Shanay and Nischal ; my best friends who I could call "family". Thank you for all your caring, patience, and support. I am grateful from the deep of my heart to have you in my life.
My dear little family in TSP ; Amir, Reza, and Meroua thank you for your help, humor, and support. I am grateful for each one of you.

Abstract

Mobile Network Operators (MNOs) provide connectivity to billions of users at all times, based on centralized architectures, some of whose founding principles were designed as standardization in the 1980s and 1990s. This has impacts on mutualization costs (e.g., for sharing the infrastructures, such as RAN, that can be used by several MNOs and providers), as well as on energy consumption and environmental impact (i.e., because of deploying infrastructures in stand-alone mode rather than sharing). This also complicates the implementation of more collaborative business models with other providers (of services and content), or even with business customers (for example, for private 5G). Moreover, the complexity of 5G and beyond 5G networks may surpass the capability of one MNO to manage the cost and the complexity of connection for a huge number of interconnected elements. Finally, due to their centralization, existing MNO architectures can be subject to technical risks and vulnerabilities. For example, single points of failure can impact availability, and storing user data in a centralized database increases the risk of data leakage or loss. Although the current systems are functional and efficient, a study of alternative architecture principles, based on the achievements of distributed systems, seems important to carry out in the perspective of after 5G and 6G. This is the subject of this doctoral work.

Addressing these challenges is not a straightforward journey. MNO architectures have been defined for nearly three decades by well-established standards organizations. However, we believe that there is an interest in proposing within the telecommunications research community a new approach, starting from needs and not from existing architectures. First, we propose a comprehensive study of the challenges existing in current cellular networks regarding the commercial and collaboration aspects between actors, as well as the technical and security issues. The results of this study led us to propose two main contributions. Our first contribution concerns the cooperation between the different actors of the cellular network ecosystem (i.e. MNOs, service or content providers, enterprises, vendors and end users). Our second contribution concerns the collaboration between MNOs (and possibly with regulatory authorities) for the management of identities and profiles. It is important to mention that the field of validity of this work is limited to actors wishing to collaborate on the market of cellular networks, after 5G and in the perspective of 6G, while maintaining their independence regarding their services. and operations.

Such alternative solutions must, at least, be based on a distributed/decentralized ecosystem, ensure trust between actors, natively allow the sharing of resources between stakeholders (with the associated retribution mechanisms), provide higher automation (especially for processes involving several actors), facilitating market competitiveness and providing sufficient security and confidentiality. Any alternative meeting these requirements would be a promising potential solution. In this doctoral work, we focus on Blockchain tech-

nology, which structurally allows us to address these challenges. Blockchain is indeed a peer-to-peer, cryptographically secure, add-only, immutable, traceable, and transparent distributed ledger technology that can only be updated by consensus among the majority of participating nodes on the network. Due to their intrinsic characteristics, such as distributed nature, immutability, transparency, traceability, and non-repudiation, Blockchain, and smart contracts can reduce the cost, latency, and complexity of collaboration between entities in multi-actor cellular network systems while increasing the reliability, traceability, and monitoring capabilities in the collaborative network.

More precisely, the first contribution proposes a new hybrid telecommunications ecosystem (distributed-decentralized) based on Blockchain for the core network of MNOs as a starting proposal to shape the design of the network beyond 5G and 6G. This method offers the possibility of eliminating any central authority (while integrating regulatory bodies), increasing system fault tolerance, simplifying IT procedures and securing payment between entities. The proposed system covers the main functions of MNOs such as user subscription and profile management, authentication and key management procedure, access control, user registration in the network (registration procedure), mobility management, session management, and billing.

The second contribution introduces a new management of user profiles and a porting of numbers and profiles between MNOs using Blockchain technology and smart contracts. This method aims to eliminate central authority in the porting process by creating a more collaborative and distributed system, increasing automation and trust, and addressing the noticeable delays of existing porting methods between MNOs. It offers the possibility of porting user profiles to the recipient MNO as well as the telephone number with an automated procedure without centralized authority or third parties.

To evaluate the proposed system and analyze its implementation feasibility, we proposed three deployment scenarios in which the Blockchain can be logically positioned either in RAN, core network, or service layer. We implemented the third solution (i.e., positioning the Blockchain at the service level) and connected the core network of the private cellular network to the Blockchain. However, some parts of the evaluation (e.g., authentication and key agreement, handover) are implemented and evaluated only in Blockchain and using virtual users. The evaluation results show that the system is scalable enough regarding the number of actors and collaborators, and based on the network requirements, its performance and security level are adjustable. Moreover, security analysis shows that the system is resilient against common threads for communication (i.e., mostly focused on authentication and access control which are crucial parts of all other MNO procedures). Finally, the obstacles and limitations of real-world implementation of the novel architecture regarding latency, scalability, standardization, storage requirements, and incentives for different parties are discussed.

We hope that these contributions can serve as a basis for discussion within the telecom community for the definition of new architectures for cellular networks.

Keywords

Mobile Network Operators, Multi-actor connectivity system, Cellular network, Core network, Blockchain, Distributed architecture, Competitive market, Business model, Scalability, Security.

Résumé

Les Opérateurs de Réseaux Mobiles (ORM) fournissent à chaque instant de la connectivité à des milliards d'utilisateurs, en se basant sur des architectures centralisées dont certains des principes fondateurs ont été conçus en standardisation dans les 1980 et 1990.

Cela a des impacts sur les coûts de mutualisation (par exemple, en complexifiant le partage des infrastructures actives, comme le RAN), ainsi que sur la consommation d'énergie et l'impact environnemental (en raison du déploiement des infrastructures actives de façon autonome plutôt que partagée). Cela complexifie également la mise en place de modèles d'affaire plus collaboratifs avec d'autres fournisseurs (de services comme de contenus), ou même avec des clients entreprise (par exemple, pour la 5G privée). De plus, la complexité des réseaux 5G et au-delà de la 5G peut dépasser la capacité d'un ORM à gérer le coût et la complexité de la connexion pour un grand nombre d'éléments interconnectés. Enfin, de par leur centralisation, les architectures ORM existantes peuvent être sujettes à des risques techniques et à des vulnérabilités. Par exemple, les points de défaillance uniques peuvent impacter la disponibilité, et le stockage des données de l'utilisateur dans une base de données centralisée augmente le risque de fuite ou de perte de données. Bien que les systèmes actuels soient fonctionnels et performants, une étude de principes d'architecture alternatifs, basés sur les acquis des systèmes distribués, semble importante à réaliser dans la perspective de l'après 5G est de la 6G. C'est l'objet de ce travail doctoral.

Relever ces défis n'est pas simple. Les architectures ORM sont définies depuis près de trois décennies par des organismes de normalisation bien établis. Cependant, nous croyons qu'il y a un intérêt à proposer au sein de la communauté de la recherche en télécommunications une approche nouvelle, en repartant des besoins et non des architectures existantes.

Dans un premier temps, nous proposons une étude complète des défis existants dans les réseaux cellulaires actuels concernant les aspects commerciaux et de collaboration entre acteurs, ainsi bien sûr que les problèmes techniques et de sécurité. Les résultats de cette étude nous ont amenés à proposer deux contributions principales. Notre première contribution concerne la coopération entre les différents acteurs de l'écosystème du réseau cellulaire (c'est-à-dire les ORM, les fournisseurs de services ou de contenus, les entreprises, les vendeurs et les utilisateurs finaux). Notre seconde contribution concerne la collaboration entre les ORM (et éventuellement avec les instances de régulation) pour la gestion des identités et des profils. Il est important de mentionner que le domaine de validité de ce travail est limité aux acteurs souhaitant collaborer sur le marché des réseaux cellulaires, après la 5G et dans la perspective de la 6G, tout en gardant leur indépendance vis-à-vis de leurs services et opérations.

De telles solutions alternatives doivent, pour le moins, reposer sur un écosystème distribué/décentralisé, assurer la confiance entre les acteurs, permettre nativement le partage de ressources entre les parties prenantes (avec les mécanismes de rétribution associés), ap-

porter une automatisation plus élevée (notamment pour les processus impliquant plusieurs acteurs), faciliter la compétitivité du marché et fournir une sécurité et une confidentialité suffisantes. Toute alternative répondant à ces exigences serait une solution potentielle prometteuse. Dans ce travail doctoral, nous nous concentrons sur la technologie Blockchain, qui permet structurellement d'adresser ces défis. La blockchain est en effet une technologie de registre distribué peer-to-peer, cryptographiquement sécurisée, à ajout uniquement, immuable, traçable et transparente qui ne peut être mise à jour que par consensus entre la majorité des nœuds participants sur le réseau. En raison de ses caractéristiques intrinsèques, telles que la nature distribuée, l'immuabilité, la transparence, la traçabilité et la non-répudiation, la blockchain et les smart contracts peuvent réduire le coût, la latence et la complexité de la collaboration entre les entités dans les systèmes de réseaux cellulaires multi-acteurs tout en augmentant le fiabilité, traçabilité et capacités de supervision dans le réseau collaboratif.

Plus précisément, la première contribution propose un nouvel écosystème de télécommunications hybride (distribué-décentralisé) basé sur une architecture Blockchain pour le réseau central des ORM en tant que proposition de départ pour fonder la conception du réseau au-delà de la 5G et de la 6G. Cette méthode offre la possibilité d'éliminer toute autorité centrale (tout en intégrant les organismes de réglementation), d'augmenter la tolérance aux pannes du système, de simplifier les procédures informatiques et de sécuriser le paiement entre les entités. Le système proposé couvre les principales fonctions des ORM telles que l'abonnement des utilisateurs et la gestion des profils, la procédure d'authentification et de gestion des clés, le contrôle d'accès, l'enregistrement des utilisateurs dans le réseau (procédure d'enregistrement initial et périodique), la gestion de la mobilité, la gestion des sessions et la facturation.

La seconde contribution introduit une nouvelle gestion des profils utilisateurs et un portage des numéros et des profils entre ORM à l'aide de la technologie Blockchain et des smart contracts. Cette méthode vise à éliminer l'autorité centrale dans la procédure de portage en créant un système plus collaboratif et distribué, à augmenter l'automatisation et la confiance, et à remédier aux délais notables des méthodes existantes de portage entre ORM. Elle offre la possibilité de porter les profils des utilisateurs vers l'ORM destinataire ainsi que le numéro de téléphone avec une procédure automatisée sans autorité centralisée ni tiers.

Pour évaluer les systèmes proposés et analyser leur faisabilité de mise en œuvre, nous avons proposé trois scénarios de déploiement dans lesquels la Blockchain peut être logiquement positionnée soit en RAN, cœur de réseau, ou couche de service. Nous avons implémenté la troisième solution (c'est-à-dire positionner la Blockchain au niveau du service) et connecté un cœur de réseau cellulaire Open Source à la Blockchain. Cependant, certaines parties de l'évaluation (par exemple, l'authentification et l'accord de clé, le transfert) sont mises en œuvre et évaluées uniquement dans Blockchain et à l'aide d'utilisateurs virtuels, à cause de modifications qui devraient impacter les terminaux et les profils SIM. Les résultats de l'évaluation montrent que le système est suffisamment évolutif en ce qui

concerne le nombre d'acteurs et de collaborateurs, et en fonction des exigences du réseau, ses performances et son niveau de sécurité sont ajustables. De plus, l'analyse de la sécurité montre que le système est résilient face aux menaces les plus communes sur les communications (notamment au niveau de l'authentification et du contrôle d'accès, qui sont des éléments cruciaux pour toutes les autres procédures des ORM). Enfin, les obstacles et les limites d'une mise en œuvre dans le monde réel sont discutés, notamment concernant la latence, l'évolutivité, la normalisation, les exigences de stockage et les incitations pour les différentes parties.

Nous espérons que ces contributions pourront servir de base de discussion au sein de la communauté télécom pour la définition de nouvelles architectures pour les réseaux cellulaires.

Mots-clés

Opérateurs de réseaux mobiles, Réseau cellulaire, Réseau central, Blockchain, Architecture distribuée, Marché concurrentiel, Modèle économique, Évolutivité, Sécurité.

Publications

Journal Papers

J.1. F. Ghaffari, K. Gilani, E. Bertin, N. Crespi, "Identity and access management using distributed ledger technology: A survey," *International Journal of Network Management.* 2022; 32(2): e2180. https://doi.org/10.1002/nem.2180 - *Published*

J.2. F. Ghaffari, E. Bertin, N. Crespi, S. Behrad and J. Hatin, "A Novel Access Control Method Via Smart Contracts for Internet-Based Service Provisioning," *in IEEE Access*, vol. 9, pp. 81253-81273, 2021, doi: 10.1109/ACCESS.2021.3085831. - *Published*

J.3. F. Ghaffari, E. Bertin, N. Crespi, "A novel Blockchain-based Architecture for Mobile Network Operators: Beyond 5G", *ACM Transactions on Distributed Ledger Technologies: Research and Practice (Special Issue on Distributed Ledger Technology (DLT) for Beyond 5G Systems)*, submitted in September 2022- *Under review*

J.4. F. Ghaffari, E. Bertin, and N. Crespi. "Designing Mobile Network Operators for Beyond 5G and 6G: A novel Blockchain-based Core Network Architecture," submitted to *IEEE Network magazine*- submitted in June 2023, *under review.*

J.5. F. Ghaffari, N. Aryal, E. Bertin, N. Crespi, and J. Garcia-Alfaro. "Widening Blockchain technologies toward mobile access control," submitted to *Sensors*- submitted in February 2023, *Published.*

J.6. F. Ghaffari, E. Bertin, and N. Crespi. "Distributed Ledger Technologies for Authentication and Access Control in Networking Applications: A Comprehensive Survey," submitted to *Computer Science Review- published.*

Conference Papers

C.1. F. Ghaffari, E. Bertin, N. Crespi, "Blockchain-based User Profile and Mobile Number Portability for Beyond 5G Mobile Communication Networks", *In 2022 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)- published*

C.2. N. Aryal, F. Ghaffari, S. Rezaei, E. Bertin, N. Crespi, "Private Cellular Network Deployment: Comparison of OpenAirInterface with Magma Core", *18th International Conference on Network and Service Management*, 2022- *published*

C.3. K. Gilani, F. Ghaffari, E. Bertin and N. Crespi, "Self-sovereign Identity Management Framework using Smart Contracts," *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 2022, pp. 1-7, doi: 10.1109/NOMS54207.2022.9789831 - *Published.*

C.4. F. Ghaffari, E. Bertin, and N. Crespi. "A novel approach for network resource sharing via blockchain." *In Proceedings of the SIGCOMM'21 Poster and Demo Sessions*, pp. 50-52. 2021 - *Published.*

C.5. F. Ghaffari, E. Bertin, J. Hatin, and N. Crespi, "Authentication and Access Control based on Distributed Ledger Technology: A survey," *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, 2020, pp. 79-86, doi: 10.1109/BRAINS49436.2020.9223297 - *Published*

C.6. F. Ghaffari, E. Bertin, and N. Crespi. "User Profile and Mobile Number Portability for Beyond 5G: Blockchain-based Solution," submitted to *26th Conference on Innovation in Clouds, Internet and Networks 2023- published.*

C.7. N. Aryal, F. Ghaffari, E. Bertin, N. Crespi, "Moving towards Open Radio Access Networks using Distributed Ledger Technology", *Submitted to BRAINS 2023- accepted*

C.8. N. Aryal, F. Ghaffari, E. Bertin, N. Crespi, "Subscription Management for Beyond 5G and 6G Cellular Networks using Blockchain Technology", *Submitted to CNSM 2023- accepted*

Book Chapters

B.1. N. Aryal, F. Ghaffari, E. Bertin, N. Crespi, "Moving Towards Next Generation Networks with Distributed Ledger Technologies: SWOT Analysis", *Submitted to Blockchain and Smart-Contract Technologies for Innovative Applications published by Springer- Accepted*

Table of contents

Abbreviations

RAN Radio Access Network

DN Data Network

3GPP 3rd Generation Partnership Project

AMF Access and Mobility Management Function

SMF Session Management Function

SBA Service-based Architecture

UPF User Plane Function

PCF Policy Control Function

UDM Unified Data Management

PDU Protocol Data Unit

AUSF Authentication Server Function

MNO Mobile Network Operator

SSL/TLS Secure Socket Layer/Transport Layer Security

UDR Unified Data Repository

SUPI Subscription Permanent Identifier

PII Personally identifiable information

SUCI Subscription Concealed Identifier

UE User Equipment

ECIES Elliptic Curve Integrated Encryption Scheme

NEF Network Exposure Function

QoS Quality of Service

RRC Radio Resource Control

AAC Authentication and Access Control

RFID Radio Frequency Identification

USB Universal Synchronous Bus

DAC Discretionary Access Control

CapBAC Capability-based Access Control

MAC Mandatory Access Control

RBAC Role-Based Access Control

ABAC Attribute-Based Access Control

DoS Denial-of-Service

DDoS Distributed Denial-of-Service

IDS Intrusion Detection System

IPS Intrusion Prevention Systems

MitM Man-in-the-Middle

LoRaWAN long-range wide-area network

IoT Internet of Things

WSN Wireless Sensor Networks

AKA Authentication and Key Agreement

ICN Information-Centric Networking

BFT Byzantine Fault Tolerance

NFV Network Function Virtualization

PII Personally Identifiable Information

µO Micro Operator

MVNO Mobile Virtual Network Operator

MNO Mobile Network Operator

TTP Trusted Third Parties

SDN Software-Defined Networking

BS Base Station

DHT Distributed Hash Table

P2P Peer-to-Peer

IPFS InterPlanetary File System

HTTPS Hypertext Transfer Protocol Secure

MNP Mobile Number Porting

EVM Ethereum Virtual Machine

Chapter 1

Introduction

Contents

1.1 Introduction

Since August 2022, nearly 60% of web traffic all over the world has come from mobile devices, which are used by more than 6 billion unique mobile subscribers (in cellular networks) globally [1–3]. This statistic emphasizes the vital role played by Mobile Network Operators (MNO) in providing services, connection, and content to the ever-expanding Internet and cellular network users [4, 5]. Next-generation networks Beyond 5G (B5G) and the 6^{th} Generation of cellular networks (6G) attempt to provide numerous new opportunities via integrating different technologies and services to provide broader connectivity, deliver seamless mobility at higher speeds, enhance security, etc. [6, 7]. Due to the significance of these goals for the next generation of networks and providing interconnection among different technologies in the new/next generations of cellular networks, it is essential to introduce a multi-actor ecosystem in which different entities and actors can collaborate effectively, efficiently, and trustfully to provide more collaborative, competitive and innovative market in cellular networks. Moreover, providing high security, privacy, trustworthiness, availability, and integrity are the other critical requirements.

This work provides a novel and clean-slate proposal for a multi-actor mobile connectivity system that provides a distributed, trustful, automated, low-cost, and secure solution for the entities and actors of the cellular network ecosystem who aim to collaborate beyond 5G and 6G on top of Blockchain technology. This system can deliver trustworthiness, high automation, high coverage, a competitive market, and high security. In this regard, two general solutions are proposed to 1) facilitate the cooperation among different actors of the cellular network ecosystem such as MNOs, service providers, small-scale businesses, vendors, and end-users, and 2) facilitate the collaboration among MNOs and authorization bodies. It is important to mention **that the validity domain of the proposed system is to introduce a solution for a secure, trusted, automated, scalable, and distributed system for collaboration among entities and actors of the cellular network ecosystem.** In other words, the proposed system aims to broaden the collaboration of the cellular network actors from their current business to other areas such as regulation, incentives, business, services, etc. However, the proposed solution is able to address some technical issues such as eliminating the single points of failure, improving security, providing more automated IT procedures, etc.

To provide a unique terminology for the rest of the manuscript, we define some titles as follows:

- End-user: The individual or organization that utilizes the network, product, or services as the final user.

- Mobile Network Operator (MNO) is an independent communication service provider that owns the complete telecom infrastructure, including Radio Access Network (RAN) and Core network, to deliver wireless voice and data communication through cellular networks.

- Connectivity/network provider is a third party that is approved by MNO to provide the radio connectivity part of the telecommunication. This definition of connectivity provider can be a new phenomenon to include even very small-scale enterprises or groups of people who can provide RAN part of the connection for a small number of users (e.g., in the scale of a building).

- Service provider: In general, the service provider is an organization that provides different kinds of services, including storage, social media, entertainment, video streaming, consulting, gaming, communication, etc. In this work, the service provider is a third-party organization that provides different services to the users, on top of the internet, for instance, application service providers (ASPs), storage service providers (SSPs), internet service providers (ISPs), etc.

1.2 Motivating scenario

Assume the following future possibilities from the perspective of various parties in the cellular network ecosystem, such as MNOs, service providers, vendors, connection providers, and end-users.

From the **standpoint of the MNO**, they now operate in a stand-alone ecosystem in which they cannot have automated and non-manual contract-based coordination with other ecosystem participants. The issues raised by this stand-alone implementation will be discussed in detail in the next subsections (see section 1.4). For the time being, let us outline the future situation in terms of collaboration from the perspective of the MNO. MNOs can build a multi-actor mobile connectivity system in the next-generation networks and new cellular network ecosystem, in which they can collaborate with an unlimited number of actors such as connectivity providers, small-scale providers, organizations, service providers, content providers, and so on, to establish automated, secure, and trusted cooperation through non-stand-alone architecture. Moreover, despite the fact that the complexity of next-generation networks may surpass the capability of one MNO to manage the cost and the complexity of connection for a huge number of interconnected elements [8], the MNOs in multi-actor mobile connectivity systems are able to decrease the IT operation complexity without need to trust third parties and pay the imposed costs of outsourcing.

Furthermore, the MNOs are able to decrease the processing loads and latency of managing the collaboration contracts with other service providers, companies, organizations, and users. So, the process is managed more agile and automatically.

From the viewpoint of the **small-scale enterprise or other connectivity providers** that aims to provide network connectivity and call services for the end-users, in our scenario, they will be able to start the collaboration with MNOs with minimum effort of agreement, contract management, manual paper-work procedure, tracing, etc. In this scenario, they can deploy an automatic contract with a group of MNOs to be able to connect to their core network and provide call/network services to their end-users (note that, this goal is along with the MNO's ability to serve more users and provide better services in remote areas, as well as improving the user experience and satisfaction regarding accessibility). Apart from the collaboration, in the future, these independent providers (who can provide services to their users in stand-alone mode as well) will get paid based on their provided services to the end user through their collaborative connections with MNO.

Service and content providers (e.g., cloud computing, video streaming, online games, remote meetings, storage provisioning, etc.) will also be able to build a collaboration with MNOs and other providers, on top of existing capabilities of cellular networks such as authentication, access control, etc. to provide services for the users. Same as the connectivity providers, service providers are also able to deploy their contracts automatically with minimal manual effort for managing the contracts, payments, etc. Indeed, this capability has a huge benefit for the service providers, their business, and operational costs.

As the final part of the futuristic scenarios, we would say that putting the previously mentioned scenarios into practice will bring some possible use cases and opportunities for the **end-user**. In this case, the end-user will access the network services, regardless of their geographic location or base MNO (i.e., the MNO to which Alice is subscribed, and has its SIM card). It means, even if they are living in a very remote rural area with a minimal number of habitats, there is a possibility of providing a network connection by the small group of people living in that area.

1.3 Derived requirements

To put the aforementioned scenarios into practice, there are many challenges that need to be addressed. Indeed providing an exhaustive list of the challenges is not in the scope of this work and we only focus on the most important ones that evolve more actors. It is important to mention that to handle the growing demands of users and businesses for innovative services in next-generation mobile networks, flexibility, distribution, and security have become vital issues. One of the first steps to address these requirements is to propose

an open market with a more flexible ecosystem. The free market supports the entrance of diverse competitors to provide more innovative services, encourage investment, improve fairness, enhance user experience, introduce more cost-effective models, and enrich the technical aspects [9, 10].

As some examples, first of all, providing a multi-actor collaborative ecosystem that needs the cooperation among MNOs and different external providers can bring many concerns regarding **trust** in this environment. Moreover, managing the huge number of agreements, contracts, and engagements among actors requires, not only trust in the ecosystem but also a high level of **automation** and **scalability** (note that, here, we mean scalability regarding the number of actors in the system, users, IoT devices, etc.). Moreover, due to the nature of cellular networks, system **availability** is another important challenge to address. From the performance viewpoint, 5G and next-generation networks need very low-latency communications. So, **latency** is another vital issue to address. The **privacy** of actors, their businesses, user identity, etc. is highly critical when an unlimited number of service and connectivity providers can connect their businesses to the core network of the MNO. Another challenge is the compatibility of possible implementations with the existing architecture. The **compatibility** challenge will raise new kinds of questions ranging from being compliant with the existing cellular network standards to being compatible with the legacy software, hardware, and architecture. Finally, the implementation, maintenance, and operation cost of the proposed system should be taken into account.

In summary, the following requirements need to be met. Note that, some of these requirements are common for any mobile connectivity system (not only for multi-actor systems).

R1 ***Higher automation***: The entrance of new entities in a multi-actor collaborative system of cellular networks, would increase the processing overhead of IT operations, agreement management, contract handling, the cost of tracing, SLA tracking, etc., in MNO and other providers. Dealing with the further complexity of these operations requires higher automation in the procedures of trustful, automatic, and agile management of agreements among the main actors, and managing the secure access to the users' or application's data.

R2 ***Trustworthiness***: In order to provide the possibility of participation for different entities, competitors in business scale, and different technologies in the network, it is inevitable to have an ecosystem that can deliver trust in a distrustful environment containing unlimited numbers of service providers, connectivity providers, and MNOs.

R3 ***Distributed/ decentralized ecosystem***: To overcome several existing defects of stand-alone and centralized architecture (in which the whole process of providing connectivity is in the hand of one entity) such as high cost of mutualization and sharing, low scalability regarding agreement management and contract handling, low fault tolerance in centralized points, and availability, and to decrease the complexity of IT procedures in the cellular network and other service provider's systems, having a distributed or decentralized ecosystem is critical. Here, on one hand, the decentralized ecosystem indicates that the process of MNOs and, in general, connectivity provisioning will be handled in collaboration among different entities. On the other hand, the distributed ecosystem, or architecture in more precise words, is related to the implementation of the system. is mostly related to how the connectivity management will be handled.

R4 ***Scalability***: System scalability in the scale of the massive number of users (ranging from end users to IoT devices, etc.), collaborators, connectivity providers, and service providers is an important issue to address. Because, in the use-case of cellular networks and next-generation networks, scalability, availability, and low latency are critical features.

R5 ***Security and Privacy***: Indeed, due to the importance of transferred data in cellular networks and their applications in next-generation networking, such as user's Personally Identifiable Information (PII) data, healthcare information, financial data, etc., these networks are very attractive targets for malicious activities and preserving its security is a critical requirement to address. On the other hand, the information (including the customer lists, revenue statistics, costs, asset information, etc.) is the most critical asset of the businesses. In the collaborative ecosystem, providing sufficient security and privacy for businesses is an important requirement as well. Moreover, providing an open data architecture for 5G service provisioning such as flexible spectrum sharing, data sharing, multi-user access, etc. requires high data immutability and transparency [10].

R6 ***Compatibility***: For an agile migration to a new ecosystem, its compatibility with legacy systems, hardware, software, and architectures is the first requirement. Moreover, in standardized ecosystems, such as cellular networks (that are standardized by organizations such as 3GPP), being compliant with the standards is a vital issue.

1.4 Limitations of existing telecommunication ecosystem

To implement the aforementioned futuristic scenario, current cellular network architecture is suffering from several weaknesses. In this section, we are going to explain these defects and how they can negatively affect the user's and other actors' experience in this ecosystem.

Mobile Network Operators (MNO) are major telecommunication organizations to provide cellular and Internet services (i.e., communication services) to users. A Mobile Network Operator (MNO) is a telecommunications service provider that owns the complete connection infrastructure for hosting and managing mobile communications.

The current ecosystem of MNO, depicted in Fig. 1.1, shows that the suppliers, MNOs, business-scale customers, and the end-users are the main actors. MNO suppliers are the vendors that provide the hardware and software infrastructures of communication. For the hardware, the vendors sell the Base Station (BS), antenna, towers, storage, etc. to the MNOs. These suppliers are called TowerCos or InfraCos [11]. Moreover, other suppliers would provide the required software such as network functions. Using the supplied infrastructure by vendors, MNOs are able to launch the Radio Access Network (RAN) and core network to provide wireless/wired connections for the end users and business-scale customers. (i.e., the enterprises and companies that use the infrastructure provided by MNO to serve their users with different services such as content provisioning, video streaming, remote conferencing, Internet-based calls, and other internet-based services). These businesses may have a separate contract with MNOs to provide these services.

Focusing more on the MNOs, in the following part of this section, we list and explain the existing *business* and *technical* challenges in the current cellular network architecture that can negatively affect the possibility of implementing the explained futuristic scenario.

1.4.1 Business-related challenges of existing MNO architecture

As defined earlier, An MNO is a telecommunication provider that owns a complete cellular network infrastructure to manage the mobile communication between subscribed users and other users or data networks. From a high-level perspective, the MNO architecture is divided into Radio Access Network (RAN) and Core Network [12]. Currently, the existing cellular networks are deployed in a stand-alone manner in which the whole infrastructure is provided by MNO and **there is no collaboration among MNOs or other providers and actors** in the cellular network ecosystem to cooperate in service provisioning. The following business challenges are considerable regarding the existing stand-alone centralized architecture of cellular networks:

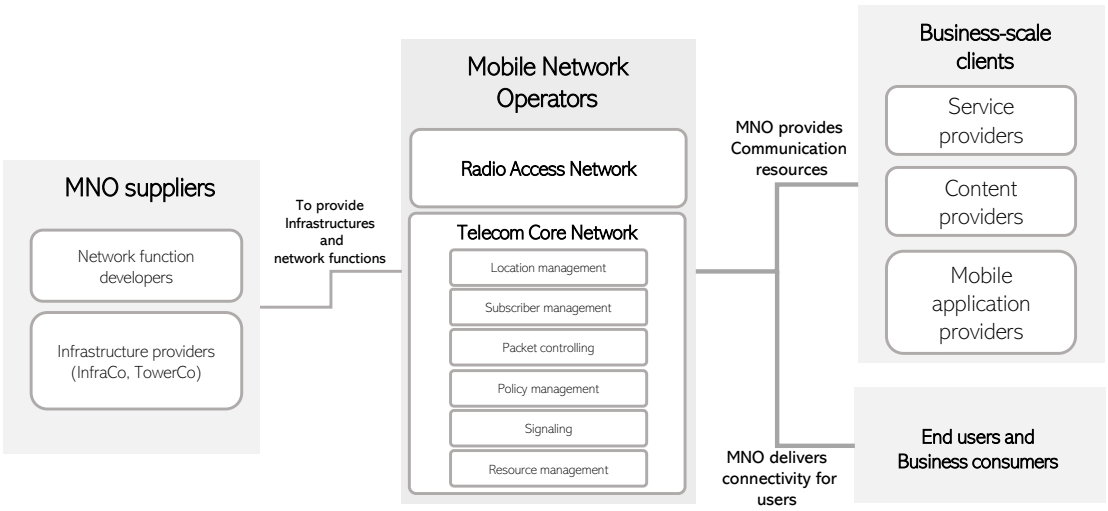


Figure 1.1: The current ecosystem of mobile network operators in the telecommunication market

- **High operation and installation costs for MNOs and providers**: Due to the lack of collaboration among entities (i.e., MNOs, connectivity providers, etc.), every MNO has to install a complete infrastructure of the cellular network containing RAN and core. Indeed, **the deployment of MNO infrastructure is a highly expensive investment, both for buying spectrum and deploying networking entities.** Mutualization of the resources such as RAN, storage, etc. can decrease these operational costs in MNOs and service providers. Moreover, this limitation can be the source of some other problems such as **monopolization of the market** and **imposed services and prices**. In other words, because the users select an MNO based on their proposed price, coverage, and service reputation [13], it is highly unlikely they choose a small-scale MNO with an unknown service satisfaction reputation. So, in the first step, many of the new businesses would be rolled out, and the cellular network market would be monopolized by several large-scale organizations (e.g., in 2014, four large MNOs in the US collectively comprise more than 95% of the cellular network market [14], and in 2021, 96% of market share belongs to three MNOs [15–17]). This may appear in contrast with the competition policy [18, 19] that aims to promote competition and thus customer welfare. Furthermore, because of the high installation costs and the lack of competitors in this sector, the dominant MNOs can impose their services and prices. In other words, there is no motivation for MNOs to add more value to their products or upgrade them [9]. In case of upgrades from the MNO side, their revenue will be flat, but their investment must increase.

- **Environmental effects and energy consumption of the separated and stand-alone infrastructure**: if the MNOs provide connectivity for the users, are highly more than the scenario in which the MNOs can share these infrastructures.

- **lack of collaboration between content/service providers and MNOs:** In the current cellular network architecture, rather than no collaboration among MNOs, there is no cooperation among MNOs and other service providers. As a result, MNOs always have to expand their networks, connections, services, coverage, etc. without any collaboration or payment from other providers. As a result, the revenue of MNOs will be flat, but their investment must increase (which is not highly beneficial for the stakeholders in this market). Recently, Europe has passed a law to require large-scale providers to provide more investment resources for MNOs to expand their coverage and services. In this case, only users are not the actors who are paying the MNOs for their services.

- **Contract management complexity:** Increasing the number of collaborators in cellular networks, would result in tremendous growth in the number of contracts that make their management impracticable for MNOs with current stand-alone, non-automatic, and centralized architecture [9]. In other words, managing and tracing the complexity of engagements, agreements, Service Level Agreements (SLA) and contracts have skyrocketing operational and financial costs for MNOs. As an example of this issue, let's assume that MNOs aim to broader their coverage in different geographical areas. Indeed, scaling the coverage of the connection provided by the MNO, without installing the whole infrastructure, needs a huge number of contracts with other connectivity providers. In the current mobile networks, a user can obtain service from an MNO with which she has a pre-established contractual agreement. This agreement can be a direct or indirect contract (i.e., in direct type, the user has a contract with MNO, and in the indirect one, they have a contract with MNO A which has another agreement with MNO B that is authorized to serve the user). Using these contracts, MNOs can serve more users, outside of their coverage range. These contracts between MNOs and the ones between MNOs and users bring the trust about accessing the user's information for authentication and billing them. In this scenario, when the number of small-scale providers increases (either connectivity providers or other functionalities in the core network), this solution *scales poorly* because all the agreements need to be handled and tracked manually and carry high transaction costs. Indeed, today managing this number of agreements is a feasible task, but in an environment with many smaller-scale providers, the number of agreements required to ensure broad coverage would quickly become untenable [9]. Moreover, third-party

access to user data is inevitable for authentication and billing procedure. this can bring *privacy concerns* for the users.

1.4.2 Technical challenges of existing MNO architecture

Apart from business-related challenges in existing MNO architecture, several technical issues negatively impact the security, reliability, and availability of the current cellular networks. Some challenges are listed below:

- **Centralized architecture:** Currently the MNO architecture lays on a centralized architecture in which the implementation and hosting of the infrastructure are located in limited sites and managed by one/limited number of entities. This centralized architecture suffers from single-point failures, which can affect the availability and fault tolerance of the system [10, 20]. Moreover, the overall performance and scalability of the network (to provide communication and security services) are highly dependent on the capacity of the centralized entities [21, 22]. Furthermore, the centralized architecture of the MNOs can decrease the flexibility of the billing systems and address the requirements of all entities [21]. From the functional viewpoint, the current centralized network design raises a major challenge in handover (i.e., mobility) management and performance enhancement [23, 24]. From the user identity management viewpoint, in current MNOs, a centralized unit handles subscribers' profile management and stores their subscription data. As mentioned by Tahir et al. [8], centralized storage can be a single point for a data breach or data leakage.

- **Complexity of IT operation:** The centralized architecture of the conventional MNO results in handling all the connections in a centralized party. This model increases the processing load and overhead in the central point, reducing the quality of service (QoS) and increasing the complexity of the IT operations [25]. Moreover, due to the ever-growing complexity of beyond 5G networks, one MNO may not be able to handle its cost and operational overhead [8]. As mentioned by [10], the security management in 5G is more complex due to various types of and a massive number of devices connected, and it brings higher IT complexity to the current networks. Rather than the complexity of IT operations, resource management also suffers from high complexity in the current MNO architecture. Due to the growing demand for resource provisioning and the variety of services, the management of communication resources (i.e., connection channel, processor, memory, bandwidth, and storage) is faced with an unprecedented level of complexity [10, 26].

- **Lack of automation**: Conventional network and service management solutions in current network operators suffer from a lack of automation. for instance, in current MNOs, the billing process is complex and inefficient based on Trusted Third Parties (TTP)s, which is not suitable for spectrum and infrastructure sharing [27, 28]. Another example is the lack of automation in the multi-domain area to address the requirement for End-to-End (E2E) network and service management [29].

- **Security and privacy issues:** these challenges can be categorized as follows:

  - *Authentication and access control procedure:* Generally, the complexity of authentication, access control, and data integrity in the conventional architecture of MNO are critical security challenges [30]. Rather than complexity, the centralized authentication and access management process limits the scalability of the AAC procedure in next-generation networks with high demand on the interconnecting tremendous number of devices (e.g., in IoT environment and smart cities) [21].

  - *Data storage:* Moreover, the user's personal information in cellular networks is an attractive target for advertisement and intelligent agencies. So, privacy turned out to be a significant concern of the users [31]. In current cellular networks, the user's privacy can be violated in the storage and applications of third parties, end-to-end data transmission through several stakeholders, and storing of user's data in a shred environment [30]. As mentioned by Tahir et al. [8], centralized storage of the user's identity can bring a significant challenge regarding security and privacy.

  - *Challenges proposed by new technologies in cellular networks:* To solve many problems existing in previous generations (i.e., 2G to 4G), several technologies such as Software-Defined Networking (SDN), Network Function Virtualization (NFV), and cloud computing are proposed. Indeed, these technologies are highly effective in providing greater application responsiveness, better network programmability [32] and management [21] by decoupling hardware and software [33, 34], introducing the micro operators [35], delivering differentiated services with network slicing [21, 36], etc. However, they may aggravate some of the existing challenges such as network reliability, security vulnerability, data privacy and immutability, multiple access control, authentic VNFs [21, 36, 37], etc.

  - *Trust issues:* As mentioned before, the main goal of next-generation networks and beyond 5G, is to provide an open and diverse ecosystem in which different

entities can participate and deliver/use services. In this kind of ecosystem, trust establishment is a cornerstone to adopt the technology [27, 38].

– *Privacy issues in TTP*: Due to storing huge amount of user's PII data (e.g., identity, location, and other sensitive data) in the storage of TTPs, privacy issues are often raised from TTP perspective; Since TTP have the privilege to access the confidential data, the malicious entities may use their ability to compromise the user's privacy [27, 29].

1.5 Blockchain-based cellular network architecture

The existing challenges in the current cellular networks on the one hand, and the requirements of the next generation networks (i.e., providing distributed trustful ecosystem with higher automation, scalable, and security that can guarantee the actors' privacy) on the other hand, indicate that in the future networks, the stand-alone and highly independent cellular network architecture will not be effective. So, in this work, we aim to propose a **multi-actor mobile connectivity system**. It is important to mention **that the main goal (and validity domain) of the proposed system is to introduce a solution for a secure, trusted, automated, and distributed system for collaboration among entities and actors of the cellular network ecosystem.** In other words, the proposed system aims to broaden the collaboration of the cellular network actors from their current business to other areas such as regulation, incentives, business, services, etc. Indeed, if the entities aim to act completely independently and are not interested in collaborating with other entities, this solution is not valid.

Any alternative addressing the requirement of this kind of system (i.e., distributed, trustful, scalable, secure, highly compatible, and highly automated) would be a promising potential solution to go through the new clean-slate solution to shape the cellular network market, ecosystem, and architecture.

Looking at both aspects of technical challenges and business challenges, Blockchain technology is one of the interesting candidates to be considered as a role-player in shaping B5G and 6G networks. Blockchain [39] is a peer-to-peer distributed ledger, cryptographically secure, append-only, immutable, traceable, and transparent technology that is only updateable via consensus among a majority of the participating nodes on the network [40]. Blockchain is a distributed ledger, structured into a linked list of blocks that contain an ordered set of transactions. To create a link with the previous block, each block uses the hash of the previous block. Due to its unique features, this technology can bring many unprecedented opportunities in the architecture of cellular networks. This technology and its

extension, smart contracts [41], can revolutionize the cellular network market [10,27,42,43] because of its decentralized deployment (i.e., the governance of Blockchain is not in the hand of one entity) and distributed processing (i.e., all nodes in the system participate in transaction validation), immutability, transparency, traceability, non-repudiation, and trustworthiness. Table 1.2 provides the benefits of this technology for cellular network architecture and shows how this technology can transform the existing ecosystem.

Based on the challenges and the features of the Blockchain, **the main research question in this work is that How B5G and 6G can benefit from Blockchain technology?**

Due to its unique features to address the needs of next-generation networks, in the following of our work, we profit from this technology to propose a novel architecture for core networks and cellular network-related services beyond 5G and 6G. Our proposal can potentially bring many contributions to the definition of future 6G cellular networks. **The proposed architecture benefits from decentralization regarding the governance and the organizations/entities that are managing the network and are its owner; It means, the entities and actors of the network are not limited to one MNO, but other connectivity or service providers are also capable of collaborating in the system and participating in its security, management and governance, and to serve their users. Moreover, the distributed implementation of the functions (i.e., operation of functions in different nodes in Blockchain) brings the opportunity to address the security and availability issues.**

The proposed architecture focuses on migrating the following main functionalities of core networks to a distributed system empowered by Blockchain consensus and its intrinsic security features:

- User subscription in MNO and user profile management;
- User registration to the network and key agreement (conventional AKA procedure);
- Access control for resource provisioning in the core network and the service layer;
- Mobility management (handover);
- Trusted payment procedure (billing);

The main reasons for selecting these functions for migration are:

- The inherent properties of Blockchain technology are quite useful in these functions. Immutability, non-repudiation, and the use of PKI, for example, can be advantageous in creating AKA procedures with a fewer number of message passing.

Table 1.1: How Blockchain technology is beneficial for cellular network

Feature	***How can be beneficial for cellular network***
Transparency	Blockchain maintains a complete history of past transactions within the distributed ledger. So, the users can track the data with full transparency [40]. This feature can bring *trustworthiness and reliability in a distrustful environment*. Since, beyond 5G would integrate many diverse types of entities, organizations, stakeholders, and businesses, it is inevitable to provide a transparent and trustful environment. Moreover, this feature can provide traceability in the system that is required for billing, data/resource sharing, etc. [22]
Immutability	Along with transparency, immutability of Blockchain *increase the trustworthiness* of the system [10, 44]. Moreover, this feature highly improves the rule, and policy immutability which is a prerequisite of service provisioning, authentication, access control, and billing in B5G [10]. Furthermore, management of the user's subscription and identity in the cellular network needs high data, credential and identity immutability [10,45], which can be provided by Blockchain's tamper-proof and immutable storing manner. Another benefit of immutability in a cellular network is to facilitate performing accounting tasks. For instance logging of session statistics and usage information for billing, resource utilization, and trend analysis [10,46].
Decentralized architecture	Due to the Blockchain's distributed architecture, there is no need for any third party, intermediary or central point to handle the process, manage operations or provide security in the system. This unique feature eliminates the single point of failure risks due to the disruption of central authority, saves operational costs, and enhances trustworthiness in the network. Moreover, it eliminates any security risks related to the central point such as privacy violation in TTPs [27, 29]. Moreover, the scalability limitations of centralized systems can be eradicated by the Blockchain to face the envisaged massive connectivity demand in B5G [21].
Consensus	Since there is no central authority to manage the security of the system, Blockchain uses consensus algorithms to provide integrity of transactions and data in the system. Reaching consensus in the system, along with immutability and transparency, not only brings high data and transaction integrity to the system but also highly increases the trustworthiness of the system.
Automation	Different from the conventional authentication, access control, and mobility management operations that often use a centralized server to perform the functions, smart contracts (as a piece of code that would be executed once the predefined conditions are satisfied) on top of Blockchain, can implement decentralized AAC and mobility management procedure. This feature can highly increase the automation of the operations in the system.
Non-repudiation	Using the users' signatures (i.e., encryption by the user's private key) on each transaction eliminates the possibility of action denial that is highly required in secure authentication and access control procedure.

- Some operations, such as resource management, are largely tied to internal MNO processes, and each MNO or provider may use different techniques to handle the functionality; hence, adopting Blockchain in these functions cannot be advantageous (at least in our use case).

- Migration of some functionalities to Blockchain might have a detrimental impact on system performance. For example, migration of the signaling function to Blockchain can raise the need for storage and high bandwidth to support the number of messages passing.

- To choose the fewest functions for migration to Blockchain in order to provide: 1) the ability of collaboration for entities, and 2) the basic functionalities for the user's connection to the network in the novel architecture.

Fig 1.2 depicts a high-level schematic of the proposed architecture. In the conventional MNO architecture, the user plane and control plane are handled by mobile operators using a centralized approach. To fulfill the pre-mentioned requirements of next-generation networks, we proposed an architecture that combines centralized/decentralized and distributed solutions to introduce a semi-distributed cellular network architecture. As a result, the external providers -either connectivity providers who would participate in RAN and increase the coverage, or service providers, who would provide application layer services such as video streaming, conferences, etc.- can benefit from providing their added-value services using the pre-exited infrastructure.

In our context, the main beneficial feature of Blockchain is its capability to provide trustworthiness and reliability in a distrusted environment. Indeed, the main concern of opening the architecture and allowing the entrance of the new external entities, to provide connections and services in MNOs, is the challenge of providing trust in the new environment. Blockchains intrinsic features such as immutability, non-repudiation, consensus, and transparency are able to provide trust in the network between collaborators. So, we can state that, Blockchain-based networks could more easily allow Micro Operator (µO)s, small-scale connectivity/service providers, or even individuals to enter and flourish in the cellular network markets. As an example of a win-win condition for MNO and small-scale RAN provider, assume that a small group of users is living in an unpopulated rural area that is a white spot for MNOs (in which the cellular network antenna has not any good coverage). Providing RAN part of the connection for these areas with a low population is not beneficial for the MNOs. In this case, a small group of users who supply the RAN part of the connection can help to provide better coverage for the users in that area. So, if they can serve their users by connecting their RAN to the MNOs core network and at a

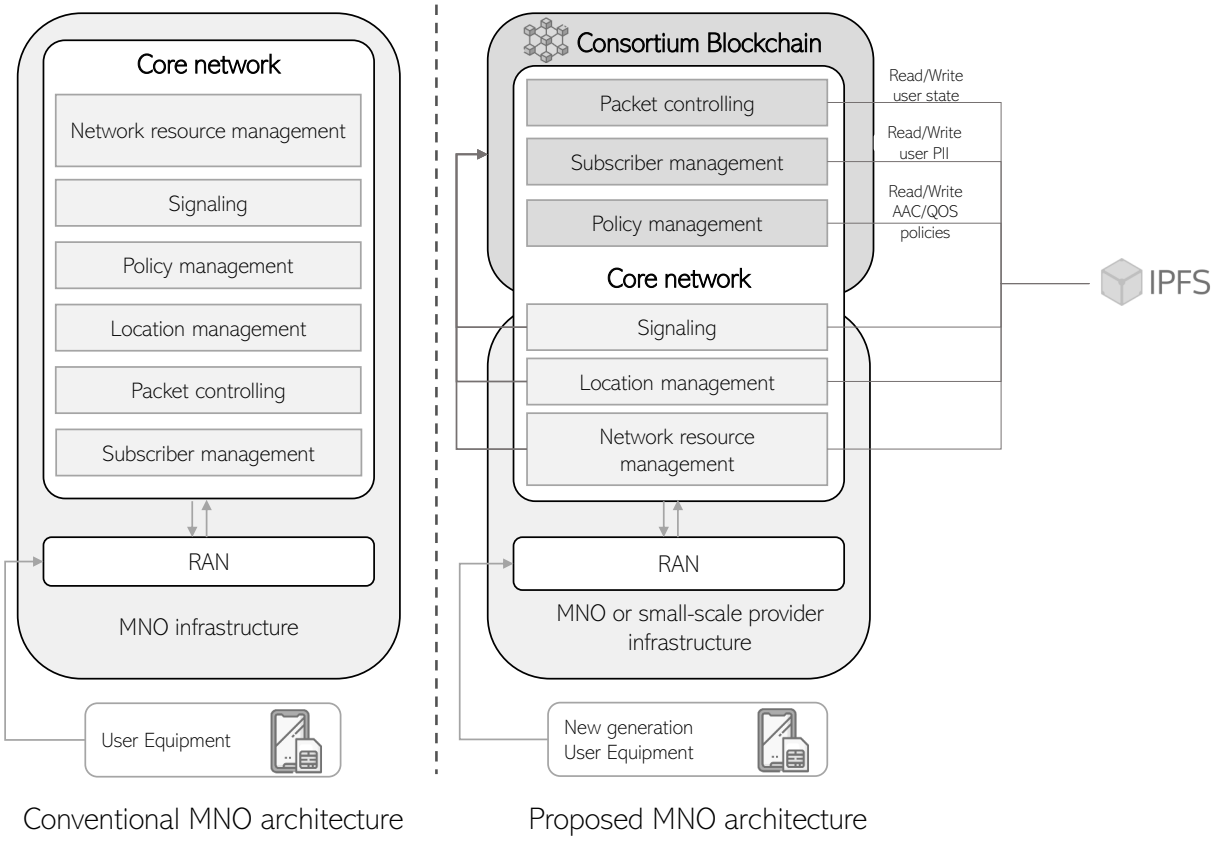


Figure 1.2: The High-level overview of existing MNO architecture vs. the proposed architecture.

low price, all parties (MNO, users, and small-scale providers) would benefit. Moreover, the Blockchain's promising security properties can offer a new innovative solution for security, privacy, and performance improvement in cellular networks [10].

1.6 Incentives and benefits of proposed method

The key requirement for introducing a new architectural approach is the acceptance of the different actors (i.e., users, MNOs, small-scale connectivity providers, independent vendors, and service providers) of the ecosystem. So, several advantages of the proposed method and its benefits for the different actors are also listed in the following.

- ***Collaboration opportunity:*** The proposed distributed architecture for beyond 5G and future 6G networks, allows the MNOs, small-scale providers, service providers,

connectivity providers, etc. to collaborate and cooperate to provide services for the users while decreasing the mutualization costs, environmental effects and MNO or providers operation costs. Moreover, entering the new providers in the cellular network market gives them the opportunity of serving their users at a very low initial expense (i.e., by registering in the list of authorized external entities in the consortium Blockchain to be able to serve their users). They can deliver connectivity services (e.g., by providing RAN or connection antenna and connecting it to the core network provided by MNO, similar to MVNO), security services (e.g., by outsourcing the AAC and identity management procedure to the Blockchain), and data management services. Its result would be creating a more competitive market that leads to lower prices and higher customer satisfaction [4, 18].

- ***Automatic and cost-effective contract management:*** As mentioned earlier, one of the big issues in managing collaboration in the existing ecosystem is manually handling the huge number of contracts and engagements. This management imposes high operational and financial costs on the MNOs, which makes it infeasible for the next-generation networks. Thanks to smart contracts and their features, the proposed system provides the opportunity for automated deployment, tracking, payment, and termination of the agreements.

- ***Opportunities to develop new business models:*** Using the new architecture, the MNOs, connectivity providers, and service providers can deliver more innovative business models. For instance, the **users pay directly to the service/connectivity providers**, MNOs can deploy a **hybrid billing model** of "pre-paid" and "pay-as-you-go" solutions, MNOs can charge service/connectivity providers based on their usage, etc. Another example is the **connection with an external RAN provider**. In an area with a small population, it is not profitable for MNOs to deploy a complete connectivity structure. So, µOs or small-scale connectivity providers can cover this area using the core network infrastructure provided by MNO (e.g., MNOs can benefit from broadband user access at lower cost) [9]. Moreover, by migrating several procedures from the MNO core network, the complexity and operation cost would decrease for MNO. Apart from the aforementioned opportunities, Using the new architecture, different companies, businesses, universities, corporations, etc. will be able to introduce their own **private cellular networks beyond 5G (i.e., private B5G)**, by which they are able to serve their specific users based on their needs. Indeed, implementation of the proposed method in private networks is also feasible.

- ***Technical benefits:*** Firstly, in the proposed model, the billing procedure is re-

moved from the MNO side, so the user can directly pay service providers. Not only, the user's direct payment to the external providers would be more beneficial for their businesses, but also it decreases IT complexity and maintenance costs from the MNO side. Second, user registration, AAC procedure, and mobility management are handled by smart contracts. So, the procedures will be performed in a secure, immutable, and automatic manner. Third, this migration provides the opportunity of outsourcing the procedure to a distributed system which results in increasing the availability and fault tolerance of the system. The proposed method can inherit the Blockchain's overall benefits (as provided in Table 1.1). For instance, offering the immutability of rules, prices, and data due to Blockchain's cryptographically secure nature; improving the accountability and non-repudiation because of signing the transactions in the Blockchain, and providing higher privacy and anonymity [21, 47–49] by using pseudo names instead of the user's PII data in the network.

1.7 Contributions

To provide the possibility of implementing a multi-actor mobile connectivity system in B5G, in which different stakeholders and cellular network actors can collaborate more effectively, trusted, automated, and securely, this work makes the following contributions. These contributions are mainly disseminated in the original publications as shown in Table 1.2.

1. A comprehensive study on the existing challenges in current cellular networks regarding business and collaboration aspects, technical issues, and security. These studies are more focused on authentication, access control, and identity management or providing services to legitimate users and how DLT-based solutions may address these challenges in a broad range of different networking use cases and applications.

2. A novel hybrid (distributed-decentralized) Blockchain-based architecture to introduce a multi-actor collaboration system **among different actors of cellular network ecosystem (i.e., MNOs, service providers, small-scale businesses, vendors, and end-users)**. In this regard, a hybrid core network is proposed for MNOs beyond 5G and 6G, which provides the opportunity for trusted collaboration among entities, creates a competitive market, increases the fault tolerance of the system, simplifies the IT procedures, and provides a secure payment among entities. The proposed method system covers the principal core network procedures such as user subscription in the MNO, profile management, authentication, access control procedure, registration requests, mobility management and handover, session management, and billing.

Table 1.2: Contributions and research articles

Contribution	Publication
1	[J.1], [J.6], [C.5], [C.7]
2	[J.2], [J.3], [J.4], [C.8], [J.5], [C.2], [C.3], [C.4]
3	[C.1], [C.6]

Moreover, it supports a novel solution based on smart contracts to provide flexible access control in a new business model for service provisioning in cellular networks.

3. Proposing a new user profile management and mobile number and profile porting solution on top of Blockchain and smart contracts to introduce a multi-actor collaboration system **among MNOs and authorization bodies**. This method relies on the Blockchain addresses as an identifier, the user's key pair in the Blockchain for authentication and key management, smart contracts for process automation, and a distributed database to manage the user's profile. The proposed solution addresses the high latency of the existing method for porting the MNOs. Moreover, it brings the opportunity of porting the users' profiles to the recipient MNO as well as the phone number with an automated procedure without a centralized authority or third parties.

It is important to mention that to provide a more comprehensive explanation of the proposed system, we did not provide an explanation of each contribution in the separated chapters (i.e., the following chapters of the manuscript are divided based on the subjects and objectives of the method, not based on the contributions).

1.8 Organization

The rest of this work is organized as follows: Chapter 2, provides essential background on distributed ledger technologies, cellular network architecture and their functionalities, access control, and authentication methods. Chapter 3 surveys existing works regarding the concerned topics of the work such as distributed authentication and access control solutions in networking applications, DLT-based mobility management solutions, DLT-based identity management solutions, and decentralized cellular network architectures. The high-level architecture of the proposed Blockchain-based Cellular network architecture for Beyond 5G and 6G is provided in Chapter 4, followed by the explanation of designed smart contracts for Blockchain-based Cellular networks in Chapter 5, and the details of distributed network functions and their functionalities in Chapter 6. To evaluate the

proposed method, Chapter 7 discuss different implementation solutions and the model which we used in our work. Chapter 8 provides a scalability evaluation of the method and some security analyses. Finally, Chapter 9 is dedicated to different discussions on the proposed architecture, possible future works, and conclusions.

Chapter 2

Preliminaries

Contents

2.1 Introduction

Due to the growing demand for novel and innovative services delivered by Mobile Network Operators (MNO), flexibility, distribution, and security have become vital issues to address in next-generation networks. While the existing centralized architecture of MNOs is providing connectivity to billions of users, they are still vulnerable to attacks or failures, hardly flexible for innovations, and expensive to deploy and maintain. To address these defects, we propose a novel Blockchain-based architecture, as a clean-slate solution for beyond 5G and 6G networks to create a competitive flexible market, increase the system's fault tolerance, and provide a secure payment among entities.

In this regard, different functionalities of the conventional core network are migrated to the Blockchain as follows (as mentioned in contribution number 6 of Chapter 1.7):

- The existing Authentication and Key Agreement (AKA) procedure of cellular networks is substituted by a Blockchain-based authentication method using its intrinsic public-key-based authentication.
- The access control procedure (either in the core network or application/service layer, as provided in contribution number 4) is proposed to be done on top of the Blockchain with an Attribute-Based Access Control (ABAC), combined with Role-Based Access Control solution.
- The user mobility and session management procedures are proposed to be handled by Blockchain, thanks to its inherent public-key-based non-repudiation and mutual authentication.
- The centralized architecture of the user subscription and profile management (i.e., in both contribution numbers 5 and 6) procedures are replaced by distributed Blockchain-based solutions along with using distributed database solutions such as InterPlanetary File System (IPFS).

Before going into the details of these contributions, in this section, we provide the necessary background on the following subjects: 1) distributed ledger technologies (section 2.2) in which the different categories of this technology, their features and the consensus mechanisms in distributed systems. 2) The mobile network operator ecosystem and architecture (section 2.3) including different network functions, their main functionalities, and features. 2) authentication and access control methods (section 2.4) that explained different authentication solutions based on knowledge, possession, biometric and multi-factor solutions. Moreover, access control methods such as DAC, MAC, ABAC, and RBAC are

introduced, and finally, 4) the Blockchain-based authentication and access control solution in different network-based use cases.

2.2 Basics of Distributed Ledger Technologies

Distributed Ledger Technology (DLT) is a general term for technologies that utilize replicated, shared, and synchronized digital data among the users of private or public distributed computers located in multiple geographical sites [50, 51]. In general view, immutability, distributed/decentralized nature, consensus, transparency, non-repudiation, and being append-only is the common feature of all DLTs. Any change in the state or value in the ledger can be accomplished through consensus among the nodes. Increasing the number of nodes participating in the consensus procedure decreases the probability of monopolization of the network by several malicious nodes. Also, with more extracted blocks, the immutability of the information is improved [51].

This technology can be grouped into different categories. Regardless of their category these technologies share the same features and use different consensus models to provide security and trust in the network. The next subsections are devoted to these concepts.

2.2.1 The categories of DLTs

Based on its data structure and deployment model, distributed ledger technologies can be categorized into several groups as listed below.

2.2.1.1 DLT categories based on data structure

Based on its data structure, DLTs can be categorized into, at least, the following three groups [52]. Indeed the categories are not limited to these examples:

- ***Blockchain and Smart contracts***: Blockchain was introduced by Satoshi Nakamoto in 2008 and implemented in 2009 by a cryptocurrency called Bitcoin [53] to provide a peer-to-peer payment system. As a brief description, Blockchain is a peer-to-peer distributed ledger, cryptographically secure, append-only, immutable, traceable, and transparent technology that is updateable except via consensus among a majority of the existing nodes on the network [40]. Blockchain is a distributed ledger, structured into a linked-list of blocks that contain an ordered set of transactions. To create a link with the previous block, each block uses the hash of the previous block (see Fig. 2.1). The number of transactions in each block can be varied based on the number of input transactions per second and the difficulty of the consensus puzzle. In its structure,

each block has a header and a body. Regardless of the variety in the implementations of Blockchain and the blocks, most of the block headers have the following parameters: 1) a block version that defines the rules; 2) the hash of the previous block which ensures that previous blocks cannot be changed; 3) a hash of the Merkle tree root that stores the hash amounts of all transactions in the current block; 4) a timestamp for traceability; 5) a random number as a nonce, and 6) the hash amount of all the data in the header and body of the current block. Meanwhile, each block body contains a transaction counter and the transactions in the current block [40]. Fitting the transactions into a block in the order of their occurrence and publishing a new block in the ledger are accomplished through consensus among the nodes. Increasing the number of nodes participating in the consensus procedure decreases the probability of monopolization of the network by several malicious nodes. Also, with more extracted blocks, the immutability of the information is improved [51].

Introduced by Szabo in 1998 [54], ***Smart Contracts*** are computerized transaction protocols that execute the terms of a contract on top of Blockchain. After the proposition of Blockchain, this idea has found a proper infrastructure for its realization. The main objectives of smart contracts are to satisfy common contractual conditions, minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Today, different Blockchain platforms support the smart contract paradigm [55] (e.g., Ethereum[1], NEM, Hyperledger Fabric [2]); Ethereum is the first to implement smart contracts, in 2014 [56]. The most well-known smart contract language is Ethereum's Solidity [57] which has a "Turing Complete" virtual machine to run distributed applications and allow the execution of smart contracts [41]. Flint [] and SCILLA [58] are two other languages.

- ***BlockDAG***: Block Directed Acyclic Graph (BlockDAG) replaced the linked-list structure of Blockchain with the Directed Acyclic Graph (DAG). In this technology, the blocks are linked together via DAG [59]. In mathematics, a DAG is a one-direction graph without cycles connecting the edges. The main hypothesis of BlockDAG is to serve/validate transactions and blocks as fast as possible. To provide consistency in the system, the miners of new blocks decide on the order of the transactions [60]. Avoiding a single-chain structure in BlockDAG results in an increase in the mining procedure performance, the throughput, and the scalability of the system [61]. However, on the negative side, BlockDAG is more complex to deploy and more susceptible to some DLT-based attacks [62]. Tangle is an example of

[1]https://ethereum.org/en/what-is-ethereum/
[2]https://www.hyperledger.org/about

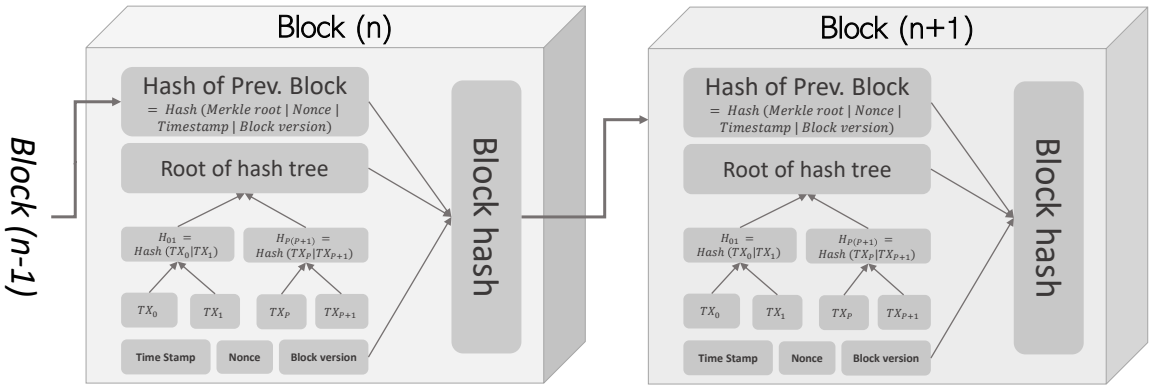


Figure 2.1: Block architecture in the Blockchain

a BlockDAG [63].

- ***TDAG***: transaction-based DAG or Block-less DAG removes the concept of the block. The impetus for this technology is that even in BlockDAG, different blocks may contain overlapped transactions which can increase the bandwidth requirements. So, in TDAG, transactions are linked directly together in the DAG structure, and there are no blocks at all [52]. IOTA and Nano are two examples of TDAGs. Similar to BlockDAG, TDAG has higher throughput and scalability, but it has more difficulties in terms of system consistency.

All of the above-mentioned methods share three common characteristics: 1) they use consensus methods to reach an agreement in the network and provide security, 2) they use a distributed peer-to-peer structure, and 3) they use public key infrastructure to provide non-repudiation [64]. To simplify the descriptions with a more well-known concept, we focus on Blockchain in the following sections. Indeed, many concepts are the same for other categories as well (e.g., types, features, and several consensus models).

2.2.1.2 DLT categories based on deployment model

DLT platforms can be divided into two main categories, based on their deployment and access permissions [40, 65].

- *Permission-less (Public)*: This type of DLT is a completely decentralized network without any central authority for management. It is accessible to the

public, and anyone can participate in reaching a consensus, reading transactions, and writing in the ledger. All the transaction records are available to all users. Two well-known networks in this category are Bitcoin [3] and Ethereum.

– *Permissioned*: there are two subcategories of permissioned DLT. 1) *Private DLT* is developed for different departments of an organization based on their needs; only the users who are confirmed by the organization cooperate in the consensus procedure. In private distributed ledger technologies, no currency or tokens are required to process or validate transactions. Organizations must know users so that they can control their access to the system, which means that users' anonymity can be violated. Also, its management is in the hand of a single organization, so its immutability is fragile. 2) *Consortium DLT* can be used as a distributed and reliable database for pre-defined enterprises for business-to-business (B2B) purposes. Similar to private DLT, only the nodes verified by participating organizations can join in the consensus process and fees are not mandatory.

Table 2.1 compares different types of DLT deployments based on their features [65–67].

2.2.2 The Highlighted Features of DLT

DLT has developed over the years. In 2008, Blockchain, as its first extension, was introduced to support cryptocurrencies in the financial sector. After the introduction of smart contracts in 2014, several applications such as stocks, loans, mortgages, and smart properties were added to Blockchain. Various industries, enterprises, and academic communities then discovered that immutability, decentralized nature, fault tolerance, transparency, permanence, non-repudiation, and other significant features of this technology can make it a promising solution to incorporate in several contexts. Recent studies show that economic and political governance, health, science, literacy, culture, art, humanitarian and legal areas can all be reconfigured by Blockchain [68, 69]. The main features of Blockchain, their descriptions and the solutions they offer are summarized in Table 2.2 [69, 70]. This table also indicates which feature can help resolve which problem.

[3] https://bitcoin.org/en/bitcoin-paper

Table 2.1: Comparison of DLT types [65–67]

BC type / ***Feature***	***Permission-less***	***Permissioned***	
		Consortium	***Private***
Participation in consensus	All nodes	Legitimate nodes	
Read permission	Public	Public or Restricted	
Write permission	Public	Restricted	
Node access	Permission-less	Permissioned	
Validation speed	Low	High	
Immutability	High	Low	
Availability	High	Moderate	Low
Integrity	High	Moderate	Low
TRANsparency	High	Partial	Low
Anonymity	High	Partial	Low

2.2.3 Consensus Mechanisms

A consensus mechanism is a sequence of steps followed by all or most of the nodes in a system to reach an agreement on a proposed state or value [40]. Consensus mechanisms can solve Byzantine failure in the Byzantine general problem [74]. The most important requirements of consensus mechanisms are briefly described below:

1. *Validity:* the decision of every honest node and the final decision of consensus must be proposed by at least one eligible node [75, 76];
2. *Agreement (safety):* via a predefined algorithm, all eligible nodes in a network must reach the same final value [40, 51, 75–78];
3. *Termination:* all eligible nodes should reach an agreement [75, 76]; and
4. *Fault/ Crash tolerance*: The algorithm must be able to work well in the presence of Byzantine nodes [40, 76].

Note that, as mentioned in Lamport et al. [78], safety and validity features (which we had mentioned separately) are proposed in one single feature as "safety". Consensus protocols can be defined as Byzantine Fault Tolerant (BFT) or Crash Fault

Table 2.2: Highlighted DLT Features [69, 70]

Feature	***Description***	***Problem***	***DLT solution***
Immutability	No confirmed transaction or data in DLT can be altered.	– Changing the transactions – Fraud	Using the hash of the preceding block makes it difficult to change the data in a block. However, if an attacker solves the consensus problem for all of the subsequent blocks, they can be changed.
Decentralized and fault tolerance	There is no central authority to control the network, and failure of one or several nodes cannot harm the system's functionality.	– Single point of failure – Low reliability	There is no central authority and all the nodes have a copy of the transaction ledger; all changes in the network occur after consensus among all or a majority of the nodes.
Reaching consensus	All nodes in a DLT can reach a consensus, based on algorithms defined to ensure that all nodes have the latest version of the ledger.	– Lack of data integrity – trust issues	All (or a majority of eligible) nodes in a DLT contribute to solving a consensus problem; the node that generated the new block announces the latest version of the ledger to the network. As mentioned before, the blocks are almost immutable, and thus the integrity of data and transactions are maintained.
Traceability/ Transparency	All transactions are available to be seen and tracked by the nodes.	– Losing the root cause of an incident	All data in a DLT is always available and traceable at any time. This feature can be especially useful in forensics [71–73].
Non-repudiation	No one can deny their actions in the DLT-based network.	– Action denial	Using the users' signatures on each transaction eliminates the possibility of action denial. The signature is provided by the user's private key which is expected to be held only by the owner.
Permanence	All data in a DLT can be available at any time.	– Data loss – Loss of availability	DLT relies on its distributed ledger for transaction data, which means all the nodes have a copy of it, and so these data are permanent.

Tolerant (CFT) if they can function normally in the presence of a particular number of Byzantine nodes or despite the failure of some nodes. BFT protocols are naturally subcategories of CFT methods [51]. The rest of this subsection introduces several well-known consensus methods. These methods are categorized into three different groups as noted by Ismail et al. [79]: 1) compute-intensive based, 2) capability-based, and 3) voting based. It is worth mentioning that due to the abundance of existing consensus models, presenting and analyzing all of them is out of the scope of this paper, so we only introduce the methods that are used in the studied AAC mechanisms.

2.2.3.1 Compute-intensive based consensus algorithms

These algorithms require a substantial amount of computing resources to solve a consensus problem. One of the most well-known examples of this type is Proof-of-Work (PoW). This algorithm was first proposed by Dwork et al. [80] in 1992 to combat junk and spam Emails, and in 1999, it was formally called *"Proof of Work"* [81].

A PoW algorithm works based on the framework of a cryptographic block-discovery racing game. Nodes (known as miners) try to solve a mathematically complex puzzle that uses a tremendous amount of their computational resources. The first miner that finds the result is the winner and can send it to all the nodes in the network (i.e., via the gossiping rule). Bitcoin deployed the PoW protocol, as proposed in [53], and so it has become known as Nakamoto consensus in several resources [51]. In Bitcoin, the winner wins a cryptocurrency prize. PoW can tolerate 49% of Byzantine nodes. This method also has high scalability in terms of the number of nodes (e.g., see the numbers of Bitcoin network users).

Other consensus models also function based on PoW. For instance, the *Tangle* consensus model of IoTA works on top of PoW. In Tangle, when a user wants to send a transaction in the network, it must validate two other unapproved transactions. Then, the user concatenates the hash of the two transactions with her request and solves a PoW puzzle. The user can then broadcast the proof of solving the puzzle [82].

2.2.3.2 Capability-based consensus algorithms

Due to the energy inefficiency of compute-intensive-based consensus algorithms, other alternatives have been proposed. Capability-based algorithms rely on the capabilities of nodes instead of their computational power. The number of tokens owned by a

node, a node's reputation, its storage capability, and its contribution to the network is some of the parameters utilized to assess the capability. A sampling of these algorithms is described below:

i. *Proof of Stake (PoS):* This method was introduced in 2012 [83] as an enhancement to the performance of PoW that reduced energy consumption. In PoS, the block validator (the only node that is responsible for generating the next block) is selected based on the stakes it would have. The next validator is chosen randomly, and the nodes with more stakes are more likely to be selected. Stakes are coins or tokens owned by a node. To validate a block, the validator should solve a hash puzzle that replaces the process of exhaustively searching the nonce in PoW [84, 85]. To overcome the problem of monopolization by rich nodes, the "coin age" measures the tokens held by a user and their holding time. Validators who approve incorrect transactions will lose their stake [51].

ii. *Proof of Authority (PoA):* This algorithm is a reputation-based method in which the reputation of the validator is the capability parameter. PoA is utilized mostly in permissioned Blockchains. Due to its lighter message exchange loads, this algorithm provides higher performance in terms of energy consumption and time efficiency [86]. The validators (authorities) in PoA have formally approved accounts, and their identity is kept public [87]. To fairly distribute the responsibility of mining the block, this process is done in mining rotation [86,87] (i.e., the time is divided into steps, and in each step, one validator mines a block). Two well-known implementations of this algorithm are Clique [88] and Aura [89].

iii. *Proof of Elapsed Time (PoET):* Proposed in 2016 by Intel corporation [90], this algorithm requires Intel's Software Guard eXtensions (SGX) [91] and Execution Environment (TEE) [92] to run. SGX generates a random waiting number for nodes and selects the node with the minimum expiration time as the leader (responsible for generating the new block) [93]. To avoid malicious activities, the generated random waiting time is signed and distributed in the network [79].

iv. *Proof of Importance (PoI):* A model proposed by NEM in 2018 [94], in which an "importance" factor is assigned for each node in the network. The nodes with higher "importance" factors have a higher chance to be chosen as a leader for generating the next block [94]. To assign the "importance" factor, the node's stake (token) in the network (at least 1000 XEM, the cryptocurrency

of NEM) [94], and the frequency of its successful validations are considered (i.e., validators with higher frequency receive more importance [87]).

2.2.3.3 Voting-based consensus algorithms

In implementing technological democracy, the miners and validators are selected based on the voting process among network nodes. Four of these types of methods are introduced below.

i. *Delegated Proof of Stake (DPoS):* this algorithm was proposed in 2014 [95] to decrease the energy consumption of the PoW algorithm and increase the democracy among nodes. This method relies on selecting delegates (witnesses) instead of the validators of the blocks. The witnesses can be interpreted as trusted nodes in the network, chosen by election to validate the blocks instead of nodes. Each node in the network can be chosen as a witness (based on its stakes) or can delegate its stakes to another node and select it as a witness (i.e., the weight of each vote depends on the stakes of the owner). After the election process, 21 to 100 delegates are selected based on their tokens (votes), and each of them will be randomly chosen as a validator for a predefined time. Delegates must validate the blocks in their dedicated timestamp, and in the cases of validating fraudulent transactions or failure, they will be expelled and replaced by another witness. Note that DPoS can also be categorized as a capability-based algorithm.

ii. *Practical Byzantine Fault Tolerance (PBFT):* In 1999 [96], the PBFT method was proposed as a general solution to guarantee the consistency of a distributed system containing Byzantine failure nodes. This method has one "leader" node and several other nodes as "backup" in each cycle. Byzantine nodes cannot be more than one-third of the existing nodes while increasing the number of whole nodes in the network can improve the system's security. This method has five steps: 1) The "Request" step, in which the user sends a request to the leader node; 2) The "Pre-prepare" step, where the leader node puts the request in the order and broadcasts a pre-prepared message to the backup nodes; 3) The "Prepare" step, where all the backup nodes that accepted the request broadcast a "prepare message" to all the other backups and receive the prepare messages from the others. After collecting at least $2f+1$ messages, the method passes to the "commit" step; 4) In the "Commit" step, all nodes send a commit message to

all the other nodes. If a backup node receives at least $2f + 1$ commit messages, it could be that the nodes have reached a consensus to accept the request, and thus, they can execute it; and 5) The "Reply" step is where the backup nodes reply to the user. Note that f is the maximum number of Byzantine nodes that can exist in the network. This method is energy efficient, but its scalability is limited [51].

iii. *Raft Algorithm:* A consensus method proposed in 2014 [97], in which, at any time, each node is in one of the three states as *leader, follower*, or *candidate*. The leader serves the network until it crashes. The "leader" sends a periodical "heartbeat" message to the "followers" to inform them about its activeness. When a leader fails (i.e., followers cannot hear a message from their leader within their specific waiting period), the election process starts. In the election process, a follower that does not hear from its leader changes its state to "candidate". Then the "candidate" requests votes from other nodes to become a "leader". Other nodes will reply to the vote request. If the "candidate" gets votes from a majority of the nodes, it will become a "leader".

iv. *Ouroboros:* This algorithm is a consensus model based on PoS, proposed by Cardano in 2015 [98]. In this method, time is divided into fixed-time epochs and the individual units of time within an epoch are slots. In each epoch, the electors can be selected based on the weight of the stake of the stakeholder. These electors are eligible to select the leader of the next slot. To choose the leader, three steps (i.e., a Publicly-Verifiable Secret Sharing (PVSS) process) are executed: 1) *Commit*, in which each elector broadcasts a commitment message that has a random secret; 2) *Reveal*, in which the electors broadcast an opening message to reveal the previously sent secret; and 3) *Recovery*, in which all electors verify that two previous messages match, and then form a seed string with the revealed secrets. All electors have the same seed string and the same leader sequence [51].

Comparison among different consensus models and several implementation examples are provided in Appendix A.

2.3 Mobile Network Operator architecture

A comprehensive understanding of its ecosystem and current architecture is inevitable to identify the challenges and defects in the existing cellular network and the moti-

vations to move toward new solutions in next-generation mobile networks.

A Mobile Network Operator (MNO) is a telecommunications service provider that delivers wireless voice and data communication through cellular networks for its subscribed mobile users. MNOs are independent communication service providers that own the complete telecom infrastructure, containing antennas to provide RAN and Core network, for hosting and managing mobile communications between the users with users in the same and external wireless and wired telecom networks [99].

The rest of this subsection is dedicated to essential background on the architecture of the latest generation of cellular networks (i.e., 5G) and its network functions.

As it is depicted in Fig. **??**, MNO infrastructure consists of RANs and Core network [12]. RAN is responsible for providing radio connectivity. It mainly consists of antennas (called gNodeB in 5G) to provide the radio connection in a specific area based on their capacity. In our proposed model, the RAN part remains intact. So, we do not discuss it further.

The principal functionality of the core network is to establish secure sessions and to forward user data to and from the Data Network (DN) to provide connectivity [100]. Starting from the 14th release of 3rd Generation Partnership Project (3GPP)[4], the core network is divided into User plane and Control planes. The user plane carries the network user traffic, while the control plane, carries the core network signaling traffic. In 5G networks, *packet controlling, policy management, subscriber management, network resource management, signaling, and location management* are handled in the control plane [102] (See Fig. 2.2).

Following, we introduce the existing architecture of the cellular network, based on the latest generation (i.e., 5G) provided in Service-based Architecture (SBA), and its network functions in the control plane [103]:

i. *Access and Mobility Management Function (AMF)* network function is in charge of user registration, connection, and mobility management (e.g., periodic registration, and handover). *Registration* is an attachment procedure using standard signaling protocols in the core network to introduce the User Equipment (UE) to the system. Note that, when the UE does not use the network, its Radio Resource Control (RRC) state is `inactive`. To make it `active`, registration is required. The first registration is called "initial registration" (e.g., when

[4]3GPP is an organization that combines seven standard organizations in the field of mobile telecommunications to develop protocols in this environment [101]

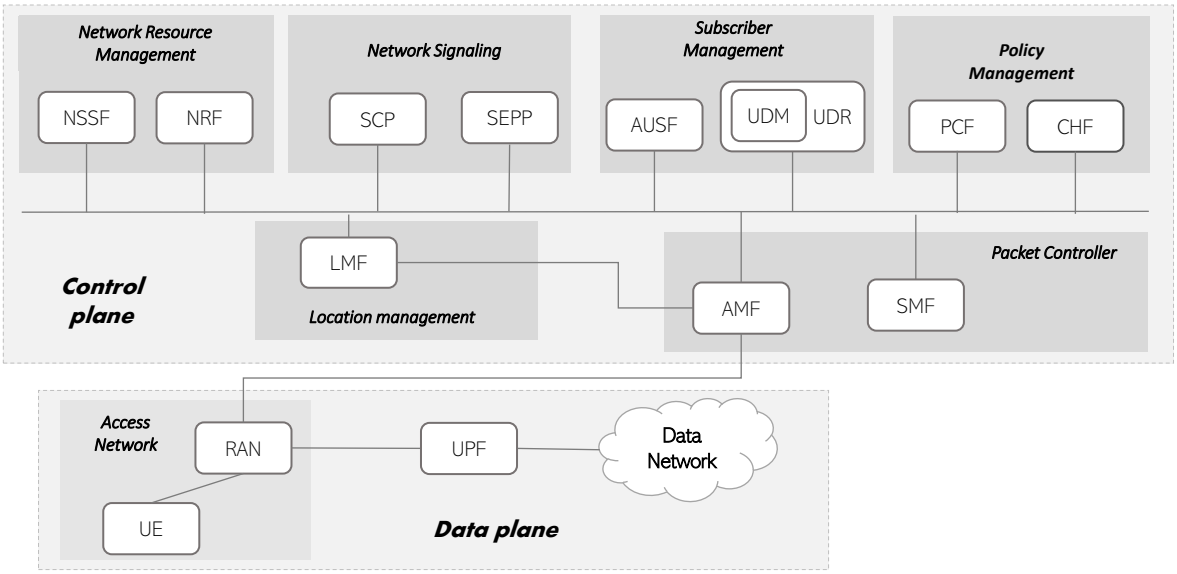


Figure 2.2: Service-Based Architecture of 5G cellular network

the user powers on the phone). In addition, periodic and mobility registration are other types of registration. Authentication and authorization of the UE and setting up a proper communication tunnel are the main operations in this step [9]. When a user moves through the network coverage area, its connection is transferred from one Base Station (BS) to another. This process is called *handover* in which the user sends the periodic measurement report to the corresponding BS (i.e., which is sent to AMF). The responsible AMF measures the Uplink/ Downlink transmission latency to decide on the initialization of a handover.

ii. *Session Management Function (SMF)* is responsible for execution of session management procedure. After registration or while handover, the session management establishes new Protocol Data Unit (PDU) session (or updates the state of the previous session) for the UE by allocating new IP (or keeping the previous one), allocating new resources, establishing a new tunnel, and release the previous resources and tunnels.

iii. *User Plane Function (UPF)* is an external point for PDU sessions to interconnect to DN (e.g., Internet). Packet routing and forwarding, Quality of Service (QoS) handling, and downlink packet buffering are some of its functionality.

iv. *Policy Control Function (PCF)* provides policy rules to satisfy the user's QoS requirement.

v. *Unified Data Repository (UDR) and Unified Data Management (UDM),* UDR serves as a centralized data repository in each MNO for the subscription data, user profile, policies, and application data [104, 105]. Other network functions including UDM, PCF, and Network Exposure Function (NEF) are connected to this component using standard APIs to fetch, update, and delete data. Subscription data and the user's profile -which are stored in UDR- include Subscription Permanent Identifier (SUPI), user's authentication data, long-term symmetric keys stored in user USIM, SIM identities, user PII data, etc. Indeed, the user's subscription credentials are confidentially stored in UDR [106]. These data are used by UDM to manage the user's registration procedure, authentication, access control, and mobility management [107]. SUPI as the most important stored PII data for the user, can not be transmitted in plain text. So, In many procedures (rather than emergency services), the Subscription Concealed Identifier (SUCI), will be used. SUCI is a privacy-preserved SUPI that is generated by the user via an Elliptic Curve Integrated Encryption Scheme (ECIES) based algorithm using the Home Network's public key [108, 109].

vi. *Authentication Server Function (AUSF)* supports authentication and authorization of the UE.

2.4 Authentication and Access control methods

This section provides a comprehensive overview of AAC methods. Authentication is the act of verifying that the subjects (i.e., someone/something that wants to use a resource) are the ones that they claim to be and that they are known by the system. Access control (or authorization) is defined as the process of accepting or denying the access request of a subject that has been verified from the authentication step to a specific object (i.e., resources that the subject wants to use) [110]. In other words, access control regulates which action a legitimate user can perform on resources in a computing environment. Note that the access control process is done after successful authentication.

2.4.1 Overview of Authentication

Authentication is a security mechanism that verifies who is the client sending the request and that they are the users they claim to be [110]. This process is accomplished through the following steps: 1) A registered user sends a request by providing the required data; 2) The authentication server records the complete log of the connection request; 3) The authentication server compares the received data with the stored identification in a database (verification), and 4) If the data match, the verification is successful, and the user can log into the system (e.g., by providing a login solution). Generally, in authentication methods, the only entity who would be authenticated is the user (i.e., when the servers don't authenticate themselves to the users). This model is vulnerable to several types of network attacks such as Man-in-the-Middle (MitM), Spoofing, and Phishing. *Mutual authentication* can overcome such attacks. This is a two-way authentication method in which both entities (i.e., user and server) authenticate each other. It can improve the security of a system by minimizing fraud and false verification.

Authentication procedures can be implemented in different ways; The most well-known examples are listed below [111]:

- ***Knowledge-based***: These methods rely on the users' *knowledge* about specific questions, such as identities (IDs) and passwords, PIN codes, lock combinations, and passphrases (i.e., something known only to a single user).
- ***Possession-based***: These methods operate based on *something that the user possesses*; this could be their credentials, a Radio Frequency Identification (RFID) card, a Universal Synchronous Bus (USB) token, or any other identifier that only the original user could have. The user's physical or logical location (i.e., where they are) is another identifier in this category.
- ***Biometric-based***: These methods rely on one or more physical features of the user and are also termed *Inherence-based* authentication. A user's unique physical and biometric features (e.g., fingerprints, facial or vein recognition, iris data) are used for this identity verification [112]. Generally, biometric authentication is more secure than other one-factor types [113].
- ***Multi-factor authentication***: This method combines two or more different solutions to make the authentication more secure. For example, a user may enter her password and a security code sent by SMS to her phone. Note that,

Table 2.3: Key Definitions in Access Control

Component	*Definition*
Subject	Someone or something that aims to access resources; can be a user, a computer process, a file, or a program.
Object	The resource that a subject wants to have access to, such as data, an IoT device, storage, or a network.
Rule/ Policy	The set of rights and regulations to follow by a subject to grant access to an object.
Owner	Someone or something that has complete control of an object and that can define the privileges for its access.
Operations/ Action	The acts and functions that can be done on an object by a subject (e.g., read, write, or execute).
Permissions	The authorizations of a subject to perform some specific actions on an object.

using the same method twice is not a multi-factor authentication, because the same type of attack can breach both. The main idea behind using multi-factor authentication is that breaching two or more different factors requires two or more different types of attack [114].

2.4.2 Overview of access control methods

Access control regulates who or what can perform which action on a network's resources [115]. Table 2.3 summarizes the definitions used in the rest of the section to provide consistent terminology. For a clear understanding of the access control procedure, the most significant steps (i.e., policy/ rule definition, access verification, and access log record) are listed in Table 2.4.

While there are several different access control mechanisms, the most well-known methods, including Discretionary Access Control (DAC), Capability-based Access Control (CapBAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC), and Attribute-Based Access Control (ABAC), are described in the rest of this subsection. Note that due to the large quantity of existing access control models, presenting and analyzing all of them is out of the scope of this paper. Therefore, only those methods used in the investigated articles are listed as follows (note that since we uused ABAC method in our proposal, this solution is explained with more details):

Table 2.4: Main Steps in Access Control

Step	***Functionality***
Policy/ rule definition	The step determines the rules of accessing an object. Each rule definition is varied based on the access control model. For example, rules can be defined in a matrix structure based on the object or as if-else statements in the XACML language [116]. Policies can be specified by the organization, manager, or owner.
Access verification	The access control server examines the received access request based on a subject's permissions. If they match, an access solution based on the enforcement method (e.g., access token or credential) will be assigned to the subject. For example, in CapBAC, the access tokens are assigned to the user.
Recording access logs	An access control system should record the logs of all activities (e.g., requests, accesses, actions, active sessions).

- ***Discretionary Access Control***: Proposed by Lampson in 1974 [117], DAC considers the owner-based administration of objects. More precisely, the owner of an object defines the access rules and policies of that object. DAC can be implemented via an Access Control List (ACL) in an access control matrix [118]. The ACL (i.e., the rows of an access control matrix) define which objects can be accessed by which subject with what type of permission (see Fig. 2.3).

- ***Capability-based Access Control***: In this model, a capability is associated with each subject [119]. The capability list is all the data in the columns of the access control matrix and is defined based on the subjects and the corresponding objects and permissions. A simple explanation of the difference between an ACL and a capability list is shown in Fig. 2.3. In the CapBAC model, users are granted access permissions based on an access token, such as a key, a ticket, a credential, etc. [120]. When a system aims to manage a large number of assets, CapBAC and DAC decrease the manageability (i.e., in the case of a change in the system, all of the access matrices should be updated) [121].

- ***Mandatory Access Control***: This model is also known as *Lattice-Based Access Control (LBAC)* [122]. MAC works based on the classification of objects and subjects. The most well-known security classification labels are "Top Secret", "Secret", "Confidential" and "Unclassified". In this model, the subject whose level's label is higher than that of the object can have access to that

object. The access decision in this method is made by a central authority and not by the owner.

– ***Role-Based Access Control***: Developed to resolve the maintainability challenge of matrix-based methods, RBAC manages a subject's access based on their role within the system, and also defines what kind of accesses are allowed to the subject of a given role [123]. When users' roles are changed in the system, their permissions will be altered and reassigned based on their new job. This feature simplifies permission management. Due to the nature of this access control model, a limited number of roles can represent many users, and it becomes easier to audit which users have which sort of permissions and what permissions have been granted to a given user [118].

– ***Attribute-Based Access Control***: The attribute-based access control method has four sets of attributes to define the access policy and manage the subject's access to the object. These sets are Subject Attributes (SA), Object Attributes (OA), Environment Attributes (EA), and Action Attributes (AA). Let define all the attributes (AT) of access policy as follows:

$$AT = (SA, OA, EA, AA) \tag{2.1}$$

These attributes can be selected based on their relevance to the received request. The four types of attributes in ABAC are [110, 124]:

  * *Subject attributes:* the identifiers that specify the subject (e.g., user roles, certifications, management-level information, and user IDs);
  * *Object attributes:* that distinguish the resources that the subject wants to access (e.g., file name, folder specification, application name, or ID);
  * *Action attributes:* specify the operations that a subject can perform on an object (e.g., read, write, execute and view); and
  * *Environment attributes:* details that describe the context in which the access is requested (e.g., the time and location of the requester, the type of communication channel).

each set of attributes are defined in below:

$$\begin{aligned} SA = \{s_1, s_2, \ldots, s_n\}, OA = \{o_1, o_2, \ldots, o_m\}, \\ EA = \{e_1, e_2, \ldots, e_p\}, AA = \{a_1, a_2, \ldots, a_q\} \end{aligned} \tag{2.2}$$

where $n = |SA|$, $m = |OA|$, $p = |EA|$ and $q = |AA|$. Each attribute in ABAC is defined as a pair $(attribute\_name, value)$. The request of the subject u to access a resource can be formulated as equation (2.3). To shorten the formulation, we avoid expanding each attribute set.

$$Req_u = \{SA_r, OA_r, EA_r, AA_r\} \tag{2.3}$$

Different validators in a system may need a subset of the attributes to validate the subject to perform a specific action based on the access policy. Let's define the attribute subset for validator v as:

$$AT_v = \{SA_v, OA_v, EA_v, AA_v\} \tag{2.4}$$

The validation result for each attribute set based on the predefined access policy is as (2.5):

$$V_{SA} = \begin{cases} 1, & \text{if } SA_v = SA_r, \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

The validation process for OA, EA, and AA sets is the same as (2.5).

Finally, access control result (AR), based on the policies defined by the owner, which is returned as "allow" or "deny" to the user, can be formulated as (2.6):

$$AR = \begin{cases} 1(allow), & \text{if } V_{SA} = V_{OA} = V_{EA} = V_{AA} = 1 \\ 0(deny), & \text{otherwise} \end{cases} \tag{2.6}$$

ABAC method is especially useful for fine-grained access control [125]. Fine-grained access control is an access control system that facilitates granting differential access rights to a set of subjects and provides flexibility in defining the access rules for users [125].

2.4.3 Main security attacks on AAC

Several types of attacks can target AAC procedures. Some of the well-known attacks and their solutions in centralized systems are described below [126–129]:

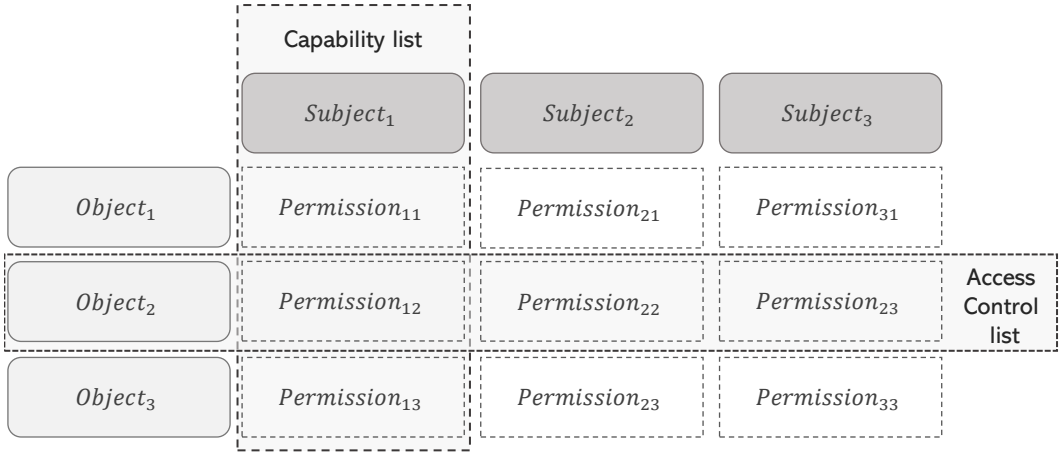


Figure 2.3: Access Control List vs. Capability List

– *Password cracking:* Attackers try to find the passwords and identifications of legitimate users by recovering them from storage. The most well-known attacks in this category are brute-force (checks all possible answers), rainbow (generates the password hash table in advance), and dictionary (uses a sample dataset of the most-used passwords). These attacks can result in privilege acceleration in the access control procedure. Some of the general solutions to protect users against these types of attacks are 1) using another authentication solution, rather than knowledge-based; 2) using multi-factor authentication; 3) employing account locking, in which the account would be locked after a pre-set number of unsuccessful login attempts; 4) initiating delayed response, in which the AAC server would return the result with a short delay, thereby preventing an attacker from checking all possible entries; and 5) oblige the user to choose a strong password with a specific standard/best practices, such as [130], which make it very difficult and time-consuming to crack passwords.

– *Denial-of-Service (DoS)/ Distributed Denial-of-Service (DDoS):* The purpose of these attacks is to make a resource unavailable for legitimate users. If one attacker tries to make a resource out of service, a DoS attack has been executed, but in the more complicated type of DoS, a large number of sources attack the destinations (i.e., a DDoS attack). Several well-known authentication methods, such as Kerberos, are vulnerable to this type of attack [131]. Request flooding, ping of death, and SYN flood are well-known DoS/DDoS attacks [132]. Although the detection and mitigation of DoS/DDoS are challenging tasks, some well-known techniques are being deployed in centralized systems. For instance:

1) using security solutions such as firewalls, Intrusion Detection System (IDS), and Intrusion Prevention Systems (IPS) to separate normal and abnormal traffic via different techniques; and 2) using redundant services to minimize the impact of an attack.

- *MitM:* The attacker relays information on behalf of the connection between source and destination, without their knowledge, and can alter, modify, or eavesdrop on their data [133]. Another form of this type of attack is the *reply* attacks, in which the attacker stores the user's identity data and uses that for subsequent connections [134]. Using SSL/TLS connections, providing mutual authentication [127], and adding timestamp and nonce to packets [135] are three of the most popular solutions to mitigate MitM/replay attacks in centralized systems.

- *Sybil:* In this type of attack, the attacker will define multiple virtual identities to target a network [136]. It means a single malicious node manages to influence the whole system using different identities. Two of the more common solutions are those using trusted certificates, and resource testing [137]. The latter ensures that the resources are matched with the number of users by different identities.

- *Spoofing:* in this type of attack, the attackers impersonate another identity in the system, aiming to steal data, get access to the network, accelerate their privilege, or launch other malicious activities [138]. Using multi-factor and mutual authentication are two of the preventive solutions to this attack.

2.4.4 Blockchain-based Authentication and Access Control methods

Due to its unique features (e.g., immutability, non-repudiation, traceability, and distributed nature), Blockchain can bring many unprecedented opportunities in AAC procedure. Moreover, Table 3.2 lists the possible solutions delivered by Blockchain to mitigate the attacks targeting AAC methods (mentioned in 2.4.3). This table lists the Blockchain feature that provides the possible solution as well.

The hierarchical architecture of Blockchain-based AAC solutions regarding different networking technologies and use cases is represented in Fig. 2.4. Note that, the Blockchain layers in this figure are based on [50, 152–154]. This architecture consists

Table 2.5: How Blockchain and Smart Contract Transform AAC

Feature	***Problem***	***Solution***
Immutability	Log or data falsification in AAC systems	– No one can change the access or authentication logs in the system. So, Blockchain can be served as a secure and immutable database for logs. – Storing credentials, identities, access policies, and rules in the Blockchain can make data alteration and falsification almost impossible.
Decentralized nature	Central database compromise	– Blockchain eliminates the requirement of a central database. So, the system will not be vulnerable to the threats such as compromising the database.
	Single point of failure	– In Blockchain-based systems, thanks to consensus for transaction validation, there is no need to have a central authority (i.e., Blockchain eliminates a single point of failure from a system). – Eliminates the risk of DDoS attacks on central servers and improves availability.
Reaching Consensus	Lack of data integrity and ordering	– After reaching a consensus, all nodes have the same ledger and the same order of transactions. It means the data integrity in the system, after several released blocks, is guaranteed.
Traceability/ Transparency	Non-auditability of access and login logs	– As all the transactions in Blockchain are validated and recorded with a timestamp, it is possible to verify and trace the previous transactions and logs [67]. – In case of malicious activity, root cause detection will be more feasible.
Non-repudiation	Action denial	– User's signature is required at each transaction. So, no one can deny their action in the system. Signing a transaction is its encryption via the private key. So, it is expected that only the user itself can sign a transaction.
Permanence	Data loss/ low availability	– This feature optimizes the advantages of decentralized nature and immutability. Elimination of the single database and removing any other single point of failure increase the system's availability and excludes the risk of data loss.

Table 2.6: DLT solution for the main attacks on AAC

Attack	*Existing solutions*	*Disadvantages*	*DLT-based solution(s)*
Password crack	password-less/multi-factor authentication	Identifiers are stored in a central database, managed by central authorities [127, 129, 139].	I) Password-less authentication along with distributed database for certifications eliminates the security problems of a central database [129]. II) Microsoft's ION [140] and Bitcoin's Identity protocol [139, 141], aim to provide a secure identifier. III) Self-sovereign identity [142] is an alternative to central model of identity management.
	Account locking after wrong login entries	Result in the locking of a legitimate account by an attacker	
	Delayed server response to slow down attackers	In large systems will result in high latency	
	Strong password	Not user-friendly	
(D)DoS	Using firewalls, IDSs, and IPS, and learning from attacks to avoid similar patterns.	Can be ineffective because of the growing complexity and novelty of attacks.	I) The distributed nature of DLT removes the single point of failure [129, 139]. II) The limited request generation rate in DLT makes DoS/DDoS attacks ineffective [143].
	Using redundant services to minimize the impact.	This solution changes the centralized architecture to a decentralized one.	
MitM	Using SSL/TLS	SSL/TLS assumption is on the trustworthiness of the certificate issuer. If no, the user receives a warning, and if they ignore it, MitM is possible [144].	I) The user's signature on transactions and the block time-stamp [145, 146]. II) Owing to immutability, certificates can not be altered [127, 147, 148]
	Mutual authentication		
	Using timestamp and nonce [135]		
Spoofing	Multi-factor authentication	The same problem of a central database in password cracking attacks.	I) The immutability of blocks guarantees genuine user identity; II) The user's signature on transactions inoculate the system against these attack [149].
	mutual authentication	The same vulnerability of certificate trustworthiness in MitM attack	
Sybil	using trusted certificates	Depends on the trustworthiness of a central authority (same as MitM attack)	I) This attack is highly complex since it needs the more adversary nodes than Byzantine Fault Tolerance (BFT) [150]. II) The blocks are traceable [151], and an abnormal increase in the chain size indicates the attack.
	Matching the resources with the number of unique identities [137].	Is not a solution to eliminating these attacks (it is a detection solution); Some studies show this method is ineffective [136].	

of six layers: *1) application layer, 2) authentication and access control layer, 3) contract layer, 4) consensus layer, 5) network layer, and 6) data layer.*

The lowest layer in the architecture is the *data layer* that encapsulates the underlying block structure. Above that, the *network layer* includes the mechanisms of distributed networking, data propagation and communication, and data verification. This layer distributes, forwards, and verifies Blockchain transactions based on predefined structures (e.g., transaction verification via digital signature based on asymmetric cryptography). Next, the *consensus layer* mainly focuses on the consensus protocol of the nodes in the network (e.g., PoW, PBFT, PoS). These algorithms can have an incentive mechanism to encourage the nodes to collaborate in the network. The *contract layer* relates to the solutions working based on smart contracts (i.e., not just based on transactions). This layer brings programmability into Blockchain and contains different languages to handle smart contracts. Two top layers in the architecture, *Authentication and Access Control and Application*, are related to the application of Blockchain in the desired context. The authentication and access control layer aims to implement different Blockchain-based AAC solutions for a variety of use cases as follows: (Note that, exploiting Blockchain in the AAC procedure can be influenced by the requirements of the specific application)

- ***Internet of Things (IoT)***: The application of Blockchain-based AAC in IoT include, but are not limited to, network security, mobility management in Wireless Sensor Networks (WSN)s through different clusters, provide secure access of the end-users to the sensor's data in smart homes and smart cities. To deliver more reliable AAC solutions and to decrease the processing load of resource-limited IoT devices, the combination of IoT and fog computing is proposed in several works. In these methods, fog nodes (having enough power and computational capabilities) can process data on behalf of the IoT nodes, connect to the Blockchain, and send their data to this network.

- ***Cloud computing***: Blockchain and cloud computing as two new advanced technologies have a high potential for strengthening performance, security, and privacy in current Web-based applications [155]. Authentication and access control via Blockchain in the cloud environment has been targeted by different works aiming to improve network security, first-level access of users to their cloud account, sharing the resources of the cloud computing environment such as computing power and memory, accessing the logs of resource sharing (to improve transparency and accountability), and data sharing in the cloud

environment. Worth mentioning that several methods exploit the cloud environment as a base technology for sharing and storing an enormous amount of critical data (such as electronic health records) and use Blockchain-based AAC to strengthen the security and privacy of data sharing.

- ***Telecommunication and cellular networks***: The existing AAC solutions in communication networks such as cellular networks are Authentication and Key Agreement (AKA)-based solutions. To resolve the challenges of existing AKA-based AAC in cellular networks, many researchers combine this technology with Blockchain. The recent studies intend to deliver the following services in cellular networks using Blockchain-based AAC methods: 1) mobility management among different service and network providers, 2) providing self-organized access to the network, 3) enabling medium access control by replacing new solutions with other existing methods such as Aloha [156], 4) network resource sharing, 5) providing Blockchain-based user connections to the Wi-Fi access points instead of knowledge-based authentication, and 6) generating Blockchain-based unique identity for users.

- ***Smart Healthcare***: Smart Healthcare is involved with all type of technologies (e.g., IoT sensors) that leads to better diagnosis of disease and sufficient treatment for patients. One of the challenging parts of this approach is to manage the electronic health records of the patients in a secure manner. So, several methods propose DLT-based AAC solutions to provide overall security in the network to store patients' data, to share the patient's health records, with proper doctors, health agencies, and research departments along with preserving their privacy, and to manage access to the patient's records.

- ***Information-Centric Networking (ICN)***: The majority of current networks have a host-centric model in which, the communication is based on named hosts, for example, web servers, PCs, and laptops. In contrast, ICN is a connection-less pull-based communication model that aims to distribute the content in a highly scalable and efficient way via named data objects, such as web pages, videos, documents, or other pieces of information [157, 158]. Based on our research, the existing Blockchain-based AAC models in ICN, are not only focused on protecting network security but also targeting producer mobility management (i.e., the challenging part is that ICN focuses on the named-based resolution mechanism) [159].

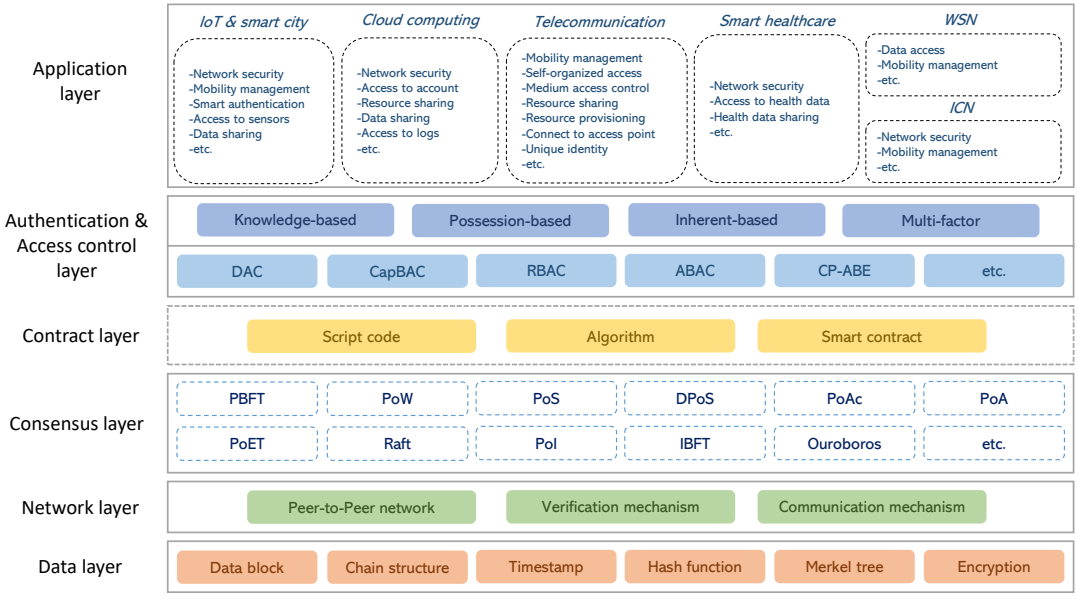


Figure 2.4: Blockchain-based AAC methods in networking applications

2.5 Summary

In this section, the necessary background about Distributed Ledger Technologies, their features and advantages, consensus models, and different deployment models are provided. Moreover, the ecosystem and architecture of cellular networks and their network functions are reviewed. Finally, authentication and access control methods are represented to have a comprehensive background about all concepts that will be used in the further chapters.

Chapter 3

Survey on decentralized architecture for MNO processes

Contents

3.1 Introduction

As described in the motivating scenario in Section 1.2, each actor of the cellular networks would benefit from different opportunities. The *end-users* can have access to many services using a single identity and assuring high privacy, by directly paying the providers. Moreover, they would have access to cellular and data networks without geographical limitations. The *connectivity providers* would be able to quickly enter the cellular network market and get the core network services from MNOs while paying them for core network service and getting paid by users based on their provided services. The *service providers* are able to provide a variety of services on top of existing core network services such as AAC. Moreover, they can provide services to the MNOs. Finally, *MNOs* can serve more users without the need to construct the whole infrastructure. Moreover, they can outsource many services to overcome the complexity of IT operations and contract management on large scale.

Several requirements are derived from this motivating scenario (i.e., broad coverage, distributed ecosystem, trustworthiness, automation, scalability, security, privacy, and compatibility) that are introduced in Section 1.3.

Surveying the existing solutions to address the requirements in Section 1.3, is the main contribution of this chapter. To do so, to the best of our knowledge, there is no comprehensive solution to address the requirements, so we categorized our survey into the following subsections. Indeed, providing authentication, access control; seamless mobility of the users are the main functionalities of the cellular networks to provide connectivity [160]:

i. Analysis of the existing decentralized cellular network architecture and their compatibility with our requirements.

ii. Analysis of the existing solution for mobility management in the cellular network and their compatibility with our requirements.

iii. Analysis of the existing identity management solutions and their compatibility with our requirements.

iv. A comprehensive study on the Distribute Ledger Technology (DLT)-based authentication and access control solutions.

3.2 Decentralized cellular network architecture

To the best of our knowledge, the concept of relying on a Blockchain-based architecture to enhance the whole MNO architecture design has not been investigated before. But, a very limited number of researchers proposed decentralized architecture for the whole cellular network ecosystem to provide a more open and competitive market for small-scale enterprises, private use cases, and end-users. The studies on this topic are not limited to Blockchain-based cellular networks.

For instance, Lou et al. [9, 160] proposed a decentralized cellular network to democratize the cellular access of the entities through a *broker* as a trusted party among the variety of RAN providers. In the other words in this alternate cellular architecture, a potentially large number of competing cellular providers can coexist. In their proposed method, part of the control plane functions- that implement standardized signaling protocols for communication with UEs- and the user plane functions- that implement packet forwarding, including classification and prioritization to enforce QoS levels, counters for accounting, etc.- is handled in MNO network. But, the defined 'management plane functions' that maintain subscriber information and perform authentication and policies would be handled either by the UE or outside of the MNO network in the Cloud.

A real-world implementation of an open and flexible system for building a low-cost wireless access network is provided by Magma open-source software platform [161]. Magma can provide access gateways that terminate the radio-specific protocols as close to the radios as possible. As a result, it allows carriers to augment an existing cellular deployment with WiFi hotspots in popular locations (e.g., athletic venues), or use LTE BSs to serve homes in rural areas, using a single core network and management platform [162].

Moreover, a long-range wide-area network (LoRaWAN) on Helium [163] provides a private wireless IoT network inside a public network using Helium Blockchain. In its novel announcement, the Helium network starts to step to 5G connections with its wireless modems [164]. Moreover, a Blockchain-based radio access network (B-RAN) has been proposed by Ling et al. [165] that uses Blockchain to handle the user's access to the RAN.

3.3 Blockchain-based Mobility management

Another critical function of the core network in the cellular network is the management of the user's mobility between cells and Base Stations (BS). Due to its immutability, and non-repudiation, Blockchain can be a proper candidate to eliminate the re-authentication overhead of the handover procedure, and increase the system performance. Due to these possibilities, several recent studies proposed Blockchain-based mobility management for existing 4G/5G networks.

Yazdinejad et al. [166] proposed an authentication method to decrease the number of unnecessary authentication during user handover in 5G networks. In this system, the Blockchain propagates the user's authenticity among other Software Defined Network (SDN) entities in the network. Using the trapdoor collision property of a chameleon hash function and the global availability and tamper-resistance of Blockchain, authors state that this system can achieve mutual authentication, key agreement, anonymity, traceability, robustness, perfect forward secrecy, master key forward secrecy, known randomness security, and universality. Moreover, Zhang et al. [167] proposed an authentication method for the seamless handover procedure in 5G networks. In this method, the user first registers in the network to insert a specific hash of the registration procedure into the Blockchain. Then, while the user is moving, this data would be used to manage the user's seamless connection. Conti et al. [159] proposed BlockAuth to enable mobility management in ICNs. This system consists of global and local clusters and their associated ledgers. After the registration of the user using an Authorization Server (AS), this data is stored in the global Blockchain. Next, the user sends the same data to the BS for validation. The BS verifies the user's identity from the AS. A single authentication server in this method can be a single point of failure. BCTrust [168] is another proposed method for mobility management in WSNs that aims to provide connectivity for WSNs in different clusters with only *one* authentication. To do so, the authentication controller (CPAN) stores the user's ID on the Blockchain. once it changed the cluster new CPAN sends a request to the Blockchain to see if the user has been authenticated before. Lee et al. [169], proposed another mobility management method in the 5G networks, as well. In this method, the authentication server sends the initial set of information to all BSs under its control. When a user joins one of the base stations, it sends the public key to the user and registers the connection in Blockchain. Then, the user sends her public key and timestamp to make connections, and then the BS sign and will broadcast these data.

3.4 Blockchain-based identity management methods

In this section, we will provide a review of existing Blockchain-based identity management solutions. Since recent identity management methods on top of Blockchain, are mostly provide self-sovereign identity as well as data management, in this subsection we will talk about these two components. uPort [170] provides the framework for users to gather attributes from an ecosystem of trust providers but does not provide identity proofing. For revocation in case of a key loss, Quorum Blockchain is used. It provides data ownership and selective disclosure however the privacy of user information in JSON data structure on the message server can be compromised. Jolocom [171] is another self-sovereign identity management that is also developed on top of Ethereum and provides similar functionalities to uPort. The Sovrin Foundation developed the Sovrin IdM [172] that uses attribute-based credentials which allow users to only reveal credentials that they choose with relying parties and WOT helps protect the user against deception. The adoption and integration of the Sovrin standard seem constructive in novel self-sovereign identity systems. In [173], the evaluation of uPort, Jolocom, and Sovrin shows shown that none of the existing systems fulfill the requirements of flexibility needs of digital identity for the heterogeneous online service. The shoCard [174] provides identity verification and as backup, it uses a stored encrypted version of the attribute certificate on the server. A central server is used as an intermediate between the user and relying on parties. In shoCard, the Bitcoin network records a commitment to personal data that was verified during identity proofing, and stores the hashes of certifications that are built upon the user's seal created by relying parties. The Blockstack [175] is another identity management system that attempts to redesign the naming system to provide elucidation of Identity. It has PKI authentication features using state machines and storage aspects in Blockchain to preserve privacy and resource identification.

3.5 Blockchain-based authentication and access control methods

The research on DLT-based AAC mechanisms can be categorized according to the four features shown in Fig. 3.1 and explained as follows:

- ***AAC mechanism***: Defines the authentication or access control type implemented in the studied work. Authentication types include Knowledge-based,

Possession-based, Biometric-based, and Multi-factor. Moreover, access control methods cover DAC, CapBAC, RBAC, and ABAC methods.

- ***DLT application approach***: According to our studies, we identified two general approaches for using DLT in AAC procedure:
  - Several studies use DLT as a distributed *database* to store credentials, identities, rules, roles, policies, and access logs. The main motivations of authors in these methods are the immutability, integrity, and permanence of DLT.
  - In the considerable portion of literature, the authors use DLT not only as a secure database but also as a *decision point* for AAC procedure (e.g., to manage the authentication process by creating and handling the tokens, to handle the client's access based on predefined policies, storing the access log). Note that, in rare cases, the authors used DLT only as a decision point, not a database. Generally, distributed nature of DLT, removing the single point of failure, non-repudiation, permanence, and having programmable contracts, are the main motivations of the authors in these works.
- ***In which step DLT is used***: In the authentication procedure, DLT is used in recording credentials/identifiers in the ledger, the verification step (i.e., to verify the user, server, or token), providing access solutions (e.g., token issuing), log management (i.e., storing identities or logs) or in several of these steps. In the access control procedure, Blockchain can be used for the following purposes:
  - *Distributed database:* storing access rules and policies (including as well access logs) into *bct and smart contracts as a tamper-proof solution;
  - *Policy modeling:* defining the access policies and rules using smart contracts;
  - *Verification method:* verification of the user's access request using smart contracts;
  - *Policy enforcement:* enforcement of access control decision to allow or deny the access of a user to the system.
- ***Use-cases***: The following use cases were identified for AAC methods in a networking context: communication networks, IoT devices, smart cities, smart healthcare, cloud computing, ICNs, and WSNs. Regardless of the application,

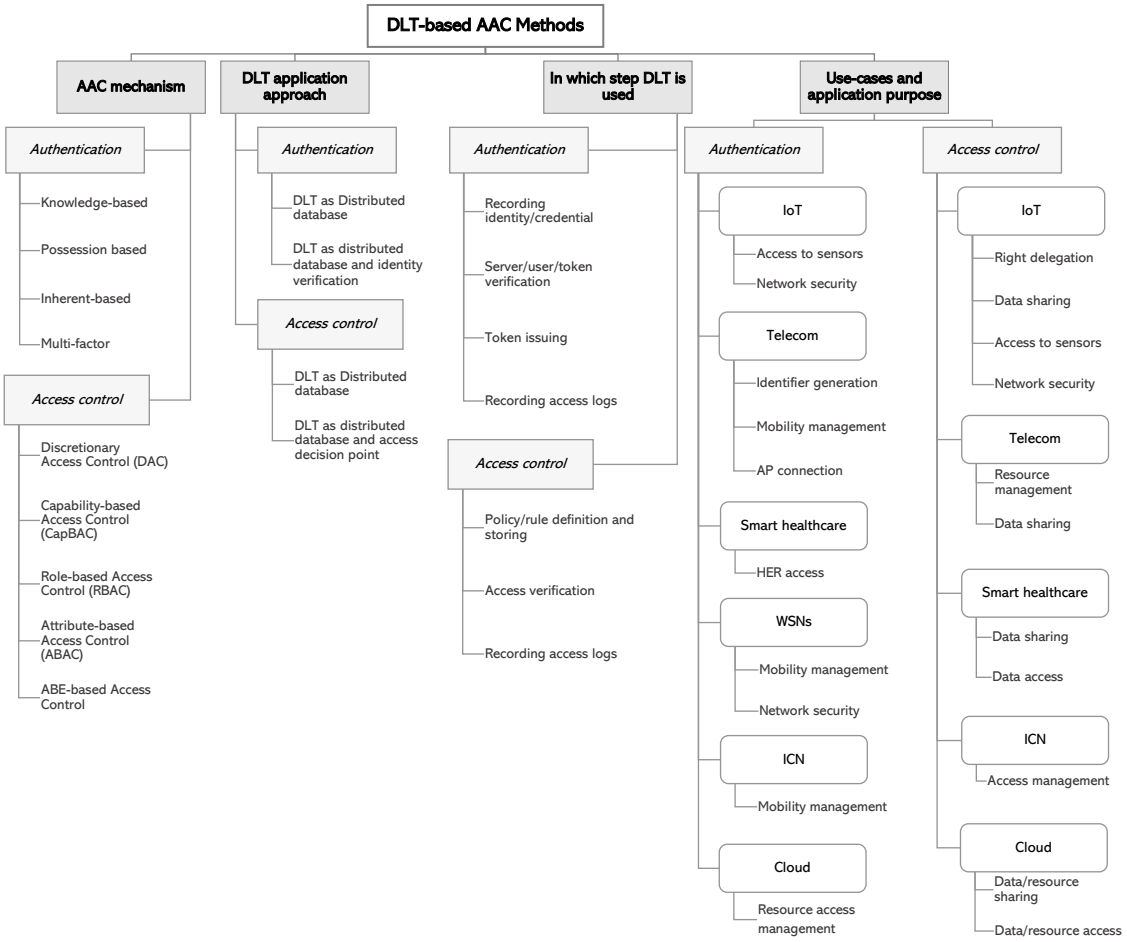


Figure 3.1: Taxonomy of existing AAC methods based on DLT.

some methods are general-purpose methods and thus can be used in all types of use cases.

3.5.1 DLT-based Authentication Methods

This section presents the current authentication methods that rely on DLT (mostly Blockchain and smart contracts). Firstly, these methods are divided based on their approach to using DLT and then their application use cases.

– ***DLT as a distributed Database***:

* *Methods for Telecommunication*: Lee et al. [176] proposed BIDaaS that generates a Blockchain-based ID for users (instead of conventional IDs in cellular networks), and then this ID is registered on the Blockchain for further mutual authentication process.
* *Methods for IoT/ smart cities*: Huh et al. [177] proposed an automatic door locking system via fingerprint-based authentication. The hash of a user's fingerprint is stored on the Blockchain and the users can authenticate themselves through mobile devices. [178] proposed a two-factor authentication method that uses an out-of-band channel to perform secondary authentication.
* *Methods for smart healthcare*: Mohsin et al. [179] proposed an authentication method using RFID and FV, in which a hybrid, random binary pattern of the user's FV and RFID is derived and stored on Blockchain, to be extracted in verification.
* *Methods for cloud computing*: several authentication methods aiming ***access management*** are proposed in cloud environment. Deep et al. [129] proposed a method to authenticate insider and outsider users. It checks the user's credentials and valid Blockchain node parameters to verify the user's identity. Another method, called SAMS [180], uses a master node to manage the security of the whole system. Before connection, the client creates a block and sends the nodes' and block's information to the master node. The master node creates a block with the received information to check the identity.

– ***DLT as a distributed database and verification solution***:

* *Methods for Telecommunication*: Xue et. al [181] proposed an authentication method to handle the user's movement in mobile vehicular networks. In this method, an intermediary smart contract is used to make a connection between foreign and home networks. The moving users receive a session key, which is also stored and managed by the smart contract, for further connections. Moreover, for access point connection, Sanda et al. [182] proposed a method in which the user installs "Auth-Wallet" to be verified by exchanging the "Auth-Coins" instead of her information. The user connects to the access point using its unique ID. The access point sends Auth-Coin to the user for verification and signing with the Bitcoin address. If the verification is successful, the token will be broadcasted to

the Blockchain and the user can be connected to the internet. Another system is proposed by Niu et al. [183] for Wi-Fi hotspot access. In the first step, the user requests a signature on Bitcoin address from the service provider by sending the real identity. Then, the digital signature would be sent to the user. Because the user's credentials are saved in the Blockchain when the user requests to connect to the network, the service provider and the Wi-Fi hotspot get valid credentials and provide the connection.

* *Methods for IoT/ smart cities*: Ourad et al. [147] proposed a smart contract-based solution, in which the Ethereum address is the authentication identifier. The smart contract broadcasts an access token to the sender's Ethereum address if validation of the user is successful, the user combines and signs several data and sends them to the IoT device to verify.

3.5.2 DLT-based Access Control Methods

This section presents the state of the arts in DLT-based access control methods. The categorization of the methods is the same as Section 3.5.1.

– ***DLT as a distributed Database***:

* *Methods for IoT/ smart cities*: Ali et al. [184]proposed an access control solution focusing on the right delegation in smart contracts (containing the delegation policies). BlendCAC [185] implemented a CapBAC method in which the smart contracts store the access control matrix. The main challenge of this method is that a subject cannot obtain rights from more than one subject. This challenge is addressed in [186]. Dramé-Maigné et al. [187] designed an ABAC solution in which administrators establish trust relationships for their devices, and the users deploy the attribute contract. When a user sends the access request, the target device connects to its gateway to retrieve attributes and evaluates the request against the policies. Another ABAC mechanism is proposed by Pinno et al. [188] by using four separated Blockchains to store several parameters, credentials, and relationships.
* *Methods for Smart healthcare*: Zhang et al. [189] proposed a hierarchical model for sharing the healthcare data. In this method, the Blockchain acts as a distributed ledger of permissioned clients to store and verify the keys and record the hash values of auditing logs.

* *Methods for cloud computing*: Qin et al. [190] proposed an ABAC method to share data in the cloud environment. In this system, a central authority manages the security of the whole system and issues an attribute key to the user and cloud service provider in the smart contract. Alansari et al. [191] proposed an ABAC method for cloud federation. In this system, federated cloud organizations can define attribute-based rules and store them in the Blockchain to provide fine-grained secure data sharing for the users.

– ***DLT as a distributed database and verification solution***:

* *Methods for Telecommunication*: Ling et al. [192–194], proposed Blockchain Radio Access Network (BRAN) as a solution to implement self-organized access for users and providers, along with enabling mobility management. In their recent work, Ling et al. [165] proposed a Blockchain-based medium access control method. Moreover, Fan et al. [195] proposed a data-sharing scheme for Cognitive Cellular Networks (CCN) in 5G. SBAC [196] aims to achieve hierarchical access by proposing an ABAC method for data sharing in ICN.
* *Methods for IoT/ smart cities*: Sultana et al. [197, 198] proposed a data sharing and access control system via smart contracts. A similar method is proposed by Zhang et al. [199]. Due to its flexibility, many researchers implemented *ABAC* solutions. For instance, Putra et al., [200] uses smart contracts for authorizing the nodes based on their reputation. Fabric-iot [201] uses three kinds of smart contracts to store the URL of resource data, manage and store ABAC policies, and implement an access control method for non-admin users. In the proposed method by Ding et al., [202], the owner of the IoT device sends access policies to the Blockchain. The user chooses a satisfied subset of the policies regarding her needs. Then the owner checks the requester's identity in the Blockchain and allows/denies the access request. Yutaka et al. [203] proposed a method that uses four smart contracts to perform access decisions. Tang et al. [204] proposed a cross-domain ABAC method. Apart ABAC method, Hwang et al. [205] proposed a dynamic RBAC scheme in which policy generation can be done dynamically by manager nodes and the policies stored in Blockchain. In another method, IoTChain [206], firstly, the owner creates a smart contract for her data with an access policy and sends it to the Blockchain. When a user asks for authorization Blockchain generates an access token.

Novo [207] introduced an access control method focusing on scalability and energy consumption via sending the access request to Blockchain through the closest management hub.

* *Methods for Smart healthcare*: Rajput et al. [208] proposed a Blockchain-based data sharing system, in which, after registration, the emergency doctors can retrieve the patient's data by sending access requests and the patient's ID via Blockchain and smart contract. Nguyen et al. [209] use Blockchain for sharing patient's healthcare data on the Internet of Medical Things (IoMT) networks. Li et al. [210] proposed a system based on certificate-less cryptography, in which, the data owner creates an ACL and then stores it in the Blockchain for further validation.

* *Methods for cloud computing*: Yang et al. [211] proposed AuthPrivacyChain in which policies and access logs are stored in Blockchain, and access management is done by the smart contract. PrivacyGuard [212] is another system that focuses on user and data privacy on the cloud. Owners can define their access policies in smart contracts. The user invokes the owner's contract to ask for permission, data access rules, and deposit payment. TBAC [213] is an ABAC solution for resource sharing that exploits four types of transactions to record the information of subjects and objects, send the access request, and access decision. Wang et al. [214] proposed a fine-grained access control for cloud storage. In this method, the owner deploys a smart contract to store the essential data of the file. To grant access, the owner defines the expiration time and a secret key and adds them to the smart contract.

Regarding the state-of-the-art in access control procedure, when Blockchain is used as a distributed database, even though the non-repudiation, rule immutability, etc., are improved in the proposed systems, they could suffer from a single point of failure, because of making and enforcing access control decisions by centralized parties. On the other hand, although using Blockchain for modeling access control policies, verification, and policy enforcement, can increase the complexity of implementation, marginally decrease the time efficiency, and increases the need for storage, they can provide high scalability (in terms of the number of users), availability, fault tolerance, the immutability of rule and decision, non-repudiation, and audibility. Fig. 3.2 compares the distributed and centralized implementations of the access control methods in radar diagrams to bring a general overview of their advantages and disadvantages.

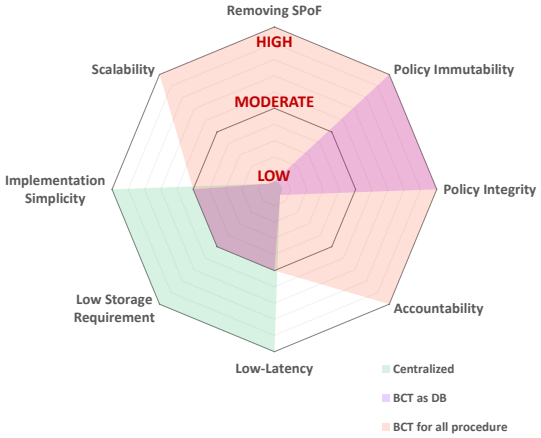


Figure 3.2: A comparison among different types of using Blockchain solutions addressing access control

Note that, a radar chart or spider chart is a visual tool to compare multiple variables on a two-dimensional plane. for this diagram different axes needed to be defined. In our case, one ax is the comparison parameters, and the other ax is either *low*, *moderate*, or *high*. The axes, in the Radar diagram, start from a common central point (in our case, the central point was *low*). Each point in the Radar indicates the rate of a particular parameter. As the result of the analysis of the Radar diagram, which use-case or entity which gains a bigger surface of the diagram, can be the more advantageous candidate to be chosen. As it is shown in the figure, using Blockchain as a distributed database, validation solution, and policy enforcement.

3.6 Summary

In this section, the related state of the arts regarding the usage of Blockchain technology in cellular networks or their dependent applications is provided. In this regard, several kinds of research on distributed cellular network architecture, Blockchain-based AAC methods, Blockchain-based mobility management methods, and Blockchain-based identity management methods are introduced. Note that, due to the abundance of works regarding Blockchain-based AAC methods, a brief and comprehensive summary and comparison of them are provided in Appendix B and Appendix C.

Generally, using DLT in cellular network applications/architecture can increase the

integrity of the data, accountability of the users, and the difficulty of data falsification regarding credentials, decrease the complexity of the IT procedure, removes the extra steps of AAC in mobility management, improves the data availability, etc. Having these advantages in mind, the proposed methods mostly suffer from high computational time, transaction fees, resource usage, and maintainability of smart contracts. Some studies use DLT as a database to store user credentials, which leads to inheriting the main problems of conventional centralized solutions. The other two significant challenges in these systems are the size of the blocks and required storage, as the system performance can be negatively influenced by an oversized chain [68].

The different existing methods of decentralized cellular network architecture, mobility management, access control, authentication, and identity management, and their compatibility with the different requirements mentioned in Section 1.3 are summarized in Table 3.1.

To sum up we can state that although providing different unprecedented opportunities (i.e., the ones mentioned in strengths and opportunities of using Blockchain) such as security and availability in cellular networks, several challenges such as the lack of openness for cellular networks, broader coverage, scalability for accepting new businesses to the entrance of small-scale competitors, the complexity of system implementation, automation and non-flexibility of the cellular networks based on the needs of different parties, remained intact.

Table 3.1: Existing methods of decentralized cellular network architecture, mobility management, access control, authentication, and identity management, and their compatibility with the different requirements

category	method	R1	R2	R3	R4	R5	R6
Decentralized core network	[9, 160]	P^*	×	✓	×	×	✓
	[161]	×	×	×	P^+	✓	✓
	[163]	×	✓	✓	P^+	✓	×
	[165]	×	N/A	×	P^{++}	✓	×
Mobility management[#]	[166]	✓	✓	×	N/A	✓	✓
	[167]	✓	✓	×	N/A	✓	✓
	[159]	✓	✓	×	×	✓	×
	[168]	✓	✓	×	×	✓	×
	[169]	✓	✓	×	×	✓	✓
Authentication and access control methods	Blockchain as database	✓	✓	×	×	N/A	✓
	Blockchain as database and verification solution	✓	✓	×	✓	N/A	✓
Self-sovereign Identity management solutions		✓	✓	N/A	N/A	✓	×[##]

P^* This method only provides higher automation in B2B contract management.

P^+ Magma and Helium are scalable for the users, but there is no specific feature for the new providers and small-scale businesses.

P^{++} This method brings higher automation in RAN-based resource allocation.

[#] All compatibility with requirements are limited to the mobility management (i.e., not for profile, identity, subscription, etc. management).

[##] All identified methods are in service level and not specific for the cellular network architecture. So, due to the requirement of the 3GPP standard, we can state these methods are not compliant with the standards.

Chapter 4

Blockchain-based Cellular network architecture for Beyond 5G and 6G

Contents

4.1 Introduction

As explained in section 1.4, addressing the requirements of motivating scenario (ref. section 1.3 and section 1.2), using the existing ecosystem of cellular network, and MNO architecture, would face several critical challenges such as stand-alone architecture that decrease the possibility of collaboration among entities, increase the operational and installation cost, environmental effect, complexity of handling the contracts, centralized nature which can limit the network scalability and performance, low innovation possibility because of a limited number of collaborators, etc. To address the existing problems and requirements, we proposed a new multi-actor mobile connectivity ecosystem benefiting from Blockchain technology depicted in Fig. 4.1. In this ecosystem, the suppliers (i.e., TowerCos and InfraCos) and businesses (i.e., service/content providers) and MNOs can collaborate to provide different services for the end users broader than their current business (i.e., regarding regulation, services, incentives, micropayments, etc.).

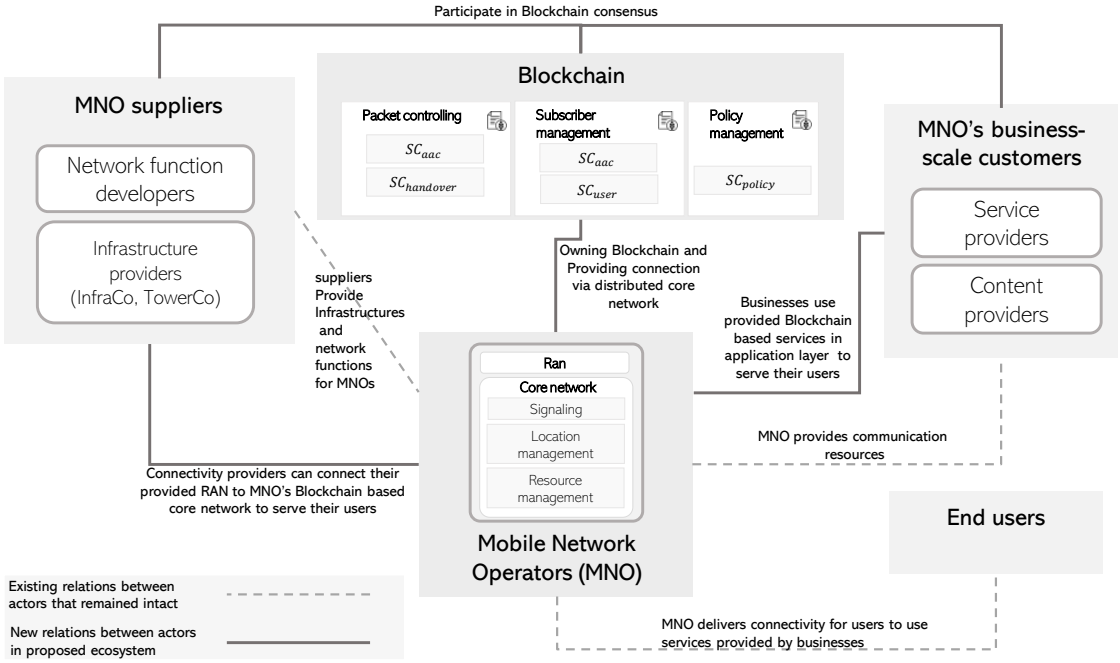


Figure 4.1: The proposed ecosystem for MNOs

First, focusing on Blockchain entity in the ecosystem, three functionalities of current cellular networks, namely, packet controlling, subscriber management, and policy management, are migrated to Blockchain. In high abstract, these functions provide

registration, authentication, access control, mobility management, identity management, and session management in cellular networks. So, migration of each entity to Blockchain gives this opportunity to the other entities of the ecosystem to participate, use, or collaborate in providing or executing different functionalities in the network. For instance, the MNO suppliers such as connectivity providers can execute the user registration procedure (i.e., AKA procedure in current cellular networks) to serve their users. On the other hand, the provided mobility management functionality, as well as access control procedure, on top of Blockchain, can be used by businesses and service providers to outsource their security services to a distributed system.

It is important to mention that this proposed ecosystem does not remove any collaboration possibility in the existing ecosystem. The main purpose of the proposed ecosystem is to improve the existing collaboration, provide more opportunities to the entities, and reduce the existing challenges. So, for instance, MNOs will have their own infrastructure, services, and Blockchain-based core networks; but on top of that, they can benefit from having new business models to serve more users, businesses, and vendors.

The main contributions of the following sub-sections are listed below:

- Analysis on how the proposed ecosystem and cellular network architecture can address the provided requirements in Section 1.3.
- An overview of the proposed architecture for MNOs on top of Blockchain and smart contracts targeting beyond 5G and 6G.

4.2 How the proposed architecture addresses the requirements

We list the possible answers of the proposed ecosystem and architecture to the listed requirements in section 1.3:

R1 ***Higher automation***: To address the ever-growing complexity of collaboration, providing new services and connections in next-generation networks, providing higher automation is an inevitable requirement. In our proposed ecosystem, several functionalities are handled by smart contracts (rather than the manual procedure in the current ecosystem) that can increase automation in the system.

For instance, managing the collaboration with the entities (i.e., in business to business scale) is handled by smart contracts in which the new entity (e.g., connectivity provider) can create their unique smart contracts with the agreement of the MNO(s) to provide service for the users. In this process, the connectivity provider needs to provide some pre-defined information for authorization and then the registration procedure with greed prices will be handled automatically using smart contracts. Looking at R1, we mentioned that one of the obstacles in providing broader coverage is the complexity of manually handling the huge number of contracts. Using the previously explained procedure, this issue is also addressed.

As another example, the users can add/remove any services from their unique smart contract in the system, without any requirement for further IT operation in other entities (i.e., all revocation procedure is validated and handled by smart contracts). Finally, the user's profile and phone number portability is another instance of this automation in which we can remove the existing centralized entity which manually handles the user's porting from donor MNO to recipient MNO.

Moreover, tracing the engagement between entities, deploying the agreements and their termination is a costly procedure for the MNOs. The collaboration processes are managed by smart contracts in the proposed system, and thanks to their traceability, accountability, immutability, and permanence, the contract management can be handled more efficiently and automated.

R2 ***Trustworthiness***: One of the crucial challenges in collaboration with different entities to provide connectivity and services is to provide trust among entities. In the current ecosystem, trust is provided by centralized entities. Referring to R2, one of the requirements for the realization of the motivating scenario is to provide a distributed ecosystem. So, in this regard, we face two conflicting requirements. Note that, in our use case, we define trust as guaranteeing the entities' privacy and payment, data/information integrity and correctness, and validity of the requests (i.e., access will be provided to the entities if they are eligible).

R3 ***Distributed/ decentralized ecosystem***: Migrating the non-trivial functionalities of the core network, such as registration, access management, identity management, and mobility management, that are currently handled by a central authority (i.e., MNOs) to the Blockchain, brings high-level of decentralization

in the ecosystem. From the technical viewpoint, instead of executing different functionalities with one/several limited numbers of network functions, in this proposed architecture, all nodes of all entities are running pre-defined functions in smart contracts as a distributed code available for all entities. So, the system availability is expected to be higher than the centralized architecture.

Using Blockchain, smart contracts, and their intrinsic features, we address these two conflicting requirements to provide trust in distrustful environments without the need for any centralized party. To this aim, Blockchain technology, not only uses the entity's private key to sign the requests and provide non-repudiation but also, the consensus procedure in the system guarantees the correctness and validity of a transaction based on the agreement of all/majority of the network entities.

R4 ***Scalability***: The scalability requirement, can be examined from two different viewpoints as follows:

* System scalability regarding the increasing number of collaborators: for the entrance of different collaborators we refer to R1-R4. This part of the requirement can be fully addressed by the proposed system and the current maturity of Blockchain technology.
* System scalability regarding the increasing number of users: scalability challenge with Blockchain emerges mostly when the number of users/nodes increases. This problem results in increasing transaction validation and consensus convergence time. This aspect of the scalability is discussed in Chapter 9.

R5 ***Security and Privacy***: These requirements are addressed in several aspects of the proposed ecosystem as listed below:

* Benefiting from the intrinsic public-key infrastructure in Blockchain for the registration procedure and mutual authentication of the users and network entities. This procedure not only provides highly secure authentication but also preserves the user's privacy due to the anonymity of the Blockchain addresses in the system.
* Key-agreement procedure in the registration and mobility management is handled by the diffie-hellman algorithm between the user and network entity.

- Storing the users' PII data outside of the Blockchain in other distributed databases such as IPFS. To provide security and integrity of these data, we proposed a hybrid cryptosystem. Using this method, we store the encrypted version of the access keys to the user's data inside their smart contract. Since the keys are encrypted, they are only accessible by legitimate entities in the system.
- System resistance against several attacks is also discussed in Chapter 8.

R6 ***Compatibility***: To address the compatibility issue of the Blockchain-based core network architecture we proposed three steps (i.e., three implementation scenarios) in which the network elements can be migrated to the Blockchain in different levels. In the first step, the application-level services such as access control for service providers, profile management, and number portability can be migrated to Blockchain-based solutions with full compliance with the standards. In the second scenario, the network entities can be connected to Blockchain-based functionalities, and deployed in smart contracts, through designed gateways that are able to parse the network functions' messages to the smart contract functions and vice versa. This step is also compliant with the existing standards, but it provides the possibility of proposing bottle-neck at the point of gateways and increasing the signaling messages in the network. The third scenario is a clean-slate proposal for the beyond 5G, in which the network functions are fully migrated into the Blockchain and different smart contracts play the role of current network functions in the system. Indeed the latter one is not compliant with the 3GPP standard.

Apart from addressing the requirements, **the proposed method can provide a solution for broadening the coverage of cellular networks while not increasing the MNO's investment**, by sharing the existing infrastructure and mutualizing the costs. As mentioned earlier, providing broad coverage using the current technologies, is a hideous expense for the MNOs, which results in limiting connectivity provisioning in more feasible places with a higher population. This can guarantee the MNOs' revenue and their money back based on their investment in different geographical sites. In the current MNO architecture, due to the lack of scalability and trustworthiness, providing broader coverage needs either MNO's expense to install infrastructure or to have a contract with other large-scale third parties (let's call it a connectivity provider) with an acceptable reputation to provide connectivity for the user's in the remote area. The ever-growing number of users and their specific needs make

this procedure almost infeasible for the next generation networks. Migration of user subscription, identity management, and registration procedure to the Blockchain-based system, as well as providing the collaboration possibility between connectivity providers and MNOs, gives this opportunity for the connectivity providers to use the distributed identity management system for applying the same authentication, key agreement, and registration procedure for the users. On the other hand, since the entities in Blockchain (in our scenario the network entities) are using their signature for transaction submission in the system, it provides the required trust in the system without any centralized third party. For instance, let's explain a win-win scenario in this regard. Assume that a small group of people is living in an unpopulated rural area that is a white spot for MNOs (in which the cellular network antenna can not provide appropriate and complete coverage). Providing RAN and antenna with high initial expense, for a licensed spectrum, for these areas with a low population is not beneficial for the MNOs. As a result, user satisfaction would decrease. In this case, a small group of users (or small-scale suppliers) who supply the RAN, can help to serve better coverage for the users in these areas. In the proposed architecture, after providing the RAN part, the suppliers can register in the system to be able to use the distributed core network functions. So, they can serve their users with inferior initial expenses and reasonable prices. As a result, all parties (MNO, users, and small-scale suppliers) would benefit.

4.3 Overview on Blockchain-based core network architecture

In this section, we provide an overview of the proposed Blockchain-based core network architecture. As mentioned before, in the conventional MNO architecture, the user plane (including RAN) and control plane are handled by mobile operators (usually in a centralized approach). To address the business-related and technical challenges of the existing system and to provide the aforementioned advantages (see Chapter 1), we combined decentralized and distributed solutions to introduce a semi-distributed core network architecture. Decentralization of the functionalities is mostly about providing collaboration opportunities and managing the network with more than one entity, and distribution is regarding the execution of the functions in different Blockchain nodes. A simple overview of the proposed architecture based on 5G SBA is illustrated in Fig. 4.2. It is important to note that the proposed system is a novel approach for designing 6G core network architecture and Fig. 4.2 only aims to

provide a comprehensible overview of the logical positioning of network functions on top of existing 5G Service-Based Architecture.

As shown in Fig. 4.2, the network infrastructure and functions of the proposed approach and their positioning are as follows:

- *Access Network or User plane*: similar to conventional mobile network architecture, the main elements of access networks consist of RAN, UE, and UPF:
  - *Network infrastructure:* The RAN infrastructure can be a radio access device provided by MNO or other suppliers, vendors, and small-scale providers. As mentioned before, the RAN functionality and its operations will remain intact in the proposed method, so this subject is out of the scope of this work.
  - *User Equipment (UE):* The main entity that needs to be changed in the proposed method is the User Equipment (UE). Since the proposed architecture is totally different than the existing architecture of 5G (as well as the previous generations). So, to fully support all its functionalities of it a brand new network architecture, a new type of USIM or e-SIM, and compatible standards are required beyond 5G networks. The new generation of SIM-card needs to store, at least, a shared master key hard-coded in the SIM card, the user's Blockchain wallet information that consists of her Blockchain address, and public/private key pair.
  - *User Plane Function (UPF):* This function has also remained intact in the proposed model. This network function is responsible to manage the user's data forwarding from the core network to the external data network.
- *Core network functions:* Some core network functions such as location management, signaling, and network resource management -that are responsible to find the user's exact position, handling the interconnection among different network functions, and managing/optimizing the network resource consumption, respectively- are proposed to remain in MNO's core network. The main reasons for keeping them in the core network are 1) High packet overhead (e.g., for signaling and recourse management) that can overload the Blockchain's required storage in participating entities and 2) execution of near real-time AI/ML-based algorithm for resource management in which the Blockchain does not have a high contribution.

- *Blockchain-based core network functions:* Several network functions, such as policy management, subscriber management, and packet controlling are proposed to migrate to Blockchain and be handled by smart contracts. The Blockchain in this architecture can be a consortium among the actors and entities in the network. In this work, we are using Ethereum Blockchain with smart contracts on Solidity language to implement and run the following functions:

  * *Policy management* to store, update, and revoke the access policies and handle the billing procedure.
  * *Subscriber management* to authenticate the user in the registration step. (e.g., when the user power on the phone). Moreover, the users' anonymous identity and their updated location and subscription data are kept in the Blockchain. Note that, to resolve the storage challenges introduced by Blockchain and to provide user privacy, we propose not to store all data in Blockchain and we use shared and distributed databases such as InterPlanetary File System (IPFS) to confidentially store users' PII data.
  * *Packet controlling* to authorize the user based on access policies, to propagate the user's identity to avoid re-authentication while handover, and to update the user's location and access parameters.

  Note that, the consensus mechanism for the proposed method can be based on the agreement of the actors in the system. In our implementation, we used the PoW method.

- *Distributed database* such as IPFS to record the user's PII data in a confidential manner, connection history, etc. IPFS is a distributed file storage and sharing platform relying on Distributed Hash Table (DHT) to identify its contents [215]. This storage system allows direct interaction through a secure and global Peer-to-Peer (P2P) network [216]. Once uploading a file in IPFS, this platform split it into the chunks of $256KB$ as `IPFS Objects`, and as a final chunk, it generates an empty object which links all the other objects of the file [217]. Every chunk is identified by a cryptographic hash, also named content identifier, that is computed from its content [218]. IPFS uses Merkle Directed Acyclic Graph (DAG) data structure to link the content and the objects together. **Storing the user's subscription data outside of the Blockchain can increase the availability of the data, decrease the storage requirement, and improve data privacy.**

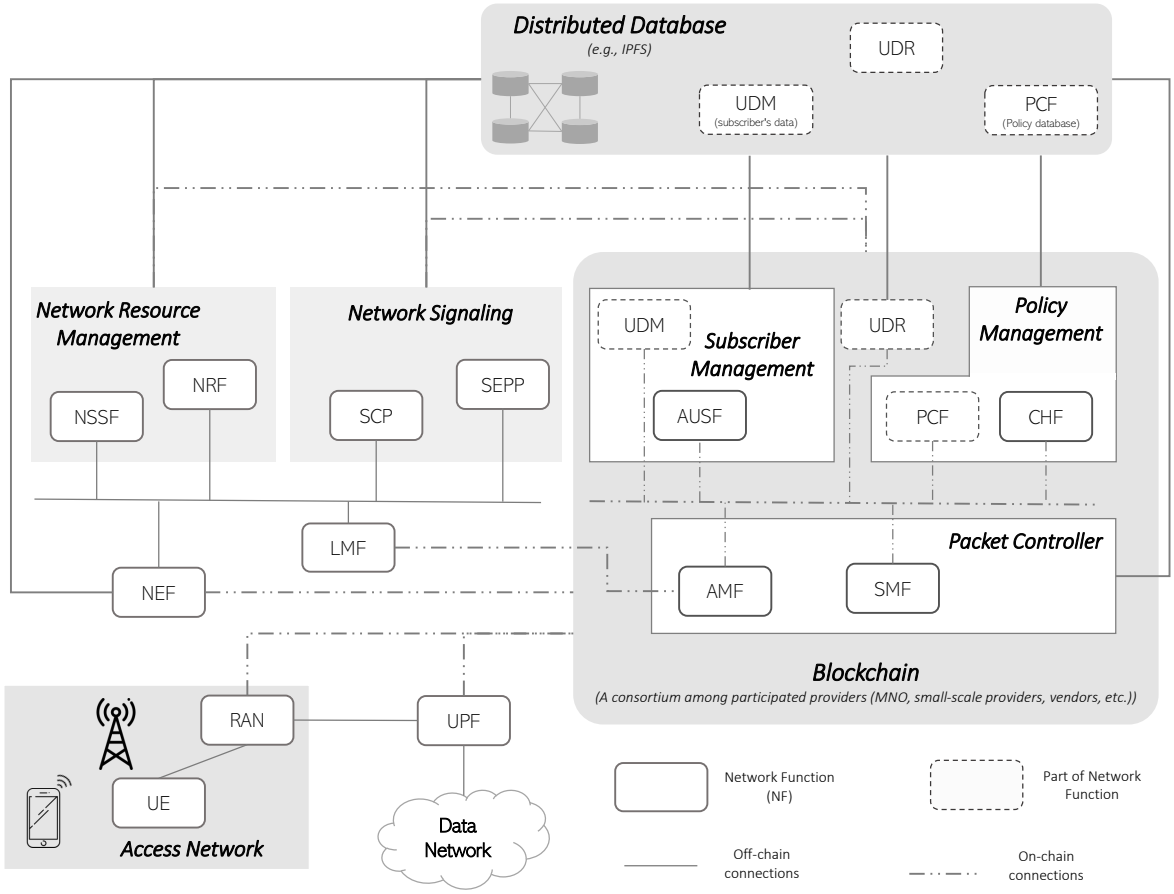


Figure 4.2: The High-level overview of proposed core network architecture for MNOs.

It is necessary to mention that we assume that the Blockchain-based core network is not directly accessible to the users (similar to the user's access level in the conventional systems). Moreover, in this model, an unlimited number of service providers (e.g., internet-based service providers such as video streaming or IP-based calls), and suppliers (e.g., a group of people or small enterprises who provided the RAN) can benefit from the functionalities of core network and serve their particular users.

Note that, our general assumptions for deploying the system are listed below:

- In all use-cases we assume that off-chain connections (those which occur outside of the Blockchain network) are secure connections, e.g., using Secure Socket Layer/Transport Layer Security (SSL/TLS), HTTPS, etc.
- The users' equipment supports e-SIM in which the user's Blockchain address and public/private key pair are hard-coded.

- All MNOs participating in the Blockchain is authorized by national regulatory bodies (i.e., the regulatory body is responsible verify the authenticity and eligibility of the MNOs).
- In some use cases (e.g., Mobile number portability), the regulatory bodies are the owner of some smart contracts (the details will be mentioned in the following subsections).

Note that, all these assumptions have been taken into account while designing the different procedures unless it is clearly mentioned that we had other assumptions for that exceptional use case.

4.4 Summary

This section provided a high-level description of the proposed ecosystem of cellular networks. In this regard, we analyzed how the proposed ecosystem can fully or partially address the requirements of broad coverage in the next-generation networks, providing the distribution in the system while addressing the trustworthiness among the users and all other entities in the network, introducing higher automation in a different procedure, scalability in different aspects, security, privacy, and compliance with the existing standards and architecture.

Finally, this section provides a high-level proposed architecture for the core network. To put this new architecture in the existing concept, we provided a mapping between different entities with the SOA model of 3GPP in 5G.

Chapter 5

Designed smart contracts for Blockchain-based Cellular network

Contents

5.1 Introduction

To implement the proposed architecture (depicted in Fig. 4.2) we designed several smart contracts on solidity language running on top of Ethereum Blockchain to handle different functionalities. As mentioned before, the concept of smart contracts refers to automated agreements among mutually distrusting parties, without the need for a trusted intermediary. The user can request the execution of a smart contract by sending transactions to the distributed ledger. Potential conflicts in the execution of transactions and their validation also get handled through the distributed ledger, using its associated consensus protocol.

In this section, we provide the description and data model of the designed smart contracts. These contracts serve several purposes in the system to handle different functionalities of core network or service-level use cases. The physical data model [219, 220] of the designed smart contract is depicted in Fig. 5.1.

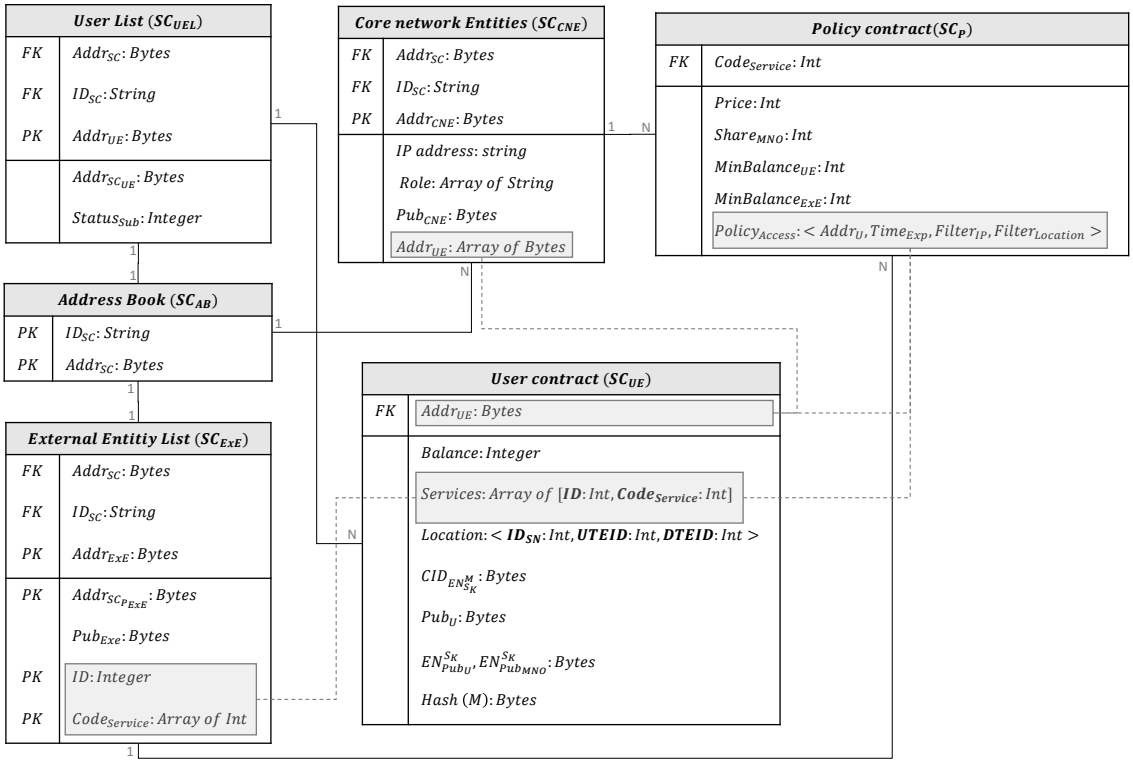


Figure 5.1: The physical data model of designed smart contracts and their relations.

Note that, to have consistent terminology and symbols, utilized abbreviations, smart contract names, and symbols are listed in Table 5.1.

Table 5.1: Symbols and their descriptions

Symbol	***Description***
	Symbols and Abbreviations
$Addr_x$	x's address in Blockchain
CID_x	Content Identifier (CID) of data x in IPFS
$Entry_{AB}$	Structure of mapping stored in SC_{AB}
$Entry_{UL}$	Structure of mapping stored in SC_{UL}
$Entry_{CNE}$	Structure of mapping stored in SC_{CNE}
$Entry_{OR}$	Structure of mapping stored in SC_{OR}
DE_x^k	The decrypted value of x by key k
EN_x^k	The encrypted value of x by key k
Pub_x	x's public key in Blockchain
Pr_x	x's private key in Blockchain
$Policy_{Price}$	The policies related to the service price
$Policy_{access}$	The policies related to the ABAC attributed to the user's access to the service
$Role_x$	The Role of subject x in the network
K_M	The master symmetric key of registration
α	Prime number in Diffie-Hellman algorithm
ϱ	Prime root of α in Diffie-Hellman algorithm
X_s	Diffie-Hellman generated private key
Y_s	Diffie-Hellman generated public key
	Abbreviations for smart contracts
SC_{AB}	Address Book smart contract
SC_U	Subscriber's specific smart contract
SC_{UL}	User List smart contract
SC_{ExE}	External Entity list smart contract
SC_{Sub}	Subscription management smart contract
SC_{HO}	Handover management smart contract
SC_{AAC}	AAC management smart contract
SC_{MNO}	MNO's specific smart contract
SC_{MNO}	MNO list smart contract
SC_P	Policy smart contract
SC_{CNE}	Core network Entities smart contract
SC_{Port}	Mobile number and profile porting smart contract
SC_{Reg}	Registration manager smart contract

5.2 Reference smart contracts

Reference smart contracts refer to the contracts those are generally used as distributed database to store different types of information in immutable, integrate and

Table 5.2: Identifiers of Address Book contract

Identifier	**Parameter**	**Address of**
'OwnerRole'	$Addr_{SC_{OR}}$	Address of Owner Roles smart contract
'Sub'	$Addr_{SC_{Sub}}$	Address of Subscription smart contract
'Reg'	$Addr_{SC_{Reg}}$	Address of Registration smart contract
'AAC'	$Addr_{SC_{AAC}}$	Address of AAC smart contract
'UL'	$AAddr_{SC_{UL}}$	Address of User List smart contract
'ExE'	$Addr_{SC_{ExE}}$	Address of External Entities smart contract
'MNOL'	$Addr_{SC_{MNOL}}$	Address of MNO List smart contract
'CNE'	$Addr_{SC_{CNE}}$	Address of Core Network Entities smart contract
'HO'	$Addr_{SC_{HO}}$	Address of Handover handler smart contract

transparent manner. Note that, it is very important not to store any confidential or PII data in these contracts. Following, the smart contracts of this category are introduced.

5.2.1 Address Book contract (SC_{AB})

To store a mapping of the identifier to the address of single smart contacts (i.e., SC_{Sub}, SC_{Reg}, SC_{AAC}, SC_{UL}, SC_{ExE}, SC_{CNE}) to make their collaboration more secure. This contract maps the predefined contract identifier (i.e., their name or code) to their addresses, as:

$$Entry_{AB} \xleftarrow{ID_{SC}} Addr_{SC}$$

where ID_{SC} is a predefined unique identifier for smart contracts (see Table. 5.2 in which the identifiers of contracts are provided), and $Addr_{SC}$ is its address in Blockchain. Note that the purposes of designing this contract are 1) avoiding the use of hard coded addresses to evade maintainability defects of smart contracts [221], 2) having a list of addresses to manage and implement the modifiers in functions to benefit from intrinsic access control capability of smart contracts, and 3) providing more secure collaboration among them and avoiding data falsification of function calls by forged smart contracts advertised to the users by an attacker.

5.2.2 Owner Roles contract (SC_{OR})

In order to support different use cases and scenarios in the system, different roles can execute different actions in the system. For instance, MNO can modify the user's unique smart contract (if it is the user's host MNO), but the regulatory body can not modify it, and it is only eligible to read some part of that. Another example is the regulatory bodies' right to add new MNO addresses to the system, this right is only dedicated to regulatory bodies.

To address this requirement to propose a comprehensive solution, Owner Roles contract stores a mapping of each owner's (or in another work entity's) identifier (i.e., Blockchain address) to its role in the system to provide the possibility o implementing RBAC model. The mapping is as follows:

$$Entry_{OR} \xleftarrow{Addr_S} Role_S$$

where $Addr_S$ is the Blockchain address of the subject who wants to execute a function in the Blockchain and $Role_S$ is its role in the system. The roles are defined as 5.3.

5.2.3 User List contract (SC_{UL})

This smart contract stores the list of users who are registered in the system. The users can be subscribed in one or several MNOs, services, etc. The user list is stored in a mapping with the following structure:

$$Entry_{UL} \xleftarrow{Addr_U} [Addr_{SC_U}, Status_{sub}]$$

where $Addr_{SC_U}$ is the user's unique smart contract in the Blockchain and $Status_{sub}$ defines the user's subscription status as active/non-active. $Status_{sub} == 0$ indicates that the user has not been activated her SIM-card, $Status_{sub} == 1$ means that the activation is progressing and $Status_{sub} == 2$ shows that the user has been activated her SIM-card and is able to get services from MNO or other external connectivity providers based on access policies defined for her.

5.2.4 Core network entities contract (SC_{CNE})

Core network entities smart contract is designed to store the list and addresses of core network entities -belonging to MNO- based on their Blockchain address. Note

Table 5.3: Owners in the system and their Roles

Role	Description	capabilities
'MNO'	The entity with given Blockchain address is identified as MNO	– Updating its user's data in IPFS and their smart contract; – Adding/ removing the user's to/ from user list smart contract
'RB'	The entity with given Blockchain address is identified as Regulatory body	– Updating the user's clearance data in their smart contract; – Adding/ removing the authorized MNOs to/ from MNO list smart contract;
'SP'	The entity with given Blockchain address is identified as a service provider	– Using the service level functions provided for the external providers such as AAC, identity management, etc.
'CP'	The entity with given Blockchain address is identified as a connectivity (RAN) provider	– Using AAC services for user registration register; – Using handover function to provide mobility management for the users;

that, each entity in core network is logically assumed as a node in Blockchain. The entries in this smart contract is a mapping with, at least, the following structure:

$$Entry_{CNE} \xleftarrow{Addr_{CNE}} [IP, role, Pub_{CNE}, Addr_U[]]$$

where $Addr_{CNE}$ is the entities' Blockchain address, IP is its IP address for network layer off-chain connections, $role$ is the entities role/ functionality in the core network, Pub_{CNE} is its public key, and $Addr_U[]$ is the list of the user's who are connected to the entity (if applicable). Note that, due to the low latency of Blockchain for data retrieval [10], finding the information of entities in the network based on their Blockchain address would be done faster.

5.2.5 External entity list contract (SC_{ExE})

This smart contract stores the list of all registered/subscribed service/connectivity providers who are participating in the network as an external entity (not MNO's

internal entities). These entities may provide connectivity (e.g., TowerCos, RAN providers, etc.) or services in service level. The entry of this contract is a mapping of entities' Blockchain addresses to its parameters with, at least, the following structure:

$$Entry_{ExE} \xleftarrow{Addr_{ExE}} [Pub_{ExE}, ID, Code_{service}[], Addr_{SC_{P_{ExE}}}]$$

where $Addr_{ExE}$ is the Blockchain address of the entity, ID is its unique identifier, and $Code_{service}$ is the list of all services provided by that particular entity (indicated as unique codes). For instance, some providers may provide both connectivity and call. Note that, we defined $Code_{service}$ to make it possible for the separation of cost and access policies for different services. $Addr_{SC_{P_{ExE}}}$ is the address of policy smart contract (i.e., SC_P) dedicated the the particular ExE (for more details about this contract see Section 5.3).

5.2.6 MNO smart contract (SC_{MNO})

MNO smart contract is a unique smart contract for each MNO, deployed by a regulatory body, at the time of their registration in the system and after their validation. This contract stores, at least, MNO's current subscribers and the list of user subscription/port requests.

5.2.7 MNO list smart contract (SC_{MNOL})

The MNO list smart contract is designed as a distributed database to store the list of all authenticated MNOs, currently, national-wide. This contract is owned by the regulatory body that keeps the MNO information in the following structure:

$$Entry_{MNOL} \xleftarrow{Code_{MNO}} [Addr_{MNO}, Addr_{SC_{MNO}}]$$

5.3 Policy management smart contracts

These smart contracts aim to store and apply the defined policies by MNO, regulatory bodies, service providers, connectivity providers, etc. In this category, there is an unique smart contract named by **Policy contract (SC_P)** deployed *for each*

service/connectivity provider, µO, or any other registered external entity (in agreement with MNO). This contract stores two main types of policies (i.e., $Policy_{price}$ and $Policy_{access}$) with the following structures:

i. $Policy_{price}$ stores the price list of all the services provided by the particular service provider or µO (based on $Code_{service}$). These policies are stored in the following structure:

$$Policy_{price} \xleftarrow{Code_{service}} [Price, share_{MNO}, MinBalance_U, MinBalance_{ExE}].$$

Where $Price$ is the minimum service price defined in a variety of granularity based on the service type (e.g., seconds, minutes, MB, GB, etc.), $share_{MNO}$ is the MNO's share/benefit from serving the service (in percentage), $MinBalance_U$ is the minimum required user balance to guarantee the user's payment to the external entity or MNO (if the service is directly provided by MNO), and $MinBalance_{ExE}$ is the minimum required balance of external entity by which the MNO can be assured that it would be able to pay the service price instead of the user.

ii. $Policy_{access}$ records the user's access control attributes based on the Attribute-Based Access Control (ABAC) model [125] in the following structure:

$$Policy_{access} \xleftarrow{Addr_U} [Time_{exp}, Filter_{IP}, Filter_{location}]$$

where $Time_{exp}$ is the expiration time of the user's access (this policy is for the application layer services, such as video streaming). $Filter_{IP}$ and $Filter_{location}$ are some environmental variables that can deny users access to/from specific locations. Note that, policies can be vary based on the service type.

5.4 Subscriber management smart contracts

In this category, the smart contracts are responsible for the user's subscription, authentication, access, control, or storage of the user's state in the network. The subscriber management smart contracts are categorized as follows.

5.4.1 Subscription contract (SC_{Sub})

The subscription smart contract is dedicated to handling the user subscription procedure in the host MNO. After receiving and validating the user's Subscription request

(similar to the user's SIM-card activation procedure in the current cellular network), this contract activates/deploys the user's unique smart contract (SC_U) and updates her status in SC_{UL}. Note that the details of the subscription procedure will be described in the next subsections. Moreover, the other important function of this smart contract is to delegate/revoke the ownership of the user's contract to the host MNO for updating her data in IPFS or SC_U (e.g., when the user switches between MNOs, the recipient MNO needs to have the ownership of `update()` function in SC_U, while the ownership of donor MNO needs to be revoked).

5.4.2 Port management smart contract (SC_{port})

One of the Blockchain-based services that can be provided by MNO for B5G is to manage the Mobile Number Porting (MNP) procedure. SC_{port} is designed for this purpose to handle the *porting process or termination* of the user's subscription. To port the user, after validating the user's request, this contract removes the user from the donor MNO-specific smart contract (SC_{DNO}) and adds her into the recipient MNO's contract (SC_{RNO}). Note that, SC_{DNO} and SC_{RNO} are the same as SC_{MNO}; here we used two naming to make them identical. Moreover, this contract delegates the ownership of the user's data to RNO. In the termination procedure, this smart contract removes the user from SC_{MNO} and SC_{UL} and destroys SC_u.

5.4.3 User contract (SC_U)

A user smart contract is a unique smart contract deployed for a particular user which stores, at least, the following attributes by mapping the user's Blockchain address to her identifiers:

$$Attr_U \xleftarrow{Addr_U} [Balance, Services[], Location < ID_{SN}, UTEID, DTEID >, \\ CID_{EN^M_{K_s}}, EN^{K_s}_{Pub_u}, EN^{K_s}_{Pub_{MNO}}, Hash(M), Number_U, Status_{legal}]$$

where:

- *Balance* is the user's current balance in her Blockchain wallet;
- *Services*[] is a list of the user's subscriber *service-layer* services (e.g., the example of real-world services can be Skype, Netflix, etc.);

- *Location* is a tuple of parameters indicating the user's current location to make MNO capable of finding the user. Note that, this location is not the precise geographical location, it is her approximate location based on her serving network. *Location* data contains but is not limited to the current Serving Network's Identifier (ID_{SN}) and Uplink/Downlink PDU session ID ($UTEID, DTEID$).
- $CID_{EN^M_{K_s}}$ is the access identifier of IPFS storage. Here, M is the user's identity, wrapped into a file. K_s is a symmetric key generated by MNO to encrypt user data;
- $EN^M_{K_s}$ is the user's data (M) encrypted by K_s;
- $Hash(M)$ is the hash of content M;
- $EN^{K_s}_{Pub_u}$ and $EN^{K_s}_{Pub_{MNO}}$ are the K_s encrypted by the user's and MNO's public key, respectively; and
- $Number_u$ is the dedicated phone number to the user.
- $Status_{legal}$ is the phone numbers legal status defined by regulatory body or MNO and stored as integer. Note that this status doesn't contain any confidential and private data, it is an identifier about legal status of the stored number, e.g., to assure that the number doesn't have any legal problem.

Note that, since in Blockchain and smart contract the data is transparent for everyone in the network, none of the user's PII data is stored in SC_U. These data are proposed to be stored confidentially in IPFS.

5.4.4 Authentication and Access Control smart contract (SC_{AAC})

AAC smart contract is a smart contract designed to manage the user's authentication and access control procedure by validating her request to access to the services, connectivity facilities (RAN), etc. against the stored access control attribute in $Policy_{price}$ and $Policy_{access}$. Moreover, this contract handles the *payment* procedure.

5.5 Packet controlling smart contracts

Packet-controlling smart contracts are responsible for managing the user's access to the network, i.e., the users initial and periodic registration in the network and

mobility management by handling the handover process.

5.5.1 Registration contract (SC_{Reg})

This smart contract manages the user registration procedure in the network to make the user known and traceable for the network. The registration procedure is done once the user turns her phone on, and also on a periodical basis.

5.5.2 Handover contract (SC_{HO})

A handover manager smart contract is a designed contract to store the list of handover requests and handle its procedure. The main goal of this contract is to propagate the user's handover request in Blockchain (as a transparent, immutable and trusted environment that provides non-repudiation of the user). This identity propagation makes the serving and target entities (i.e., RANs) able to eliminate the repetitive registration and authentication procedures while handover. In the other words, since the handover procedure is stored and signed by one of the trusted entities in the network in SC_{HO}, when the target host receives the connection request, it can identify that the user doesn't need for further *authentication* procedure.

SC_{HO} records, at least, the following data for handover, mutual authentication, and key agreement:

$$Data_{HO} \xleftarrow{ID_{Table}} [Addr_{CP_{serving}}, Addr_{CP_{target}}, Addr_U, Status_{HO}, \alpha, \varrho]$$

5.6 Summary

In this chapter, we provide a detailed description of the designed smart contracts to handle the different functionalities of the core network. In this regard the following categories of smart contracts are introduced:

- Reference smart contracts that are generally used to store the required data/information for the functionalities provided by the other smart contracts. For instance, the list of users in the system, the list of network entities and their access addresses, the list of smart contracts, etc.

- Policy management smart contracts to store the authentication and access control policies related to different application-level services or core networks.
- Subscriber management smart contracts that handle the user's subscription in the system, their access control procedure, authentication, etc.
- Packet controlling smart contracts to manage the user's registration and mobility in the network.

Chapter 6

Network functions of Blockchain-based core network and Application-level services

Contents

6.1 Introduction

To implement different functionalities of the proposed ecosystem for the cellular networks, we considered two different types of use cases that address different requirements. The first category, which can be considered as a first step to migrate some functionalities to Blockchain, provides several service-level applications such as access control in service provisioning, novel billing solutions, and mobile number and profile portability of the users in cellular networks.

The aforementioned contributions provide a broader view of the possibilities of using Blockchain in core network functionalities. So, in the second category of functionalities, the core network functions such as registration, mobility management, session management, policy management, and billing procedures are proposed to migrate to a Blockchain-based distributed system.

Table 6.1 depicts the compliance of each category with the defined requirements.

Table 6.1: Our contributions and their compatibility with the different requirements

category	*R1*	*R2*	*R3*	*R4*	*R5*	*R6*
Blockchain-based core network	✓	✓	✓	✓	✓	×
Blockchain-based application-level services	✓	✓	✓	✓	✓	✓

6.2 Blockchain-based network functions designed for B5G

To execute the main functionalities of the core network in the new architecture, the following network functions are introduced.

6.2.1 BC-SM: Blockchain-based Subscriber Management

To ensure the delivery of an authorized connection by the external providers, the first step is to register them in the network by MNO which would be done in an on-chain procedure (6.2.1.1). Moreover, the subscribers' conventional subscription procedure needs to support distributed profile management. The external entities' registration and the user's subscription and profile management procedures are described as follows.

6.2.1.1 External provider registration

As mentioned before, the main idea of the proposed method is to make different entities able to collaborate and provide an open market for external and small-scale providers to be able to benefit from infrastructure/core network entities provided by large-scale MNOs. In this regard, any service provider (SP), Micro-Operator (µO), and Connectivity Provider (CP) that aims to use the system needs to have an agreed smart contract with MNO. To do so, after reaching an agreement on the costs and prices (i.e., $Price$, $share_{MNO}$, and $MinBalance_{ExE}$), that can be through advertisement procedure, MNO would deploy the specific SC_P in the system and inserts its address and other related information in SC_{ExE}. Note that, $MinBalance_{ExE}$ is one of the required access policies to give the possibility of removing the payment procedure from MNO. To do this, the MNO requires to be assured that it would be paid by the external provider (instead of the user) after providing the connection. So, $MinBalance_{ExE}$ defines the minimum acceptable wallet balance of the external provider for MNO.

6.2.1.2 User subscription and Profile management

The subscription steps of the user (U) are as follows (See Fig. 6.1 and `Procedure 1`).

1 U sends a subscription request to SC_{Sub} by creating a transaction as: $< Code_{sim}, Hash(nonce), Addr_U >$

where $nonce$ is a random number generated by user, and $Hash(nonce)$ is the hash of $nonce$ calculated by $Keccak256$ [222] algorithm. $Code_{sim}$ is a secret code given to the user once she bought the SIM-Card, and $Addr_U$ is the user's Blockchain address hardcoded in the SIM-Card.

2 Once SC_{Sub} receives the request, stores $Hash(nonce)$ and asks SC_{UL} to verify the subscription status of $Addr_U$. If the user is not subscribed, SC_{UL} updates the user's $Status_{sub}$ to 1 that means `'verified for activation'`. Finally, SC_{UL} sends the transaction receipt to SC_{Sub} to send it to the user.

3 Once U is redirected to the subscription page of the MNO, she sends $< Tx - reciept, nonce >$. MNO can verify the request from SC_{Sub} by calling `Verify()` function of SC_{Sub} and sending $< Tx - reciept, nonce >$ to its arguments. If the following conditions pass, the `Verify()` function would return `true` as

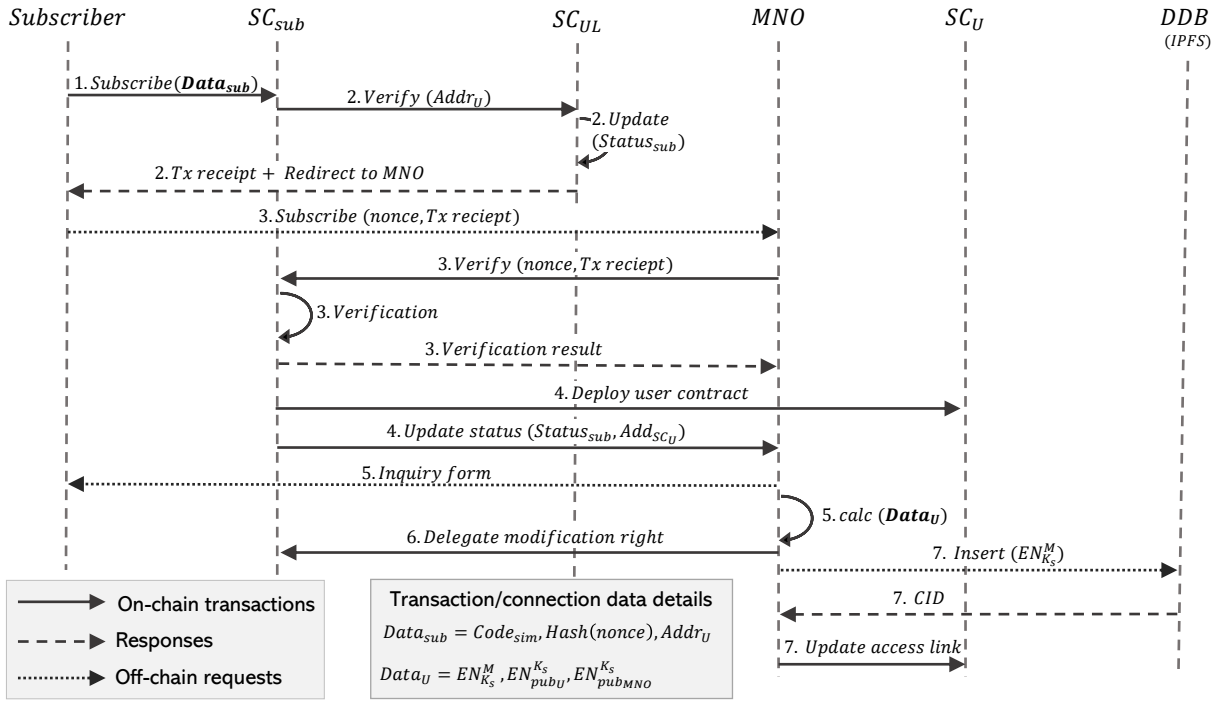


Figure 6.1: User subscription procedure

the indication of successful verification: $Hash^T(nonce) == Hash(nonce)$; $Status_{sub} == 1$

$Hash^T(nonce)$ is the hash of *nonce* received by MNO, and $Hash(nonce)$ has been stored in SC_{Sub} in *Step 2*.

4 SC_{Sub} deploys a unique smart contract for the user (i.e., SC_U) and changes $Status_{sub}$ to 2 that means `'subscribed-data`. Then, creates an event for MNO to confirm the user's subscription by sending $< Status_{sub}, Addr_{SC_U} >$. Moreover, it inserts $Addr_{SC_U}$ in SC_{UL}.

5 Once receiving the confirmation, MNO sends the subscription form to the user and receives the user's identity data (M). Because the data will be stored in a distributed database (i.e., IPFS), after receiving M, MNO needs to strictly limit the access to data. **Note that, using IPFS in the procedure is to address the scalability and storage requirements of Blockchain.** The only external entities that can have access to data are the U and MNO. To do so, we employed a hybrid cryptosystem for a multi-user environment. The hybrid cryptosystem is a technique of combining symmetric and asymmetric

Procedure 1: User subscription

1: $U \rightarrow SC_{Sub}$: **Subscription request:**
$< Code_{sim}, H_{nonce}, Addr_U >$
2: $SC_{Sub} \rightarrow SC_{UL}$: **Verification request:**
$< Code_{sim}, Addr_U >$
3: SC_{UL}: **if** $Status_{sub} == 0$ and $Code_{sim}$ *is valid*
Then $Status_{sub} = 1$ and $SC_{UL} \rightarrow SC_{Sub}$: *validated*
4: $SC_{Sub} \rightarrow U$: $< Tx\_receipt, subscriptionlink >$
5: $U \rightarrow MNO$: **Subscription request:**
$< nonce, Tx\_receipt >$
6: $MNO \rightarrow SC_{Sub}$: **Verification request:**
$< nonce, Tx\_receipt >$
7: SC_{Sub}: **if** $H'_{nonce} == H_{nonce_{Tx\_receipt}}$
Then $Status_{sub} = 2$
Then$SC_{Sub} \rightarrow SC_U$: **Deploy** user contract
Then$SC_{Sub} \rightarrow SC_{UL}$: **Add** $Addr_{SC_U}$
8: $SC_{Sub} \rightarrow MNO$: Valid subscription
9: $MNO \leftrightarrow U$: **Inquiry form for subscription** (M)
10: MNO: **Generate:** K_s
11: MNO: **Calculate:** $EN^{K_s}_{Pub_U}$, $EN^{K_s}_{Pub_{MNO}}$, and $EN^{M}_{K_s}$
12: $MNO \rightarrow SC_{Sub}$: **Delegate rights**
13: SC_{Sub}: $Status_{sub} = 3$ and $SC_{Sub} \rightarrow MNO$: Ack
14: $MNO \rightarrow IPFS$: **Store** $EN^{M}_{K_s}$
15: $IPFS \rightarrow MNO$: $CID_{EN^{M}_{K_s}}$
16: $MNO \rightarrow SC_U$: **Store** $EN^{K_s}_{Pub_U}$, $EN^{K_s}_{Pub_{MNO}}$

cryptography algorithms (e.g., PGP, Pretty good privacy, algorithm). To apply this method, MNO executes the following steps:

- Generates symmetric key K_s;
- Encrypts K_s using Pub_U and Pub_{MNO} and gets $EN^{K_s}_{Pub_U}$ and $EN^{K_s}_{Pub_{MNO}}$
- Encrypts M with K_s to get $EN^{M}_{K_s}$

6 MNO needs the write permission in `Update()` function of SC_U to modify $Attr_U$. So, it requests SC_{Sub} to execute the write delegation procedure by sending the $Addr_U$ to it. SC_{Sub} retrieves the user request and verifies the following condition: $Status_{sub} == 2$. If the validation is successful, MNO would got the update permission in SC_U. Note that, the address of the SC_{Sub} is immutably written in SC_{AB}. So, SC_U can be assured that SC_{Sub} is an eligible contract to

change the ownership.

7 MNO stores $EN^{M}_{K_s}$ in IFPS as a distributed database. After storing the data in IPFS, it would be indexed by a cryptographic hash function, which results in returning its unique content identifier (CID) to MNO. The CID (let's call it $CID_{EN^{M}_{K_s}}$) can be used for further access to the data in IPFS. Moreover, MNO can store $Attr_U$ containing $EN^{K_s}_{Pub_U}$ and $EN^{K_s}_{Pub_{MNO}}$ into SC_U. For further user connections, MNO can verify the user's address, and get its profile from IPFS.

Using this procedure, the user profile can be retrieved either by the user or the MNO. The other entities won't be able to have access to plain-text user data.

6.2.2 BC-AKA: Blockchain-based Authentication and Key-management

The user registration procedure introduces the user to the network to make it capable of finding the user. The initial registration would be executed when the user turns her phone on. After that, periodic registrations are required to keep the user known for the network. In this section, we describe Blockchain-based registration along with the session key-agreement procedure. Note that, to decrease the complexity and latency, both registration and key-agreement processes would be done in one phase. Following the procedure is explained (see Fig. 6.2 and `Procedure 2`).

1 Firstly, U sends a registration request to a connectivity provider (CP) which can be MNO or an external provider. After receiving the user's request, CP responds by asking to send the identification data and encrypting a challenge (i.e., *nonce*) with Pr_U.

2 U calculates $Hash(nonce + 1)$ using Keccak-256 and signs the result with her private key. Finally, she responds CP by encapsulating the encrypted hash along with $Addr_U$ (in plain text). The response is as follows:

$$Data^{U}_{reg} = [EN^{Pr_U}_{Hash(nonce+1)}, Addr_U]$$

After receiving the response, CP calls `Validate()` function of SC_{Reg} by transmitting $Data^{U}_{reg}$ and *nonce*, to verify the request.

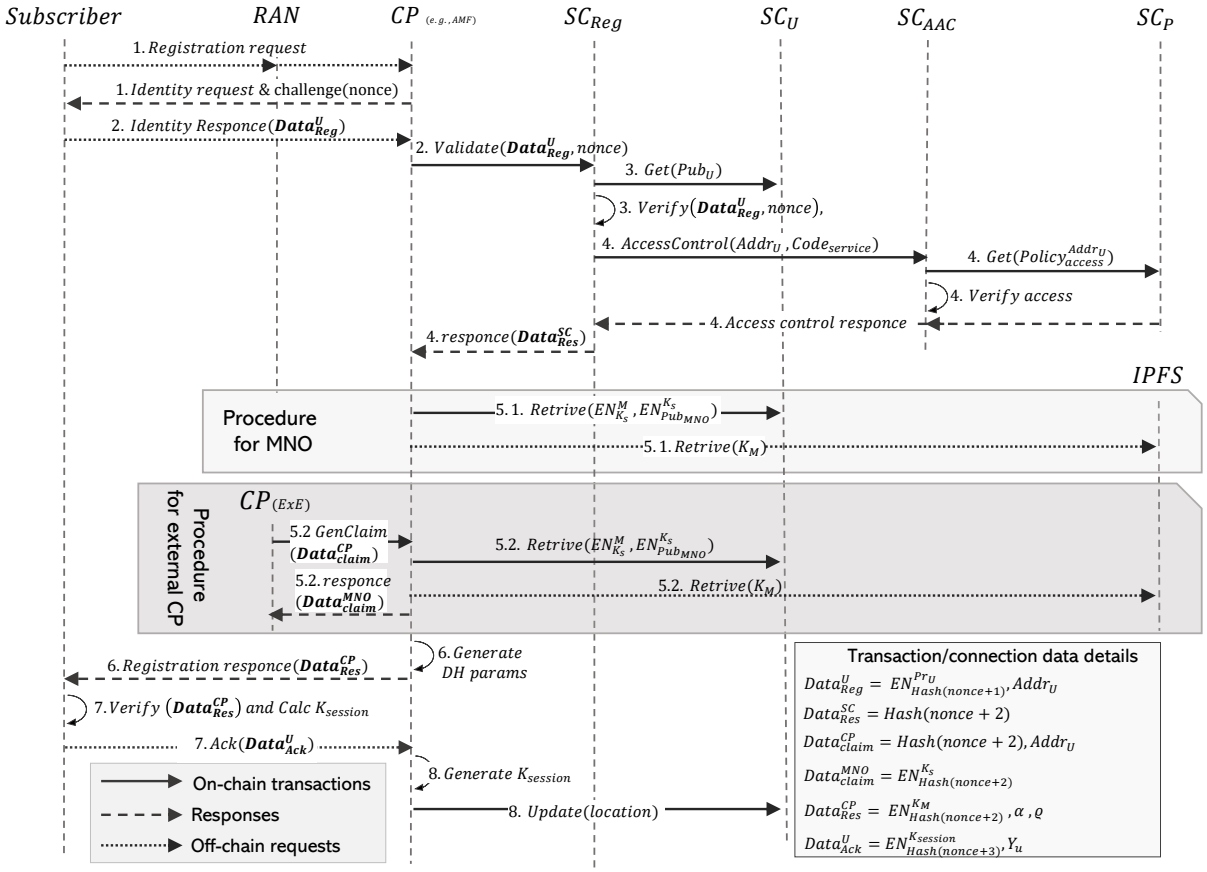


Figure 6.2: User registration (authentication) and Key-agreement procedure

Procedure 2: *User registration, key-agreement and authentication*

Parameters:

I: $Addr_u$: User's Blockchain address
II: Pub_u, Pr_u, K_{M_u} : User's public/ private and shared key
III: *nonce*: A random challenge
IV: p, g: A big prime number and its prime root in Diffie-Hellman algorithm
V: X_s, Y_s, K_s: Private, public, and session key of Diffie-Hellman algorithm

Functions:

I: $H(d)$: Calculating the hash of d by Keccak-256
II: $En(d)_k$: Encryption of d by key k
III: $Sg(d)$: Signing d (encryption of d with private key)
IV: $DHCalc(o, p)$: calculating the o parameter of Diffie-Hellman algorithm using p

Procedure steps:

1: $U \rightarrow CP$: **Request** for connection
2: $CP \rightarrow U$: **Request** for identity & **Response** to *nonce*
3: $U \rightarrow CP$: $[Sg(Addr_u) \parallel En(H(nonce+1))_{K_{M_u}} \parallel Addr_u]$
4: $CP \rightarrow SC_{aac}$: **Validate** the message of step (3)
5: $SC_{aac} \leftrightarrow SC_{ucl}$: **Get** SC_{user}
6: $SC_{aac} \rightarrow SC_{user}$: **Get** subscription data
7: SC_{aac}: **Verify** $Sg(Addr_u)$ and $En(H(nonce+1))_{K_{Mu}}$
8: **if** $verification$ is $successful$
$SC_{aac} \rightarrow CP$: $H(nonce+2)$
9: CP: **Select** p, g, X_{cp} and **Calculate** Y_{cp}
10: $CP \rightarrow U$: $[En(H(nonce+2))_{K_{M_u}} \parallel p \parallel g \parallel Y_{cp}]$
11: U: **Verify** step 10, **Select** X_u, **Calculates** Y_u, K_s
12: **if** $verification$ is $successful$
$U \rightarrow CP$: $[En(H(nonce+3))_{K_s}, Addr_u, Y_u]$
13: CP: **Calculate** K_s, and **Verify** message step 12
14: $CP \rightarrow SC_{aac}$: **Request** to **update** user data in SC_{user}

3 SC_{Reg} gets the address of SC_U from SC_{UL} using $Addr_U$ and retrieves Pub_U. Then the verification procedure is as follows:

$$Addr_U == DE^{Pub_U}_{EN^{PR_U}_{Hash(nonce+1)}}$$

4 If validation was successful, SC_{Reg} requests SC_{AAC} to verify the user's access permissions by sending $< Code_{Service_{CP}}, Addr_U >$. SC_{AAC} retrieves the user's specific access policies ($Policy_{access_U}$) and verifies the following conditions:

$$Time_{exp} > Time_{current}$$

$$IP_U \neq Filter_{IP}$$

$$Location_U \neq Filter_{location}$$

$$Balance_{UU}$$

Once SC_{Reg} get the access control result, it responds CP by sending $Hash(nonce+2)$. If the validation was not successful, SC_{Reg} would send the deny response.

5 The following perform mutual authentication and create a session key for further connections. In this regard, two main scenarios are explained as follows:

(a) *If CP is MNO:* Since MNO has access to the user profile, it retrieves $EN^{K_s}_{Pub_{MNO}}$ from SC_U to find the symmetric key to decrypt the user data. Then it retrieves $EN^{M}_{K_s}$ from IPFS and computes $M = DE^{K_s}_{EN^{M}_{K_s}}$. One of the data stored in M is K_M which is the master symmetric key in the user's SIM-Card for the registration procedure.

(b) *If CP is external provider:* Since CP doesn't have access to the user profile in IPFS, to preserve the user privacy and avoid any data leakage, CP needs to ask MNO to send a claim to show its authenticity to the user. To do so, CP sends $Hash(nonce+2)$ to MNO along with $Addr_U$ to encrypt it by K_M. MNO retrieves K_M with the aforementioned steps, and sends $EN^{K_M}_{Hash(nonce+2)}$ to CP.

6 CP (external provider or MNO), selects three parameters of Diffie-hellman [223] key agreement algorithm namely, α, ϱ, and a private key X_{CP}. These parameters are defined in Table 5.1. Using these parameters, CPcalculates its session public key,Y_{CP} and transmits the following response to user:

$$< \alpha, \varrho, Y_{CP}, EN^{K_M}_{Hash(nonce+2)} >$$

7 Once receiving the reply, U verifies:

$$Hash(nonce+2) == DE^{K_M}_{EN^{K_M}_{Hash(nonce+2)}}$$

Since K_M is known, only, for the U and MNO, if $Hash(nonce+2)$ was valid, we can claim that the sender is trusted. Then, the U chooses a Diffie-hellman private key,X_u, and using α and ϱ calculates its public key (Y_u). Next, U

calculates the session key, $K_{session}$, with the use of Y_{CP} and X_u. Finally, the user sends an acknowledgment to CP concerning successful verification and accepting the connection. To do so, it encrypts $Hash(nonce+3)$ with $K_{session}$, and encapsulates the result with Y_u in plain text.

8 Once receiving the acknowledgment from U in CP, it recalculates session key (lets call in $K'_{session}$) using Y_u, and X_{CP}. Then, using the generated key, CP can validate:

$$Hash(nonce+3) == DE^{K'_s}_{EN^{Ks}_{Hash(nonce+3)}}$$

if the condition passed, CP can update user's location (i.e., at least $< ID_{SN}, UTEID, DTEID >$) in SC_U.

Note that, up until last step, nothing is written in the Blockchain. So, it is not affected by the Blockchain's consensus latency.

6.2.3 BC-MM: Blockchain-based Mobility Management

To provide seamless connectivity for the mobile user, Blockchain's intrinsic authentication capability (using public-private key pairs), immutability, and non-repudiation can be beneficial. In this regard, we propose to eliminate the extra authentication and access control procedure while the user's handover procedure. To do so, the source network entity would propagate the user's identity in the network using a dedicated smart contract. So, the target entity can validate the identity (because the data is propagated in a Blockchain transaction and signed by a trusted party), and there is no need to redo the AAC. The proposed handover procedure is depicted in Fig. 8.4 as detailed below.

1 Based on the users periodic measurement report, the serving connectivity provider (CP_s) starts a handover procedure and firstly, selects the target connectivity provider (CP_t).

2 To propagate the user's identity in the network, CP_s calls the `insert()` function of SC_{HO} to insert a summary of the handover request, that, at least, contains $Data^{SC}_{HO}$ as follows:

$$[Addr_{CP_s}, Addr_{CP_t}, Addr_U, Status_{HO}, \alpha, \varrho]$$

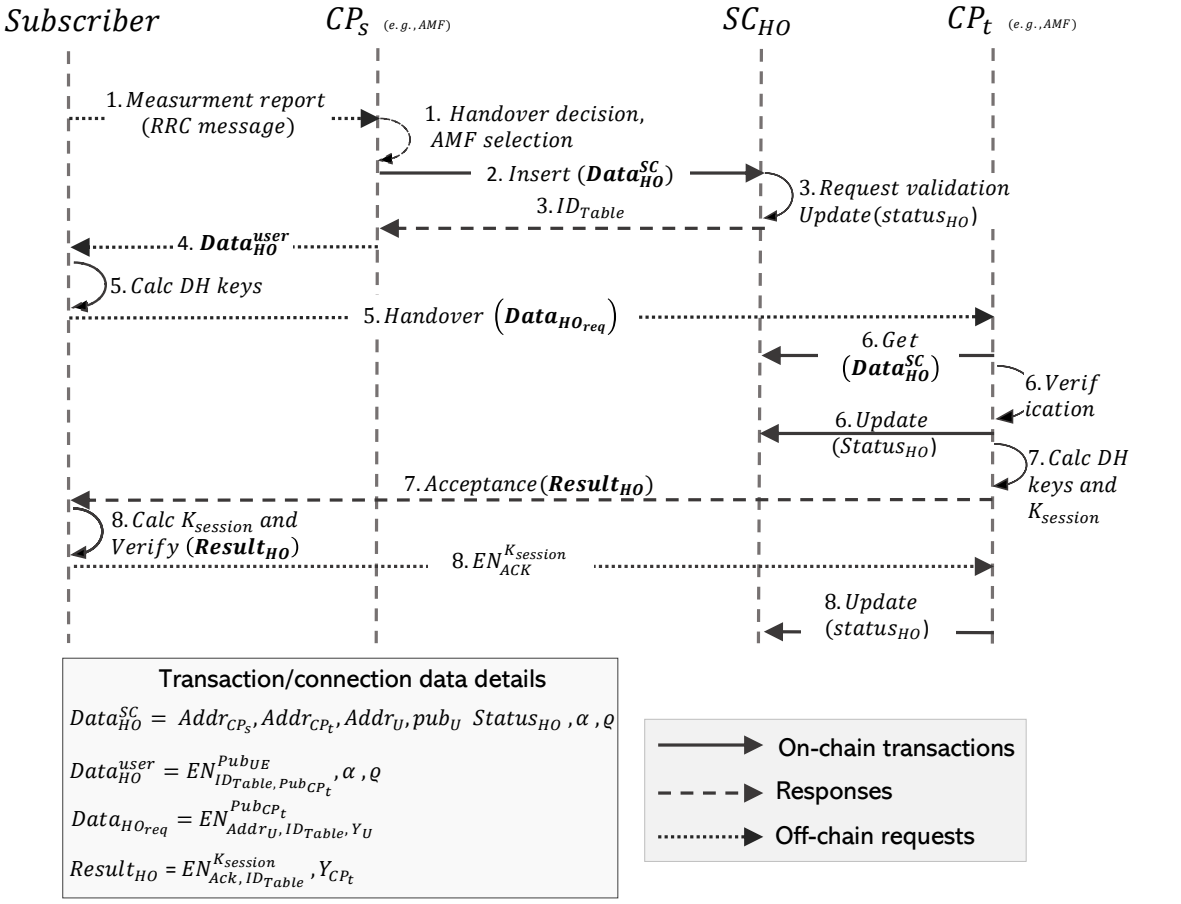


Figure 6.3: Proposed handover procedure.

where α and ϱ are Diffie-Hellman key-agreement parameters. Note that, defining these two parameters by CP_s not only will not cause security issues in the next levels (because they are public parameters), but also it would decrease the number of required steps for the key-agreement procedure. Moreover, $Status_{HO}$ defines the user's handover status that can be 1 to 3 indicating `'Requested'`, `'Validated'`, and `'Terminated'`, respectively.

3 SC_{HO} records the handover request with $Status_{HO} = 1$, and returns ID_{Table} to CP_s as a unique identifier of the successful transaction.

4 CP informs the user of starting the handover procedure by sending the $Data_{HO}^{user}$ as follows:

$$[EN_{ID_{Table}, Pub_{CP_t}}^{Pub_U}, \alpha, \varrho]$$

5 After receiving the handover signal and $Data_{HO}^{user}$, U recognizes that the handover process is started. So, it selects a private key (X_U) and calculates the public key (Y_U) based on the received α and ϱ. U decrypts $Data_{HO}^{user}$ to find Pub_{CP_t}. Then, she encapsulates the handover request ($Data_{HO_{req}}$) in following format and send it to CP_t:

$$[EN^{Pub_{CP_t}}_{ID_{Table},Addr_U}, Y_U]$$

6 Once receiving the handover request, CP_t needs to assure that the user's identity is already propagated in Blockchain. To do so, after decrypting $Data_{HO_{req}}$, it fetches $Data_{HO}^{SC}$ from SC_{HO} using ID_{Table}. Moreover, it decides to check or skip the user's access policy. If yes, the process would be the same as $Step-4$ in 6.2.2, unless CP_t verifies $Addr_U$ to be the same as the stored address in SC_{HO}. If validation is successful, CP_t updates $Status_{HO} = 2$ (indeed SC_{HO} would validate the sender to be sure that $msg.sender == Addr_{CP_t}$).

7 The next step is mutual authentication. Since ID_{Table} was encrypted by Pub_{CP_t}, it can be used as authentication parameter. So, after selecting X_{CP}, and calculating Y_{CP} by CP_t, it calculates the symmetric session key $K_{session}$ using Y_U (that is sent in $Data_{HO_{req}}$). Then, CP_t encapsulates $Result_{HO}$ as follows and send it back to the user:

$$[EN^{K_s}_{ID_{Table},Ack}, Y_{CP}]$$

8 The user receives Y_{CP} in plain text, recalculates $K_{session}$, and decrypts $Result_{HO}$ to verify: firstly, the agreed key is correct, and secondly, CP_t is an authenticated provider. Then, U encrypts its acknowledge message with the agreed session key and sends it back to CP_t. After receiving the packet, CP_t updates SC_U (not shown in the Figure, similar to $Step-8$ in Section 6.2.2) and the $Status_{HO}$ in SC_{HO} to `'Finished'`.

6.2.4 Session management procedure

After registration or handover, a session establishment procedure will be initialized. This procedure involves updating the user's PDU session, establishing a new session/ resource, and releasing the previous ones. The session handling procedure is as follows:

Procedure 3: *Handover and user mobility*

Parameters:

I: K_T: a temporal key for handover procedure

Assumptions:

I: U is authenticated in $Procedure1$

Procedure steps:

1: CP_s: **Decide** to start handover
2: $CP_s \rightarrow SC_{handover}$: **Insert** $Addr_u$, $Addr_{CP_s}$, $Addr_{CP_t}$, Pub_u, p, g
3: $SC_{handover} \rightarrow CP_s$: $Table\_ID$
4: $CP_s \rightarrow U$: $Table\_ID$, K_T, p, g, $nonce$, Pub_{CP_t}, $Addr_{CP_t}$
5: $CP_s \rightarrow CP_t$: $Table\_ID$, K_T, $nonce$
6: U: **select** X_u, **calculate** Y_u
7: $U \rightarrow CP_t$: **Handover** $[Table\_ID \parallel Sg(Addr_u) \parallel En(H(nonce+1), Y_u)_{K_T}]$
8: CP_t: **Validate** the request using stored data in $SC_{handover}$
9: CP_t: **Select** X_{CP}, **Calculates** Y_{CP}, K_s
10: $CP_t \rightarrow U$: **Accept** $Sg(Addr_{CP_t}) \parallel En(H(nonce+2), Y_{CP})_{K_T}$
11: U: **Verify** step 10
12: **if** $verification$ is $successful$
$U \rightarrow CP_t$: connected
13: U: **Calculate** K_s
14: $CP_t \rightarrow SC_{aac}$: **Request** to **update** user data in SC_{user}

Firstly, the connectivity provider creates a session update signal. So, a new uplink PDU session ID (i.e., $UTEID$) would be assigned to $Addr_U$, and if it is necessary, the user's IP can be renewed.

In the case of *handover*, to avoid data loss during the creation of the GPRS tunnel, it is needed to buffer the user's downlink data. So, the CP finds the user's previous serving network ID (CP_s) from SC_U and sends a buffer request. After receiving this signal, CP_s starts to buffer the user's downlink data, remove the user from the list of its connected users, and release the resources. Moreover, the CP updates the user's current serving network. The history of the connection (i.e., $Addr_U$, $UTEID$, and SC_U) would be registered in CP's local memory (or in a specific separate smart contract).

After finishing the registration or handover procedure, the user starts to send its uplink data to CP. By receiving the packets from the user, CP can retrieve $Addr_U$ and SC_U to find the related connection path. Then, CP would be able to allocate downlink TEID ($DTEID$) and ask the CP_s to send the user's buffered downlink data. Moreover, CP updates $DTEID$ and other related data in the SC_U.

Procedure 4: *Payment procedure*
Assumptions:
I: All connection requests have $Code_{service}$
II: U is already authenticated
Procedure steps:
1: $U \rightarrow CP/SP$: **Access request**
2: $CP \rightarrow SC_{AAC}$: **Validate** $Addr_U$, $Code_{service}$
3: $SC_{AAC} \leftrightarrow SC_{ExE}$: **Get** $SC_{P_{CP/SP}}$
4: $SC_{AAC} \leftrightarrow SC_{P_{CP/SP}}$: **Get** policies of $Code_{service}$, $Addr_U$
5: SC_{AAC}: **Then Transfer** $MinBalance_{ExE}$ to SC_{AAC}
6: $SC_{AAC} \rightarrow CP/SP \rightarrow U$ **Access permission**
7: SC_{AAC}: **Record** $Time_{block}$, $Addr_U$, $Addr_{SP/CP}$, $Price$, $Code_{service}$
After termination of the connection
8: $CP/SP \rightarrow SC_{AAC}$: **Terminate** $Addr_U$, $Code_{service}$
9: SC_{AAC}: **Validate** request and **Calculate** $Price_{final}$
10: $SC_{AAC} \rightarrow Addr_{MNO}$: **Transfer** $Share_{MNO} \times Price_{final}$
11: $SC_{AAC} \rightarrow Addr_{SP/CP}$: **Transfer** rest of $Price_{final} - Price_{Step-10}$
12: $SC_{AAC} \rightarrow SC_U$: **Update** $Balance_U$

6.2.5 BC-Pay: Blockchain-based Billing

As mentioned before, in the proposed architecture, the MNO's billing procedure is migrated into the Blockchain, relying on a new business model, to decrease the complexity while increasing the security. So, the first prerequisite is to assure MNO that if it delivers the proper service, the external provider will pay the cost on behalf of the user. To reach to this goal, we defined SC_P for each connectivity/service provider in which $Policy_{price}$ defines the agreed price between $MNO - CP$, and $CP - U$ (i.e., $Price$, $share_{MNO}$, $MinBalance_U$, $MinBalance_{ExE}$).

Note that all users are either served by the MNO or external providers. When the MNO itself is providing the service, $share_{MNO}$ is 100%. When the external service/connectivity provider is serving the user (e.g., by externally provided RAN or the private vendor that provides a connection for its IoT device), the price of connection would be calculated based on the $Price$ parameter in SC_P, and SC_{AAC} -that is also responsible for payment procedure- would pay the MNO according to $share_{MNO}$. The payment procedures are provided in *Procedure 4*.

Procedure 4 [Steps 1-4]: Firstly, the authenticated user (see Section 6.2.2) sends the access request (i.e., $Code_{service}$) to CP. After receiving the request, since the user's access control procedure is validated, CP requests SC_{AAC} to manage the pay-

ment. To do so, SC_{AAC} fetches the connection price, the user's balance, and the external provider's balance. Next, regarding the pre-defined $Policy_{price}$, SC_{AAC} makes sure that:

$$Balance_{CP} \geq MinBalance_{CP}$$

$$Balance_{UU}$$

$$Balance_{U} \geq Price$$

If the decision determines that the user and CP are eligible, U would be able to access the service without paying MNO.

Procedure 4 [Steps 5-7]: To perform the **billing procedure**, SC_{AAC} deducts $MinBalance_{ExE}$ from the Blockchain account of the service/ connectivity provider, and deposit it in its account (i.e., in SC_{AAC} as the distributed trusted party for MNO, external provider, and the user). Note that, the deducted minimum price is for assuring the connection; the actual connection price would be more or less than this amount. After transferring the balance, SC_{AAC} keeps the log of connection with $Addr_U$, $Code_{service}$, the current block's timestamp, and the price that is deducted from the provider's account.

Procedure 4 [Steps 8-12]: When the user's connection to that specific provider is terminated, a trigger would be sent to the SC_{AAC}. Then, the contract fetches the connection information from the Blockchain and pays the MNO or another connection provider (s) according to the connection time, $Price$, and the $Share_{MNO}$. Finally, the real connection cost (or predefined $Price$) would be deducted from the user's account. The MNO or external connectivity provider can withdraw their money from their account.

In the pre-mentioned procedure, the SC_{AAC} manages the user's access mostly by retrieving the user's data from the Blockchain, without updating the state. So, the consensus operation of the Blockchain would not negatively affect the user's access performance. Two steps involve changing the Blockchain's state: 1) money transfer and 2) recording a summary of the user's connection. Both these situations would be done after (or in parallel with) the access decision-making process.

Note that, indeed changing the business model of the cellular network market would be an obstacle to the acceptance of the proposed method by the MNOs. Currently, the majority of MNOs charge the users in a pre-paid solution. However, due to the nature of the proposed method (i.e., the existence of many SPs and CPs with a variety of services at different costs), providing a 'Pay as you go' solution is the

better option to guarantee the benefits of all participating entities and users. So, we proposed a *hybrid method of "pay as you go" and "pre-paid"*. In this solution, when a user requests for the services or connectivities provided by external entities, $Price$ would be decreased from the user's wallet (i.e., similar to prepaid solution). For example, assume that the $Price$ is defined in the granularity of $1hour$ connection or $10GB$ data. If the user uses less than these granularities, she is paying similarly to the pre-paid method; but if she uses more than that, the remaining cost would be deducted from her account based on the usage (i.e., 'pay as you go'). This procedure is also applicable for the external providers and defined $MinBalance_U$.

6.3 Blockchain-based Application-level services

6.3.1 Blockchain-based access control for service provisioning in cellular networks

In this section, we introduce a Blockchain-based ABAC system for service provisioning in cellular networks which provides the opportunity of introducing new business and pricing models in this market. The overall steps of access control and payment procedures of the proposed method are enumerated as follows:

i. The user registers in the provided DApp by sending a request to a dedicated smart contract through a transaction in the Blockchain.

ii. A unique smart contract is deployed for the user (only for the first time).

iii. The User chooses her desired service from the list of available services for registration.

iv. To access the service, the user sends an access transaction to the dedicated smart contract.

v. The access manager smart contract authorizes the user regarding the stored policies of the requested service.

vi. According to the pricing model of the service, the access manager smart contract blocks an amount of money.

vii. After termination of the service usage, the access manager smart contract pays the MNO, SP, and (if it is required) the user, according to the pricing model.

Fig. 6.4 depicts the connection between contracts, contract attributes, and their definition. Note that, due to the specific requirements of the use-case some smart contracts are alternated with other contracts (e.g., instead of SC_{ExE} we used SC_{SPL} which is more clear in this use case).

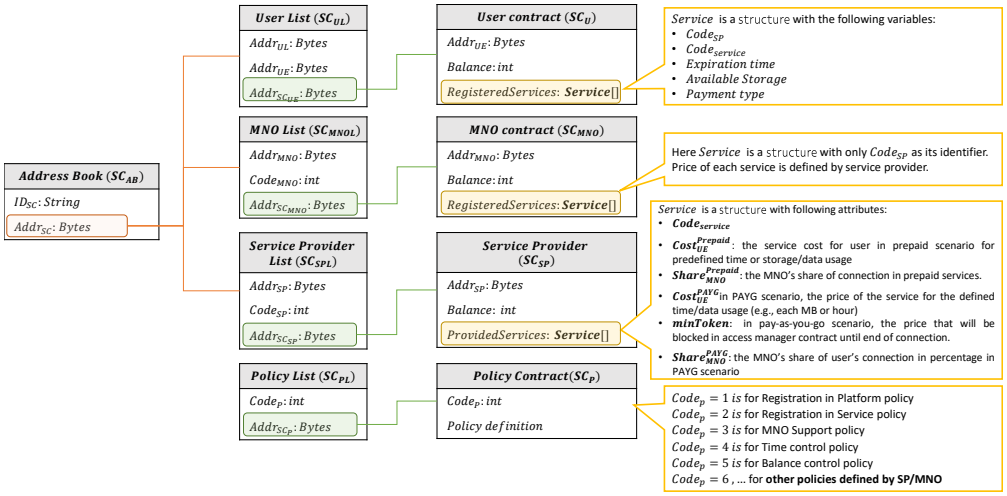


Figure 6.4: Relations and connections among designed contracts

6.3.1.1 SP subscription in the system

In this step, the subscription procedure of SPs is described. Note that, the user subscription is based on the procedure of Section 6.2.1

The registration of the SP in the system is via the following steps (Fig. 6.5 for a more detailed description of this procedure):

i. First, SP sends the registration request to SC_{Sub}. Since each SP can register only one time to the system, SC_{Sub} needs to verify that the SP has not registered before. To do so, it calls the `isExist()` function of SC_{SPL} and sends the address of the caller as its argument. Note that, here the caller is SP, so SC_{SPL} send the $Addr_{sp}$ in `isExist()` function. Since `msg.sender` in Solidity language is the address of the caller or the creator of the transaction, in the rest of the paper we use `msg.sender` to indicate the caller of the function.

ii. After receiving confirmation from SC_{SPL}, SC_{Sub} deploys the SP's unique smart contract (i.e., SC_{SP}). Note that, deployment of smart contracts for all entities in the system is only can be done by SC_{Sub}. Therefore, the `constructor()` of

SC_{SP} verifies that `msg.sender` == $Addr_{SC_{Sub}}$. After deployment of SC_{SP}, its address is sent to SP.

iii. Finally, SP as the `owner` of the smart contract can add its preferred services into SC_{SP}. These services would be advertised to the network providers and the users for further registration. While inserting the services in SC_{SP}, the SP defines the costs for prepaid and pay-as-you-go (PAYG) scenarios (note that the SPs can choose one of these payment solutions based on their preference). The following costs will be added to SC_{SP} for each advertised service (Fig. 5.1 depicts all relations between smart contracts in this use-case):

- $Cost_U^{Prepaid}$: defines the prepaid cost that the user needs to pay for registering in this service for a predefined time/data usage.
- $Share_{MNO}^{Prepaid}$: defines the fee that SP will pay to MNO on behalf of the user, after user access termination.
- $Cost_U^{PAYG}$: defines the fee that the user needs to pay per hour/MB while using a PAYG service.
- $minToken$: defines the minimum required token in the user's wallet to give access to a PAYG service. Note that, the user's real usage may be more than this amount, so, the user will be charged after access termination for the remaining part. or, if the real cost is less than this amount, the user's wallet will be refunded.
- $Share_{MNO}^{PAYG}$: defines the MNO's share in percentage from the user's real usage of service. So, the cost of the user's real usage will be separated between SP and MNO based on this value.

6.3.1.2 Attribute-Based Access Control

After a successful subscription to the system and services, the users are able to access provided services through the proposed system. The access control procedure for different pricing methods is enumerated as follows (see Fig. 6.6):

- *User access verification:*

  i. U selects a service among registered services, (this selection creates an access request transaction to SC_{AAC} smart contract).

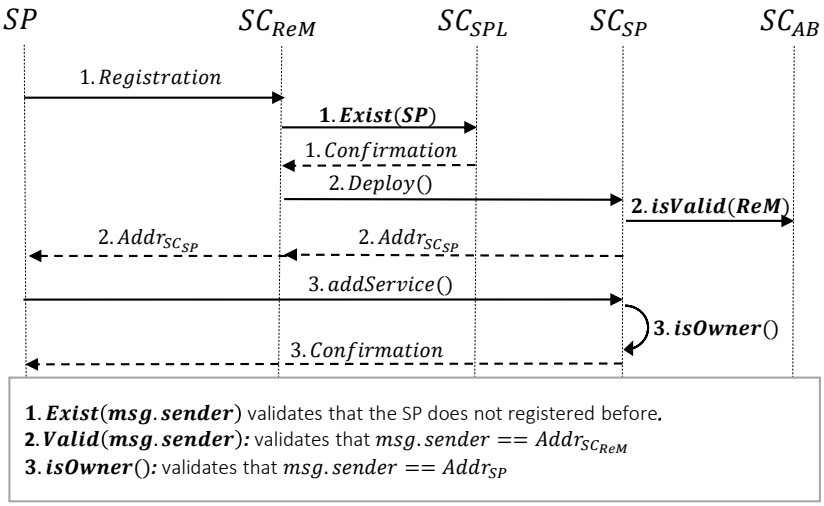


Figure 6.5: SP subscription steps

ii. After receiving the request, SC_{AAC} fetches all policies that are defined as prerequisites for access o the service (e.g., checking the user's balance, checking the geographical IP, etc.).

iii. After getting the list of policies, SC_{AAC} retrieves the address of each smart contract in the list. Then it can verify the user's eligibility based on each policy (i.e., for the verification we defined a `isEligible()` function, that compares the user's access attributes with the defined rules).

– *User access control to the prepaid pricing model:*

iv. If the access verification is successful, SC_{AAC} validates the SP's balance for further user access. It is important to mention that, in the prepaid pricing model, the user is paid to the SP while the registration step, and while using the service, the user would not pay to MNO (e.g., the user's mobile data will not be reduced while using the service); and, the SP is the entity that will pay to MNO on behalf of the user. So, SC_{AAC} verifies that $Balance_{SP} \geq Share_{MNO}^{Prepaid}$.

v. If the balance verification is successful, the $Share_{MNO}^{Prepaid}$ will be transferred from the SP's wallet to SC_{AAC} as a distributed trusted party for all entities. Note that, this transfer is based on ERC20 standard [**?**]. Record of this payment is added to SC_{AAC} as a mapping of the user's address to a

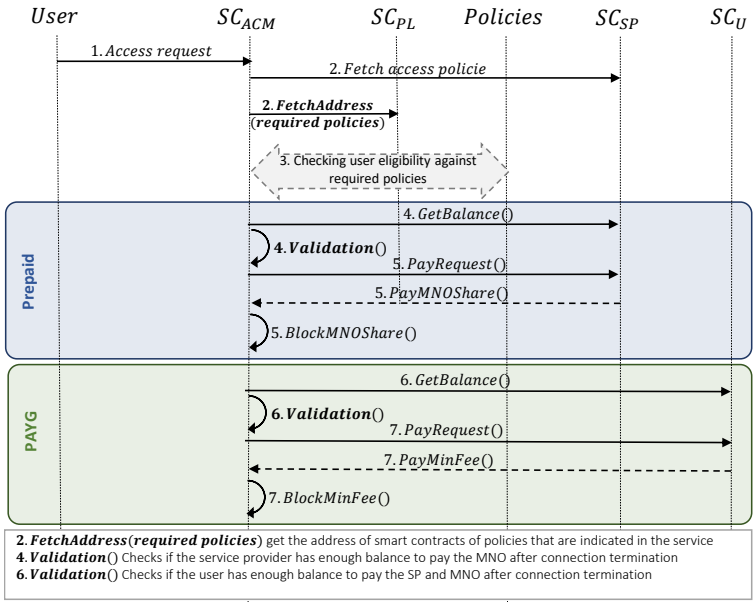


Figure 6.6: The ABAC procedure for user access to the services in the prepaid and PAYG scenarios

balance as follows:

$$Balance_U \xleftarrow{Addr_U} StoredBalance_U$$

– *User access control to the PAYG pricing model:*

vi. If the access verification is successful, SC_{AAC} validates the user's balance since in this pricing model the user needs to directly pay to the SP and MNO separately, according to the real service utilization. So, SC_{AAC} verifies that $Balance_U \geq minToken$.

vii. If the balance verification is successful, the $minToken$ will be transferred from the user's wallet to SC_{AAC}. Note that, this amount balance is only a minimum balance to guarantee the payment to the SP and MNO. It means, the user's real utilization will be sent to SC_{AAC} after termination, and the real price will be calculated at that time. same as the prepaid model, the record of this payment is added to SC_{AAC}.

6.3.1.3 Payment

Once the user terminated the service utilization, the pricing and payment procedure will be executed as follows:

- *Checking service type:*
  - i. U sends the termination transaction to SC_{AAC} smart contract. This contract checks the service type for handling the further payment procedure.
- *Payment in the prepaid pricing model:*
  - iii. In the prepaid pricing model, once SC_{AAC} receives the termination transaction, it retrieves the blocked $Share_{MNO}^{Prepaid}$ and pay it to MNO. This transfer complies with the ERC20 standard [**?**].
- *Payment in the PAYG pricing model:*
  - iv. First, SC_{ACM} calculates the real service price as follows:

$$FinalPrice = Usage \times Cost_{U}^{PAYG}$$

Then, it calculates the amount of money that the user needs to pay or reimbursed as follows:

$$UserPayment = FinalPrice - minToken$$

In this equation, if $UserPayment \geq 0$, the user needs to pay this amount, otherwise, the user will be refunded by $userPayment$.

  - v. If $UserPayment \geq 0$, payment request will be sent to user, and SC_{ACM} will receive the tokens from user's wallet.
  - vi. SC_{AAC} calculates the MNO, and SP's shares from $UserPayment$ as follows, and transfer token to each one.

$$MNOshare = (UserPayment + minToken) \times Share_{MNO}^{PAYG}$$

$$SPshare = (UserPayment + minToken) - MNOshare$$

6.3.2 BC-MNP: Mobile Number Portability for 5G and B5G

On top of the aforementioned network functionalities and the proposed architecture, some services are easier and more straightforward to implement. As an imposed rule

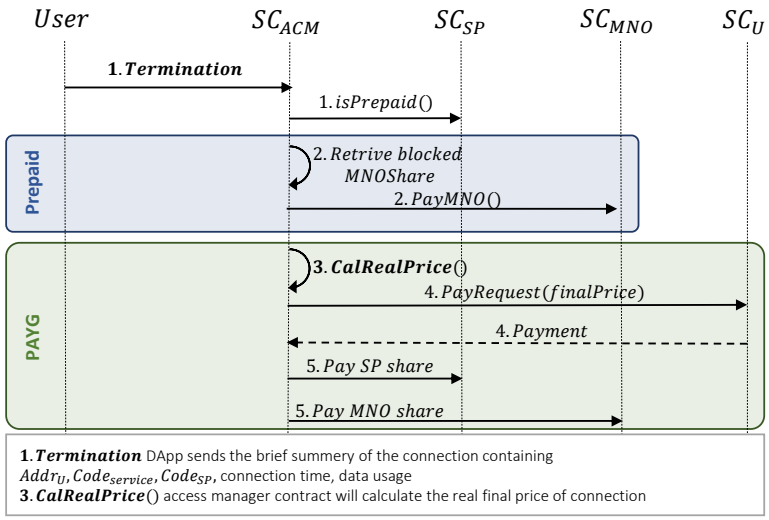


Figure 6.7: The payment procedure to SP and MNO in the prepaid and PAYG scenarios

by national regulatory authorities, *mobile number porting* is provided in more than 100 countries all over the world. More than 40% [224] of the mobile numbers have been ported in those countries, which implies the importance of this mechanism to enhance fair competition among MNOs and improve the subscriber's service quality and satisfaction. Location, Service, and Operator portability are the three types of mobile number portability, that deal with keeping the user's number while changing the location, changing the telecommunication service (e.g., between fixed telephone and mobile phone service), and switching between different MNOs, respectively [225]. In this work, our focus is on operator portability.

Mobile number porting is implemented in two general ways across the globe: Recipient-led and Donor-led [226]. In the first solution- which is mostly implemented in Europe and the U.S. - the new provider (Recipient Network Operator -RNO) is the one who arranges the required process with the old provider (Donor Network Operator -DNO), while in the latter solution, the subscribers need to contact the DNO to obtain a Porting Authorization Code (PAC), which they need to give it to the RNO for the further porting process (e.g., applied solution in U.K. [227]). The Recipient-led solution is the most dominant porting method. Fig. 6.8 depicts the mobile number porting process for both donor-led and recipient-led solutions [228].

Regardless of the MNP approach, the porting procedure has four main steps:

i. *Request:* The subscriber would request RNO to start the porting procedure (or

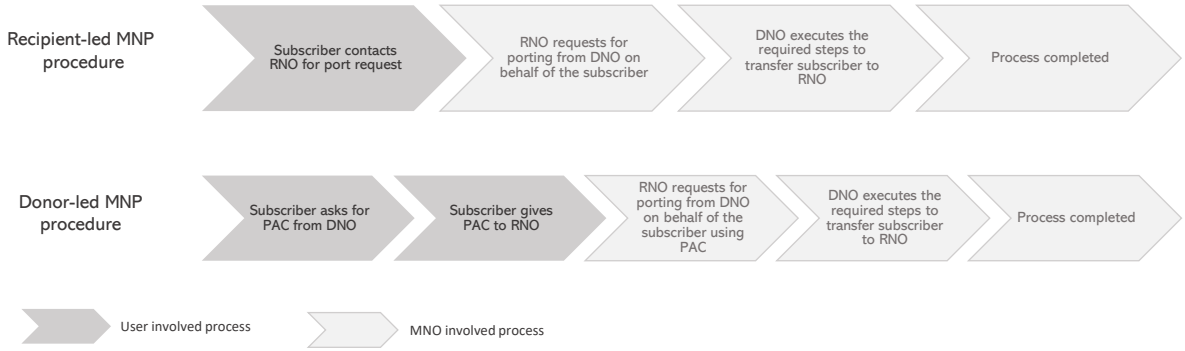


Figure 6.8: Recipient-led vs. Donor-led MNP procedures

request DNO for PAC).

ii. *Validation:* The RNO validates the subscriber's request by sending the validation request to DNO. Note that this request can be sent directly, or through another trusted third party.

iii. *Clearance:* DNO manages the legal clearance from legal authority to assure that the number doesn't have any legal issues.

iv. *Activation:* Once, the RNO (or trusted party) receives the clearance notification, asks DNO to remove the subscribers from its users and add them as the user of RNO.

several drawbacks of existing MNP procedure and user profile management are as follows:

- In current MNOs, a *centralized unit* handles subscribers' profile management and stores their subscription data. As mentioned by Tahir et al. [8], centralized storage can be a *single point for a data breach or data leakage* and bring a significant challenge regarding *security and privacy.*
- Centralized network functions are the main reason for authentication traffic spikes and malicious flooding of core network components. These attacks target user data management and porting procedure [229].
- Due to the manual MNP process, in real-world applications, the porting procedure is *highly time-consuming* from several days to weeks based on the country's regulation and the clearance delay.

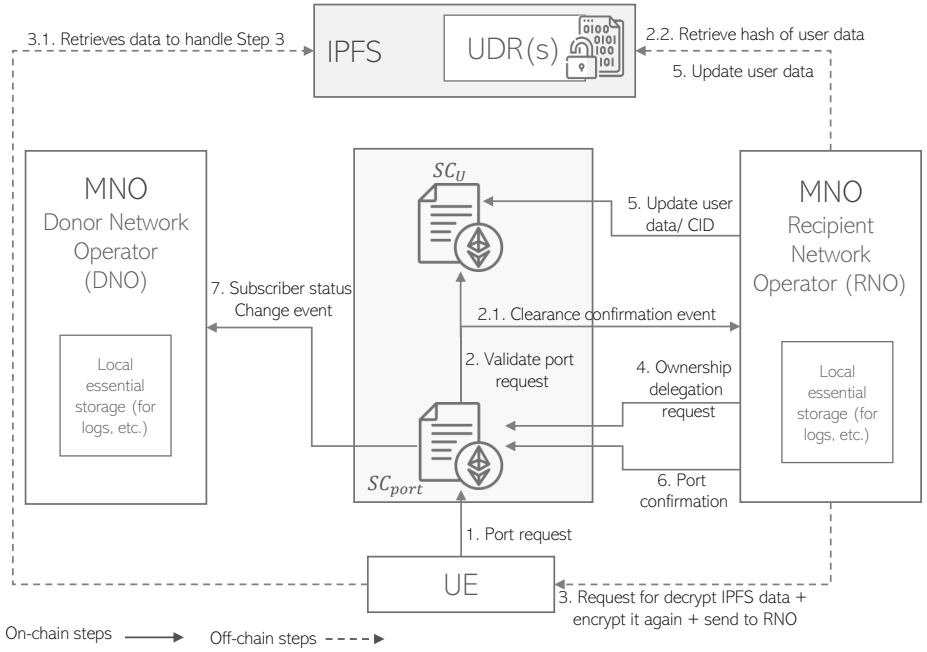


Figure 6.9: High-level overview of the proposed method for MNP and profile porting in distributed core network architecture for B5G

- In the current MNP procedure (both Donor-led and Recipient-led), the subscriber needs to *redo the subscription procedure* in RNO. It means the subscriber's profile doesn't port to the new MNO. From the user's perspective, this procedure is a repetitive function that can be avoided.

- In current systems Mobile number portability Clearance House *(MCH) -as a centralized operator-* manages the whole MNP procedure. Centralized MCH not only can be a single point of failure for *availability* and a *bottleneck for performance* of the porting procedure but also MNOs need to trust MCH which can pose a threat to user data protection.

- The MNP procedure is a complex procedure on the MCH side that has *high processing load* on this centralized point.

Addressing these constraints, we propose the following Blockchain-based distributed user profile management for beyond 5G (B5G) in which the user subscription, profile management, profile porting, and MNP procedure can be done in a distributed manner by altering the existing core network functions (see Fig. 6.9 as an overview).

This procedure relies on the subscription process introduced in 6.2.1.2. It means we

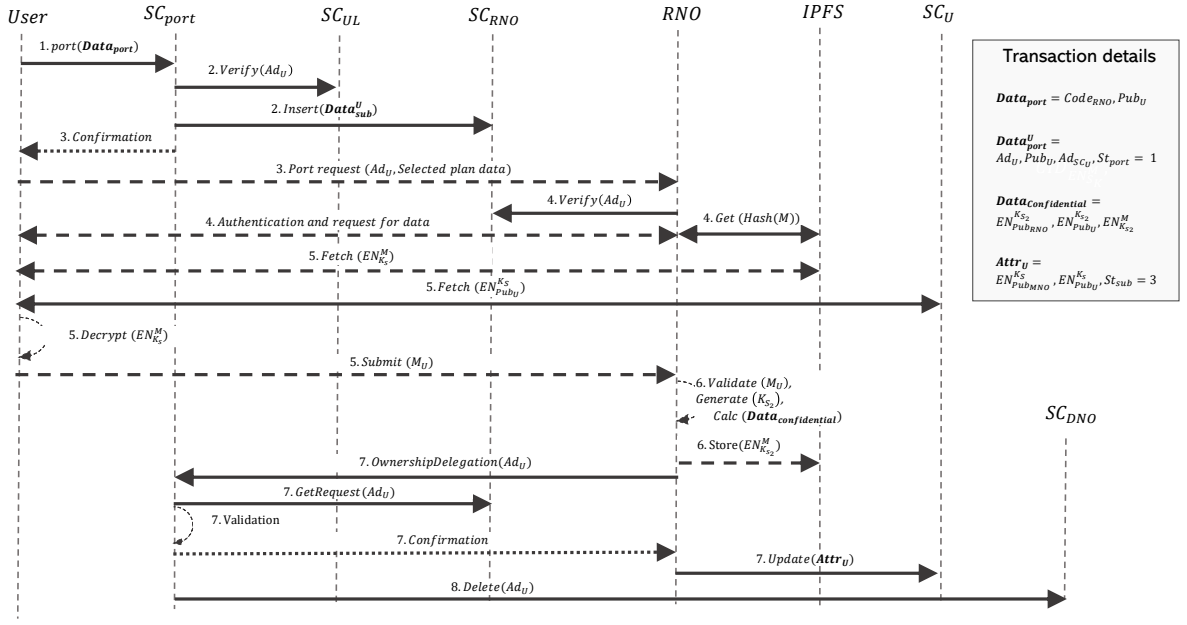


Figure 6.10: Mobile number and Profile Porting procedure

assume that U has already been subscribed to DNO, and aims to change her MNO to RNO while keeping $Number_u$. The detailed procedure is as follows (Fig. 6.10):

i. u sends the porting request to SC_{port} by creating a transaction in the Blockchain and sending:

$$< Code_{RNO}, Pub_U >$$

ii. SC_{port} verifies the user's record in the list of subscribed users, and if the user was subscribed, SC_{port} inserts the summary of the user's request in SC_{RNO}), using the following data:

$$< Ad_u, Pub_U, Add_{SC_U}, St_{port} = 1 >$$

iii. The request result would be sent to u, which redirects her to the port request page of RNO. The user sends $< Ad_u >$ to RNO and can select her proffered plan in the new operator.

iv. To validate the user request, RNO asks SC_{RNO} to confirm the user's request (i.e., Ad_U is already stored there), and verify $St_{req} == 1$. If these conditions are passed, RNO will authenticate the user using her Pub_U that is stored in SC_{RNO}. This step can be done by sending a challenge (encrypted by Pub_U) to

u and asking her to decrypt and resend it. This authentication would assure RNO that the eligible user is requesting porting. If all conditions passed, SC_{RNO} changes the St_{req} to 2.

v. Since the user data is stored in IPFS, and only DNO and the user can have access to that, RNO asks the user to send the decrypted data. User retrieves $EN_{K_s}^{M_{IPFS}}$ from IPFS, using $CID_{EN_{K_s}^{M}}$. She executes:

   - Retrieves $EN_{Pub_u}^{K_s}$ from SC_U;
   - Decrypts it with Pr_u and retrieves K_s;
   - Decrypts $EN_{K_s}^{M_{IPFS}}$ using K_s and retrieves M_{IPFS}

   Then, the user sends the data, let's call it M_U, to RNO. Note that, since the user is not eligible to modify her identity, RNO needs to verify the authenticity and originality of the received data.

vi. RNO receives M_U and validates its integrity with the previous version which is validated by DNO. To do so, SC_{RNO} retrieves $Hash(M_{IPFS})$ from SC_U. Then validates that $Hash(M_{IPFS}) == Hash(M_U)$. After successful validation, RNO generates new symmetric key, K_{s_2}, and calculates $EN_{K_{s_2}}^{M}$, $EN_{Pub_u}^{K_{s_2}}$ and $EN_{Pub_{MNO}}^{K_{s_2}}$. RNO stores $EN_{K_{s_2}}^{M}$ in IPFS and gets $CID_{EN_{K_{s_2}}^{M}}$).

vii. RNO requests SC_{port} to delegate the ownership of update function of SC_U to RNO. SC_{port} gets the record of user request and verifies that $St_{req} == 2$. If the validation is successful, the ownership will be delegated to RNO. Then, RNO stores $Attr_u$ into SC_U.

viii. SC_{port} sends a transaction to SC_{DNO} to remove the $< Ad_U, Number_u >$ from the list of its active users.

Termination phase: If the user aims to terminate her subscription with all MNOs it is required to remove the user's contract and the MNO's privilege to update her data. The following steps can be followed:

i. u sends the termination request to SC_{port};

ii. SC_{port} ensures that the message sender is the user.

iii. SC_{port} removes the user from SC_{MNO} and SC_{UL}.

iv. SC_{port} destroys SC_U. After execution of this step, no one can have write/read access to user data unless the previously downloaded versions.

Chapter 7

Testbed Deployment

Contents

7.1 Introduction

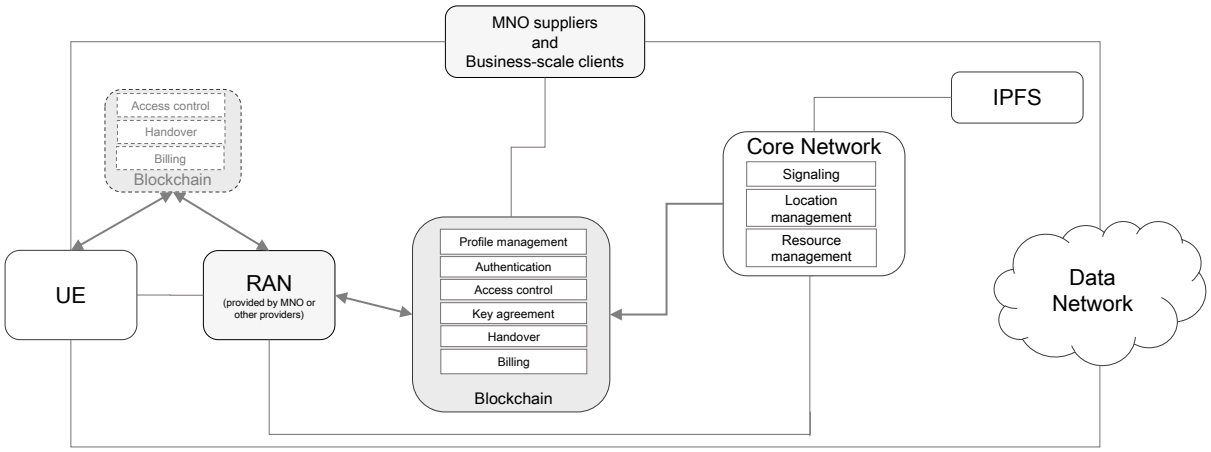
Evaluation of the proposed systems in different contributions of the work needs the deployment of the proper testbed and clarification of the deployed scenario. In this regard, the following subjects are provided in this chapter:

- Description of possible implementations of the proposed ecosystem at different levels. While several use cases of our contributions are proposed as application-layer services (e.g., service provisioning in the cellular networks and mobile number and profile portability), the complete use-case scenario of fully distributed architecture is proposed at the core network level.
- Detailed description of the deployed testbed for the assessment of the system performance, and the challenges, and problems of its real implementation in lab-scale.

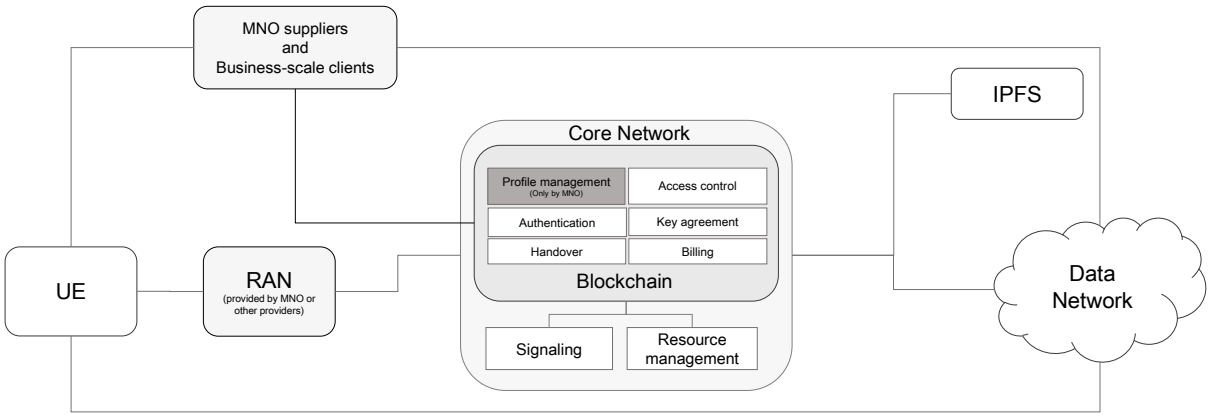
7.2 Possible implementation scenarios

To implement the proposed architecture in the real world, logically three scenarios can be used for positioning the Blockchain in the system. Fig. 7.1 (a-c) depict following scenarios. Table 5.1 provides a brief description of the feasible implementation scenarios and their pros and cons.

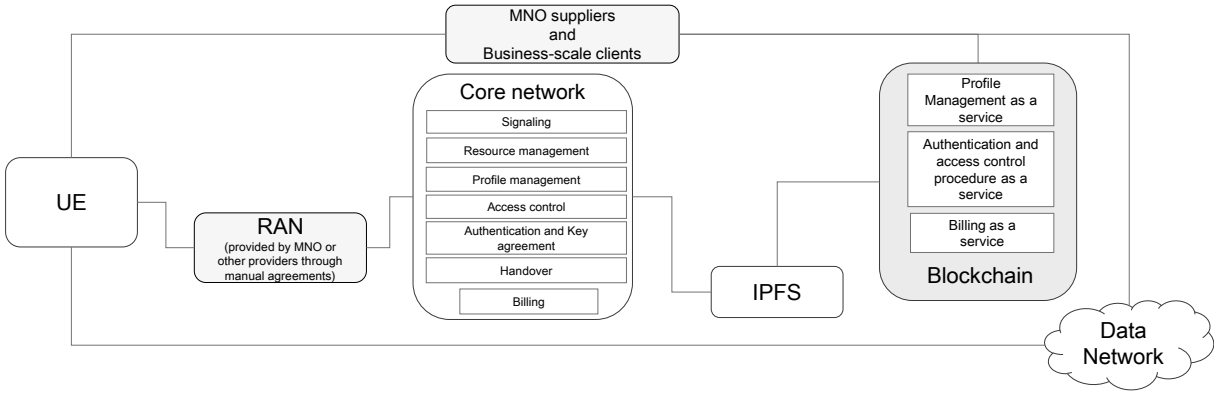
- ***Blockchain in the RAN or at the Edge***: In this scenario, as depicted in Fig. 7.1 (a), Blockchain can be logically positioned either between the user and RAN (for some functions such as AAC, key agreement and handover) or between RAN and Core Network. So, the user's connection request would be processed outside of the core network and before that on the edge. Note that, in this work, we did not consider this scenario, so, for the rest of the chapter, we are not going to explain this scenario.
- ***Blockchain in Core Network:*** In this model of implementation, as depicted in Fig. 7.1 (b), the core network functions should be adjusted to support Blockchain-based authentication, access control, key agreement, profile management, handover, and payment procedures. This scenario is the full decentralized and distributed version of the proposed method that needs to be



(a)



(b)



(c)

Figure 7.1: Possible implementation scenarios: (a) Blockchain at the RAN, (b) Blockchain in the core network, and (c) Blockchain in the application layer.

standardized by the standardization bodies before deployment. In this scenario, the external connectivity providers need to provide a compatible module to be connected directly to the core network and the Blockchain. External service providers can benefit from the application layer services provided in Blockchains such as payment and AAC (e.g., they can use the core network's AAC procedure to authenticate their users instead of doing it internally). It is important to mention that **implementation of this scenario with the existing hardware (i.e., specifically existing sim-card) and the software (e.g., existing open-source private cellular networks such as Open Air Interface -OAI) is not a feasible task.**

– ***Blockchain in Application layer***: This deployment model, as depicted in Fig. 7.1 (c), positions Blockchain after the core network in both network and application layers. **This scenario is the presentation of the applicability of using Blockchain in Beyond 5G networks**. Indeed, since in this solution, the Blockchain is positioned in service-layer, handling of the core network functionalities such as handover, registration, etc. is not possible to be done in Blockchain. So, the core network architecture of the MNOs will remain intact. In case of application layer requests, such as using distributed AAC services provided by MNO for external service providers, connection to Blockchain can be done directly from SP. For the external service providers, this scenario is an easy-apply solution for different use cases such as AAC and trusted payment.

7.3 Deployed Testbed

To deploy the proposed architecture we designed a use case based on Fig. 7.1 (c). It is important to mention that, the full implementation of the other scenarios, currency, is not possible using the existing tools (i.e., private cellular network tools such as Magma or OAI) and SIM cards. So, in order to evaluate the performance of the system and the possibility of implementation of the procedure using smart contracts, we implemented the scenario of Fig. 7.1 (c).

A high-level schematic of the implementation is depicted in Fig. 7.2. To simulate the cellular network, we employed OAI (Open Air Interface) consisting of RAN (OAI-RAN) and core network (OAI-CN) [230], and the Blockchain part is simulated using ganache-cli. Following, we are going to explain each part separately (i.e., cellular network and Blockchain).

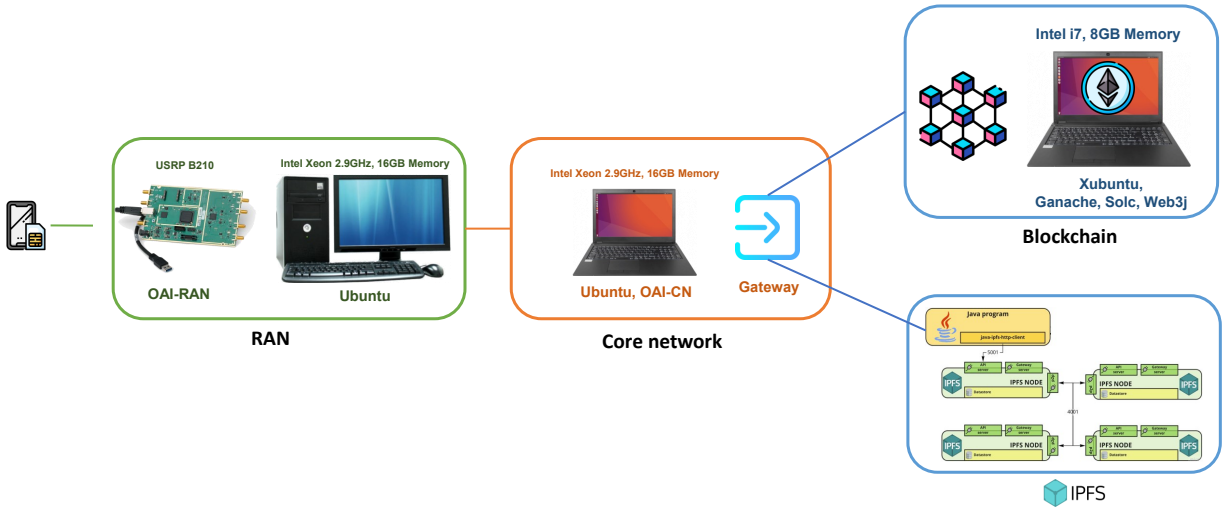


Figure 7.2: The schematic of testbed environment.

7.3.1 Private cellular network deployment

To launch the evaluation environment, we simulated the radio access network and the core network parts of the cellular network using OAI [230], which is an open-source platform implemented to support the deployment of small-scale mobile telecommunication access, network, and core solutions for 4/5G based on the 3GPP standard. OAI offers an open-source software-based implementation of the LTE system for 4G. This platform can be used to build and customize an LTE base station (OAI-eNB), User Equipment (OAI-UE), and core network (OAI-EPC) on personal computers. OAI has software-based network functionalities, reducing the implementation cost and increasing the flexibility of the deployment, allowing lab scenarios to test their proposed solutions [231]. Fig. 7.3 depicts the schematic of the deployed private cellular network. The list of hardware and software used during testbed deployment are described in Table 7.1. The information required to program the SIM card (to simulate the user) is shown in Table 7.2, and Table 7.3 describes the *eNodeB* information modified in `enb.band7.tm1.50P RB.usrpb210.conf` configuration file to connect RAN to the core network.

To build the RAN part (i.e., the network provider's base station), the OAI-RAN was executed on a PC with USB3 and Gigabit Eth. We used a `USRP B210` (i.e., an SDR (software-defined radio) board for radio communications) board for radio communications that connects to the PC through the USB3 interface. A similar PC configuration is used to launch the core network using OAI-CN.

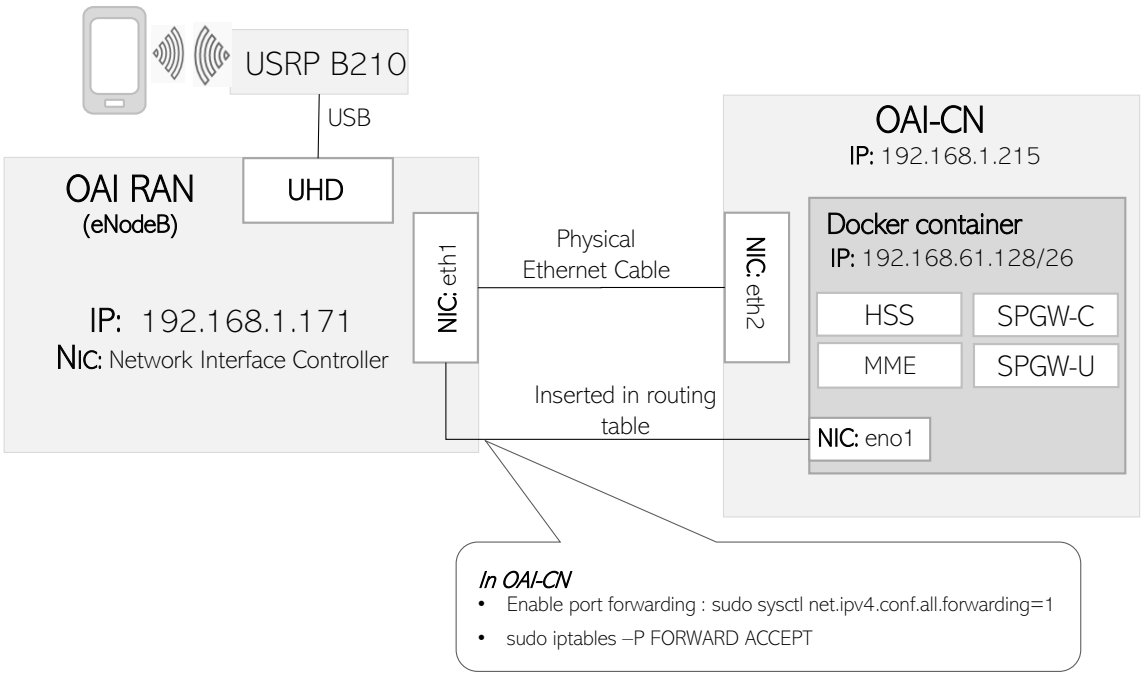


Figure 7.3: Architecture of deployed private cellular network ($Smartphone \leftrightarrow OAI - RAN + OAI - CN$).

7.4 Blockchain implementation

After deployment of the private cellular network and connecting the commercial device to this network, we need to implement the Blockchain and the distributed database parts of the implementation. To do so, In our simulation, the OAI-CN remained intact (i.e., the MNO core network would stay intact), instead, we provided a gateway that sends the user's requests to the Blockchain (i.e., private Ethereum) and IPFS (i.e., through a dedicated HTTP client) in the application layer.

To implement Blockchain we used Ganache-cli which is a Blockchain simulator, to simulate the behavior of Blockchain in the local system. This tool is a widely used and mature tool to test smart contracts and the behavior of the Blockchain network. This tool runs one full node (which is the Ganache application on the local device) on top of the local private Ethereum Blockchain with a PoW consensus model. Using this node there is a possibility of setting configurations for different parameters, which are the performance indicators of the Blockchain network. For example, configuring the block time, in the granularity of seconds, gives us the option of simulating the behavior of permissioned and permissionless Blockchains. Assume that we want to

Table 7.1: Environment specifications

Entity	Parameter	Specification
	Hardware	
OAI RAN	CPU	Intel Core i7-6700 CPU 3.40 GHz
	RAM	16 GB
	SDR board	USRP B210
OAI-CN	CPU	Intel Xeon W-2102 CPU 2.90 GHz
	RAM	16 GB
UE	Smartphone	Samsung S4
	SIM-card	Sysmocom SJS1
	SIM-card reader	Gemalto
Blockchain	CPU	Intel i7 Dual-core 1.6GHz
	RAM	6 GB
	Hard Disk	128GB SSD
	Software	
OAI RAN	OAI-RAN	master branch release v1.1.0
	OS	Ubuntu 18.04-low latency kernel
	UHD	v4.1.0.0 branch
	pysim	master branch
OAI-CN	OS	Ubuntu 18.04 Bionic
	OAI-CN	master branch
Blockchain	OS	Xubuntu
	Ganache-cli	6.12.2
	Ganache-core	2.13.2
	Web3j	1.4.1
	Solc	0.8.2

simulate a private Blockchain; In this case, it is highly possible that we need to set the block time to a highly lower number than when we want to simulate a public Blockchain.

Apart from the Blockchain network, we used the Solidity language to write our smart contracts. Solidity is an object-oriented and Turing-complete programming language for implementing smart contracts that was implemented by Ethereum's core contributor for the first time in 2014. Programs in Solidity run on Ethereum Virtual Machine (EVM) or on compatible virtual machines. To compile the Solidity codes to ABIs, we used the Solc compiler. Moreover, in order to connect to the smart

Table 7.2: SIM card configuration

Parameters	Values
ADM key	2611488
MCC (Mobile Country Code)	208
MNC (Mobile Network Code)	93
Name	OAI
IMSI	208930000008
Ki	8baf473f2f8fd09487
OPC	8e27b6af0e692e750f32667a3b14605d
ICCID	8988211000000285877
After programming the sim card and inserting it into the mobile phone, a custom Access Point Network (APN) information with name and APN variables are set with value *oai.ipv4* in the smartphone.	

[!t]

Table 7.3: RAN configuration

Parameters	Values
MCC (Mobile Country Code)	208
MNC (Mobile Network Code)	93
ipv4 of mme_ip_address (OAI-CN)	192.168.61.149
ipv4 of mme_ip_address (Magma)	192.168.61.149
ENB_INTERFACE_NAME_FOR_S1_MME	eno1
ENB_INTERFACE_NAME_FOR_S1U	eno1
ENB_IPV4_ADDRESS_FOR_S1_MME	192.168.1.215
ENB_IPV4_ADDRESS_FOR_S1U	192.168.1.215
Duplexing mode	FDD
Frequency band	Band 7
Physical Resource Block(PRB)s	50

contracts, send concurrent transactions to the Blockchain, simulate the light nodes (i.e., those that only send transactions to the Blockchain but are not participating in consensus), and test the GAS consumption of smart contracts, we used the Web3j library. This library compiles the ABI files into $.java$ classes, making it usable similar to the other Java classes in this language.

7.5 Summary

In this chapter, we proposed three deployment scenarios in which the Blockchain can be logically positioned either in RAN, core network, or service layer. We implemented the third solution (i.e., positioning the Blockchain at the service level) and connected

the core network of the private cellular network to the Blockchain to assess the performance of different contributions. Moreover, some parts of the evaluation (e.g., authentication and key agreement, handover) are implemented and evaluated only in Blockchain and using virtual users.

Chapter 8

Evaluation

Contents

8.1 Introduction

In this section, we evaluate the performance and security of different procedures described in Chapter 6 (i.e., subscription, registration, key agreement, handover, key agreement, and payment). The evaluations are done in two following parts to address the mentioned requirements:

- Scalability evaluation: The scalability of the system can be defined as changes in throughput or latency when altering a parameter in the system. We assess the scalability of the system in terms of the increasing number of concurrent requests. In other words, to see if the system can provide stable latency/throughput while increasing the number of users in the system, regarding different parameters. Scalability assessment can state how the proposed method is able to address ***R5*** and ***R1*** (regarding the system's ability to handle broader coverage with the higher number of users).
- Security analysis in which the system's security is analyzed regarding network-based attacks and smart contracts defects. This evaluation addresses the requirements ***R2, R3, and R6***.

8.2 Scalability analysis

scalability analysis of the method has been done by evaluating the tolerance of the system against increasing the number of transactions. In other words, **we want to see how many transactions we sent to the Blockchain, and the average time for transaction processing remains almost stable. Note that, the mentioned latency in the following chapters, does not represent the user's or MNO's experienced latency. It indicates the stability of Blockchain system regarding the increasing number of transactions.** An example of user-experienced latency, is provided in Appendix D. As defined in [232], the scalability of a system is the changes in throughput or latency while altering one/several other parameters. If these factors stay almost stable regarding the alteration of parameters, we can say that the system is scalable. Having a scalable system, regarding the number of users, is crucial in cellular networks.

To assess the scalability of the system, the following parameters are adjusted in our assessment [233, 234]. Fig. 8.1 to Fig. 8.6 depict the latency of the system for

different BS, BT, and C, and Table 5.5 lists the results of throughput assessment for the same parameters.

- Block size (BS): The number of transactions fitting into one block ($BS \in \{15, 30\}$).
- Block time (BT): The difficulty of consensus puzzle which results in the extraction of blocks in predefined time ($BT \in \{5, 10, 15\}$).
- Concurrent requests (C): Number of the virtual clients in the system that sends their request concurrently ($C \in \{50, 100, 200, 500, 700, 1000\}$).

The latency and throughput are calculated as:

$$Latency = \frac{t_f - t_s}{|Tx|}$$

$$Throughput = \frac{|Tx|}{t}$$

where t_s is the time of starting to send all concurrent requests, and, t_f is the time of finishing receiving the transaction receipt for all of them, $|Tx|$ is the number of transactions, and t is the total time of execution.

8.2.1 System Scalability for core network functions

As it is shown, for $C \geq 100$ the latency of the system stays almost stable while altering BS, BT. It implies that increasing the number of users (i.e., light nodes with sending transactions) in the system would not negatively affect the overall performance. So we can claim that the system is scalable regarding the number of users. Indeed, the real number of users in cellular networks is significantly more than 1000, but because of the limitation of the $web3j$ library, we restricted the number of concurrent requests. Due to the high scalability of the real-world implementation of the Blockchain network, we can conclude that this system has the same basic features.

To analyze the impacts of using Blockchain in the various procedures, we can examine the system latency/ throughput for different BT, and BS. As shown in the figure, increasing the BS and lowering the BT, result in declining the latency (increasing the throughput). So, we can conclude that decreasing the complexity of

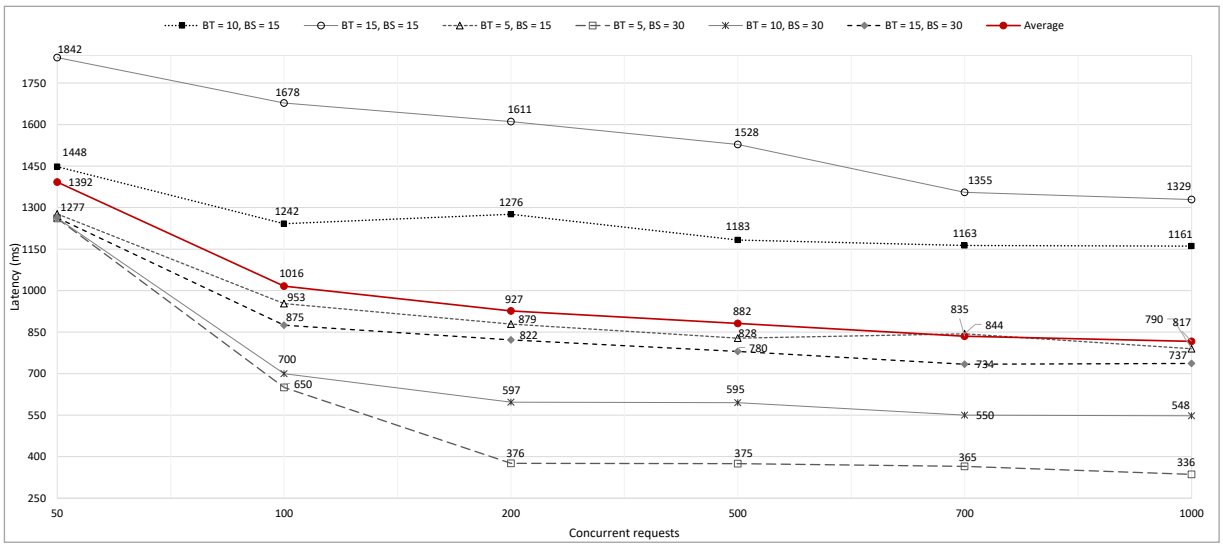


Figure 8.1: System latency with different values for BT and BS in several concurrent requests for User subscription.

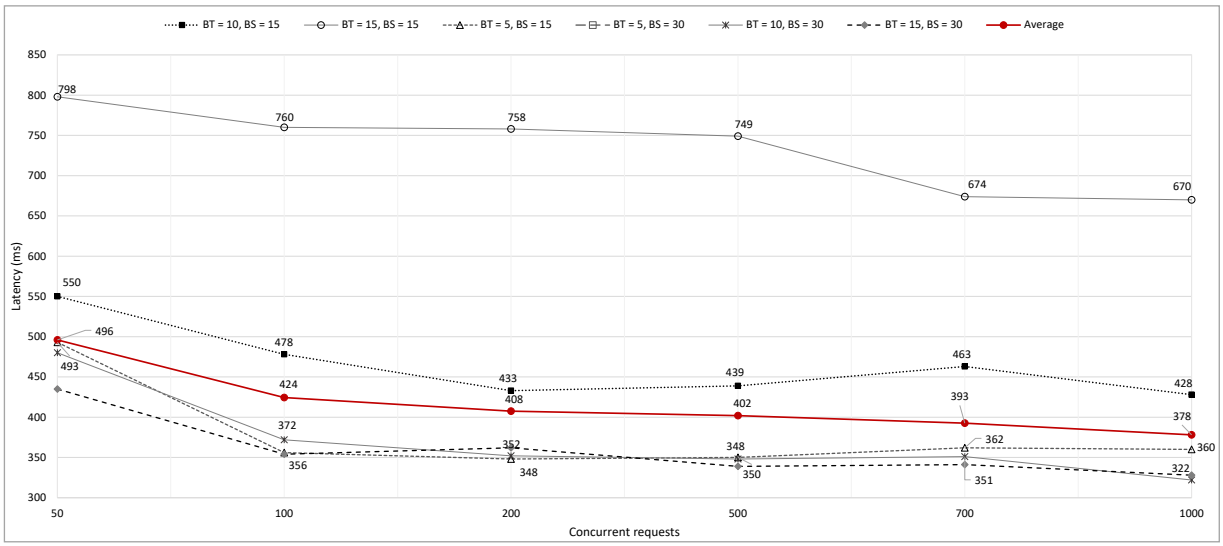


Figure 8.2: System latency with different values for BT and BS in several concurrent requests for User registration in the system and key agreement (BC-AKA).

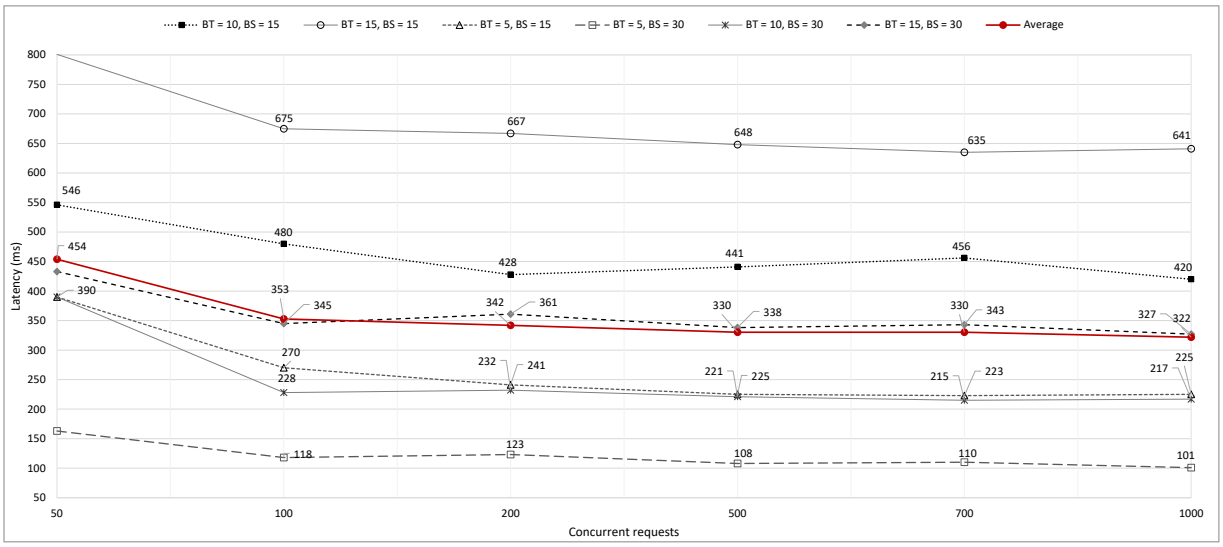


Figure 8.3: System latency with different values for BT and BS in several concurrent requests for User access control.

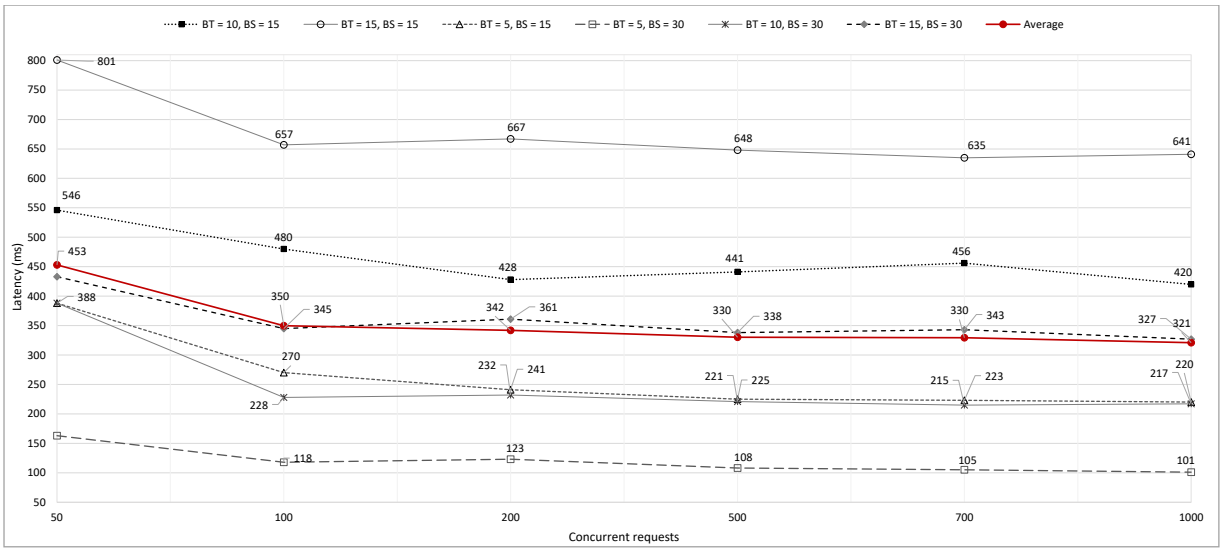


Figure 8.4: System latency with different values for BT and BS in several concurrent requests for mobility management and handover process.

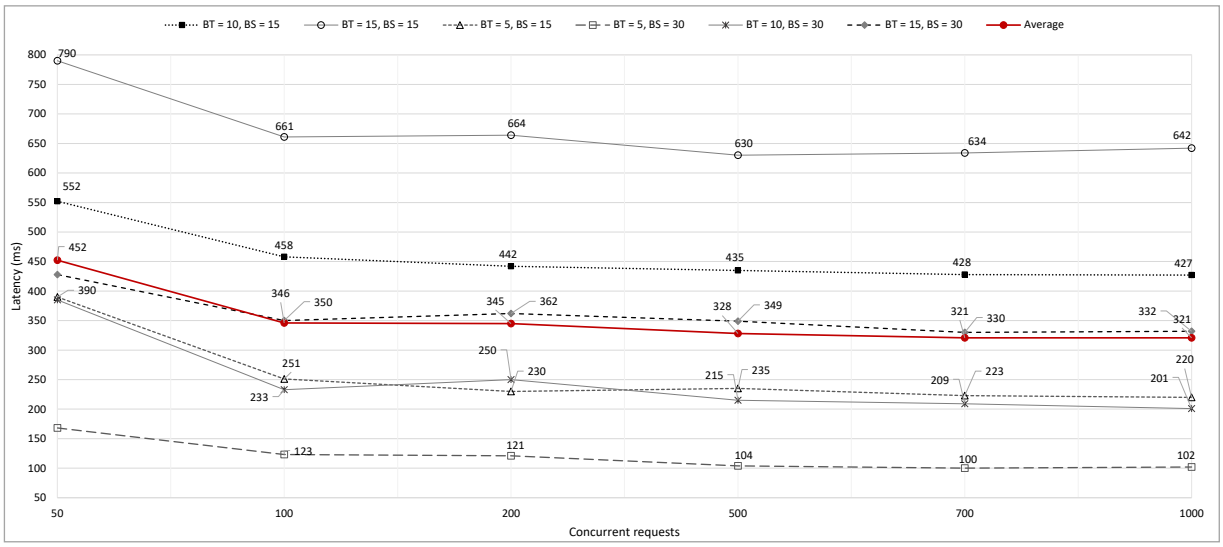


Figure 8.5: System latency with different values for BT and BS in several concurrent requests for session management process.

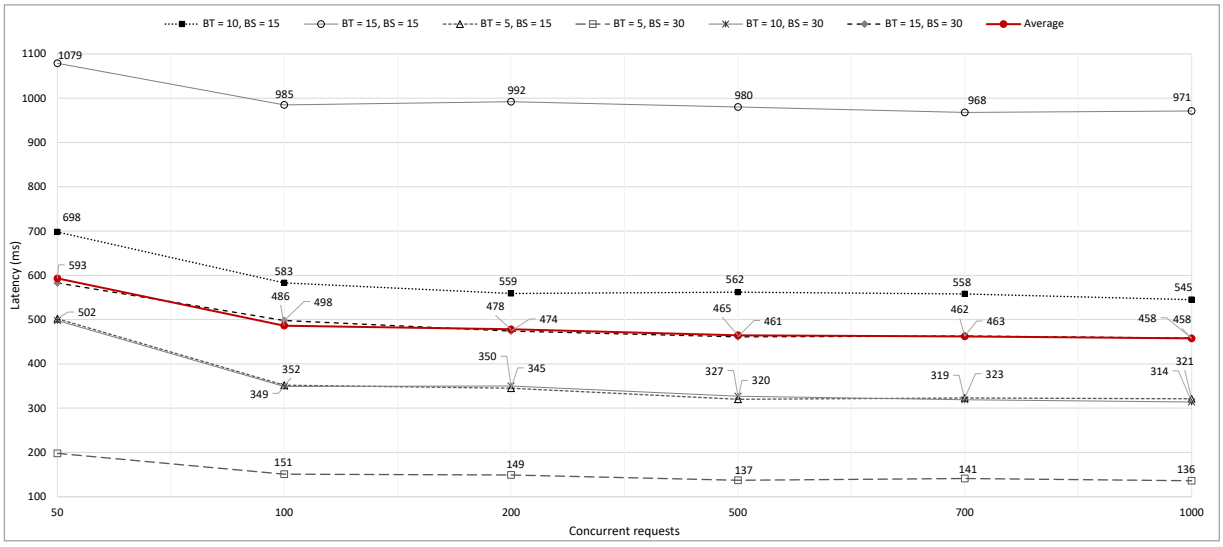


Figure 8.6: System latency with different values for BT and BS in several concurrent requests for payment process.

Table 8.1: System throughput with different parameters. BT (s), Throughput (transaction per second (tps))

P	***Registration***					***Access control***					***Handover***					***Payment***				
BS	*15*		*30*			*15*		*30*			*15*		*30*			*15*		*30*		
BTC	5	10	5	10	*Avg*	5	10	5	10	*Avg*	5	10	5	10	*Avg*	5	10	5	10	*Avg*
50	2.03	1.25	4.55	2.08	**2.01**	1.64	3.21	1.07	1.64	**1.62**	1.64	3.22	1.09	1.64	**1.63**	1.64	3.19	1.10	1.64	**1.64**
100	2.81	1.32	4.42	2.69	**2.33**	2.17	3.21	1.30	2.16	**1.99**	2.22	3.22	1.30	2.17	**2.01**	2.18	3.23	1.30	2.17	**2.00**
200	2.87	1.32	5.21	2.84	**2.42**	2.14	3.23	1.31	2.12	**1.99**	2.20	3.85	1.30	2.15	**2.05**	2.18	4.18	1.30	2.16	**2.07**
500	2.86	1.34	5.35	2.87	**2.45**	2.40	3.85	1.30	2.39	**2.15**	2.38	3.85	1.30	2.40	**2.15**	2.29	3.82	1.30	2.39	**2.12**
700	2.76	1.48	6.10	2.85	**2.58**	2.32	3.61	1.33	2.29	**2.11**	2.35	3.64	1.32	2.30	**2.12**	2.36	3.79	1.29	2.29	**2.11**
1000	2.78	1.49	6.25	3.11	**2.65**	2.47	3.92	1.33	2.25	**2.16**	2.53	3.70	1.33	2.27	**2.15**	2.50	3.89	1.30	2.28	**2.14**

the consensus puzzle and increasing the number of transactions fitted into one block would significantly enhance the performance of the system. For example, the average delay for $BS = 30$, $BT = 5$ is the lowest amount. But, if the participated nodes in the consensus procedure are not trusted, this configuration would be problematic. Because decreasing the block time results in an easier consensus puzzle, that would increase the risk of integrity violation in the PoW consensus model and the system's vulnerability against a different type of Blockchain-based threads (e.g., 51% attack [235]). Based on our assumption, all the nodes are trusted, so, we can omit this risk (and decrease the BT). On the other hand, for $BS = 15$, $BT = 15$ the system performance is not acceptable, but it is in the most secure condition. These results indicate that the system performance is adjustable based on a compromise between security and performance.

It is important to mention that the performance results depicted in Fig. 8.1 to Fig. 8.6 are the overall latency of the procedure, not the latency that the user may experience. As an example of an access control procedure, before giving access to the user, no transaction would change the Blockchain state (i.e., it means all the transactions are function 'call'). So, the consensus procedure will not be executed for them, and as a result, the user will not experience the aforementioned latency. After giving access to the user, the access log, payment, access state, etc. would be recorded in Blockchain, which needs the execution of a consensus procedure.

8.2.2 System Scalability for Mobile number and Profile porting procedure

To evaluate the proposed method, we simulated the whole procedure of mobile number and profile porting in the private Ethereum Blockchain (details are provided in Table 7.1). Following, we provided the performance analysis of the proposed method by evaluating the scalability of the system in terms of the increasing number of concurrent requests. Fig. 8.7 (a-c) depicts the latency of the system for the aforementioned configurations BS and BT. Based on the definition of scalability, if the latency/throughput stays almost stable regarding the alteration of parameters, we can say that the system is scalable [233, 234]. As shown in the figure, system latency is almost stable for $C \geq 200$. Therefore, we can claim that **the system is scalable** and can maintain adjustable and low latency in a large-scale request environment for user subscription, porting, and termination procedures.

Moreover, Table 5.6 provides the throughput of the system in different Blockchain

configurations. As shown in the table, increasing the BS and decreasing the BT can positively affect the performance by increasing the overall throughput. For instance, compare the throughput for $BT = 10, BS = 30$ (i.e., the highest complexity of consensus due to minimum trust in the network, and lowest number of transactions fit in each block) and $BT = 5, BS = 100$ (i.e., highest trust and highest number of transactions in each block). Note that, an important issue to select the configuration is the trust level among participants in the network (i.e., decreasing the block time results in an easier consensus puzzle, which can bring the risk of integrity violation in the system).

Note that in the configuration of ($BT = 15, BS = 30$) the Web3j library threw several time-out exceptions in concurrent requests 500 and 700. So, exceptionally, we do not report the system latency for this configuration.

8.2.3 System scalability regarding storage

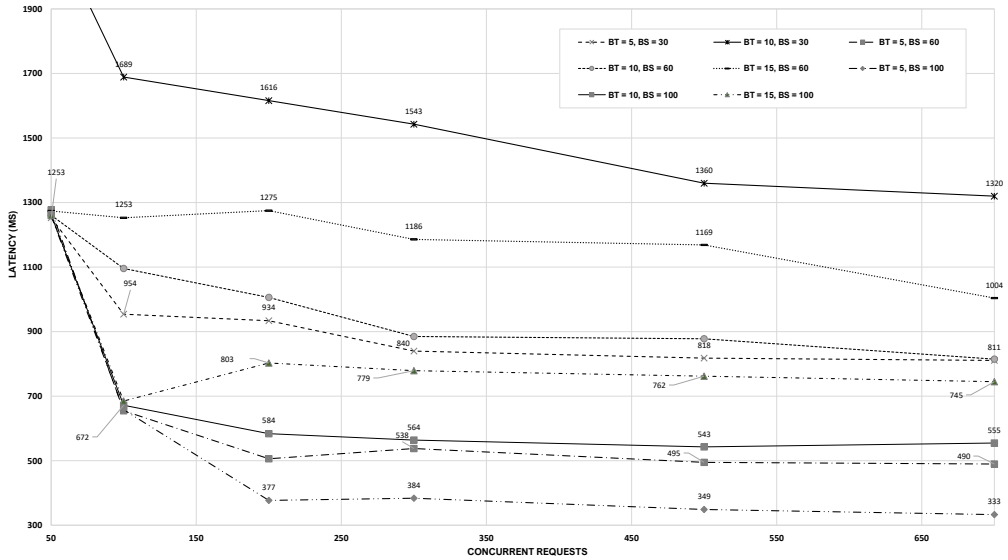
Blockchain is an append-only ledger in which there is no possibility to remove/ change the published blocks or validated transactions. In means, that the size of the ledger would be grown based on the number of generated/stored transactions. So, producing a high number of transactions for validation of a request results in an intense increase in the required storage of participated nodes.

Note that there are two types of function calls in the Blockchain. `Call` is the operation of reading a value from the ledger (without changing its state), while `Transactions` would alter the Blockchain state by modifying variable(s). `Transaction` execution in the Blockchain, not only requires the consensus procedure (increases the latency) but also needs to be recorded in the network, which results in increasing the size of the ledger. So, to decrease the storage consumption, we only need to reduce the number of `Transactions`.

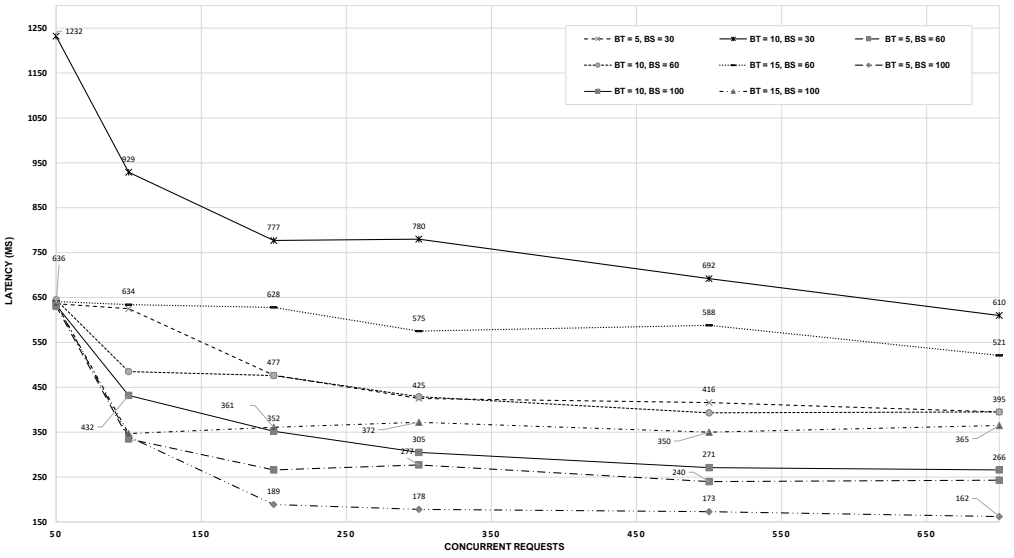
In the current version of the proposed method, the user registration procedure would create one transaction, so its complexity is $O(n)$. For the access control, handover, session management, and payment, they produce $n \times 1$, $n \times 2$, $n \times 1$, and $n \times 3$ transaction, respectively (so, their complexity is also $O(n)$). In this version, the storage performance is not optimized, especially for payment procedures. This issue and its possible solutions are discussed in the following sections.

Table 8.2: System throughput with different parameters. BT (s), Throughput (transaction per second (tps))

P	***Porting***								***Termination***							
BS	*30*				*60*				*100*				*30*			
BT / C	5	10	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15
50	0.79	0.46	0.78	0.79	0.78	0.79	0.79	0.79	1.57	0.81	1.58	1.55	1.55	1.56	1.57	1.56
100	1.04	0.59	1.52	0.91	0.79	1.51	1.48	1.45	1.59	1.07	2.97	2.05	1.57	2.94	2.31	2.87
200	1.06	0.61	1.97	0.99	0.78	**2.65**	1.7	1.24	2.09	1.28	3.74	2.1	1.59	**5.28**	2.84	2.76
300	1.19	0.64	1.85	1.12	0.84	**2.59**	1.8	1.28	2.34	1.28	3.6	2.32	1.73	**5.6**	3.27	2.68
500	1.22	0.73	2.01	1.13	0.85	**2.86**	1.84	1.31	2.4	1.44	4.15	2.53	1.69	**5.75**	3.68	2.85
700	1.23	1.01	2.03	1.22	0.99	**2.99**	1.79	1.34	2.52	1.63	4.1	2.52	1.91	**6.13**	3.75	2.73



(a)



(b)

Figure 8.7: System latency with a different configuration of BT and BS in several concurrent requests for mobile number and profile porting procedure (a), and subscription termination procedure (b).

8.3 Security analysis

In this section, we provide a brief discussion on the security of the system against several threat scenarios. In this regard, we separated the attack models into two

main categories:

i. Network-based attacks: These types of attacks are common in Blockchain-based and non-Blockchain-based systems. In these types of attacks, the attacker aims to steal the user's identity, accelerate their privilege, break down the system by committing denial of service attacks and making the system disabled to serve the eligible user, etc.

ii. Blockchain-based/smart contract-based attacks: These types of attacks are specific to the Blockchain-based systems and generally attackers try to forge the smart contracts to steal money, eliminate the system's trustworthiness, etc.

8.3.1 Network-based attacks

In this section, we introduce some of the most highlighted network-based attacks regarding authentication access control procedures. Note that, the assessment of all attack scenarios is out of the scope of this work. So, in the following sections, we analyzed the system's capability to resist different threats that are more evolved with the access to the user's or businesses' private data, data leakage, the confidentiality and integrity of rules and data, and the system's availability.

8.3.1.1 Man-in-the-Middle attack (MitM)

MitM attacks refer to the type of attacks in which an adversary position himself in a conversation between two parties to passively eavesdrop, or actively eavesdrop and modify passing information. The goal of MitM is to steal significant data that can be used for further attacks. Possible thread scenarios and proposed preventive solutions are discussed below:

– ***Scenario 1: Unauthorized subscription-*** In subscription process, after sending *code* by u ($Step-1$, Fig. 6.1), adversary A gets the encrypted data by eavesdropping the network. So, A can repeat this data and send it to SC_{Sub}. After that, A can subscribe to the network on behalf of u, fill out the inquiry form, and prevents the legitimate user from accessing the network.

Analysis: To subscribe in the system, U needs to generate a one-time random number (i.e., *nonce*), and send its hash (i.e., $Hash(nonce)$) to SC_{Sub} in $Step-1$, Fig. 6.1. $Hash(nonce)$ is stored in an immutable Blockchain as proof for the

user's claim in the off-chain subscription request. Then, in $Step-3$, U sends the *nonce* in plaintext to assure MNO that the legitimate user is the one who created both $Step-1$ and $Step-3$ requests. *nonce* is a strong secret random number generated by the user, and the hashing is a one-way function. So, since subscriber sends the $Hash(nonce)$ in the first request, A cannot find the *nonce* for $Step-3$. Therefore, the MitM attempt will fail in this scenario.

– ***Scenario 2: Unauthorized registration-*** In user registration phase, assume that user sends $EN^{Pr_U}_{Addr_U}$ and $Addr_U$ as its identifier (instead of $EN^{Pr_U}_{Hash(nonce+1)}$). Due to the intrinsic authentication capability of asymmetric encryption (i.e., $EN^{Pr_U}_{Addr_U}$ can only be generated by a legitimate user), in an ideal case, without an attacker, the registration scenario would execute safely. But assume that adversary A actively eavesdrops on the connection between the user and a CP. In the next step, A can repeat these data and send it to CP for further registration requests.

Analysis: To address this issue, as shown in Fig. 6.2, $Step-2$, we employ a challenge-response procedure using a random number (*nonce*) encrypted by Pr_U (i.e., the user's private key that only belongs to the legitimate user). So, if the attacker succeeds in reading the message, replying to the *nonce* challenge, needs the knowledge of Pr_U. Note that, *nonce* is a one-time random challenge that would be changed for the user's next registration. So, we can state that this attack will fail under the aforementioned assumptions. Another challenge in this procedure, is transmission of $EN^{Pr_U}_{Hash(nonce+1)}$. This message can be eavesdropped on and decrypted by A who aim to use it for further steps of unauthorized registration. This attack will also fail because, in $Step-5$, CP would encrypt the message with K_M that only eligible user has it. So, its verification, and the creation of acknowledge message with the agreed session key can only be done by the eligible user.

– ***Scenario 3: Forged connectivity/service provider-*** In the user registration phase, assume that an adversary CP forge itself as an eligible CP. So, if CP is successful in running the attack, it would be able to position itself in the connection between the user and CP for the rest of the user's connection.

Analysis: To address this issue, rather than using challenge-response procedure that is mentioned in previous scenario, CP needs to encrypt $Hash(nonce+2)$ with K_M in $Step-6$. It means, even if CP tries to generate nonce in each step (instead of CP) and send it to the user, in $Step-6$ he would not be able

to encrypt in with K_M which is securely stored in distributed database using hybrid cryptosystem, and the user's SIM-card. So, when in $Step-7$, user aims to validate $Hash(nonce+2)$, it would not match to the *nonce* which is created by CP. Note that, in $Step-2$, $Hash(nonce+1)$ is encrypted by PR_U, So, the user's identity would be proved for the CP. But, before key-agreement and mutual authentication procedure, the Pub_{CP} is not shared with the user. So, in $Step-6$, CP cannot use its private key to approve itself to the user. Instead, CP in $Step-6$ uses K_M.

– ***Scenario 4: Forged connectivity/service provider in Handover-*** In the handover phase, assume that an adversary CP forge itself as an eligible CP. So, if CP is successful in running the attack, it would be able to execute Scenario 3.

Analysis: To address this issue, as shown in $Step-4$ Fig. 8.4, a random number *nonce* is sent to the user in encrypted form. So, only the user can decrypt it with PR_U. Moreover, in other authentication steps (i.e., *Steps* 6, 11, 12), $Hash(nonce+x)$ must be sent by the user or CP_t. Since all these variables in all steps are encrypted by the public key of the recipient, it is not possible for CP to find their plaintext and use them for further authentication.

8.3.1.2 Confidentiality and privacy

Since in the proposed method we use distributed database (i.e., IPFS) to store user's data, it is inevitable to provide strong confidentiality and privacy solution to avoid data leakage.

Scenario 5: User data confidentiality- Assume that adversary A or forged connectivity provider CP gets the $CID_{EN^M_{K_s}}$ from SC_U. On one hand, retrieving the user's data from IPFS by CP would be a confidentiality breach (i.e., user's sensitive data leakage), on the other hand, the legitimate CP needs to have access to the data.

Analysis:To address this issue, we proposed a hybrid cryptosystem solution for two categories of subscriber data consumers (i.e., main MNO and small-scale connectivity/service providers). Regarding the data storage, the user's data is encrypted by a secret symmetric key (K_s), and this key is also encrypted in an asymmetric model with the user's and MNO's public key (i.e., $EN^{K_s}_{Pub_{MNO}}$ and $EN^{K_s}_{Pub_U}$). Since A doesn't know Pr_U or Pr_{MNO}, he would not be able to decrypt the downloaded data. Moreover, for eligible CP, the user's secure authentication is required. In this regard, we

proposed the generation of an authentication claim by MNO in which none of the user's private data is needed to be revealed.

8.3.1.3 Data tampering

Data tampering refers t thread scenarios in which any entity in the system tries to modify a user or other network entity's data in an unauthorized manner. Following the thread scenarios and solutions in this regard are provided:

- ***Scenario 6: User status alteration in subscription-*** In the subscription process, after deploying the user's contract, SC_{Sub} changes the user's status to `active-noData`. If adversary A (e.g., a malicious entity in the network) can alter this status, it can activate the user before any verification, or in the opposite case, it can prevent the user from being active in the network (regardless of successful user verification).

  Analysis: To address this threat scenario, we defined SC_{AB} that records the address of the single handler contracts such as SC_{Sub}. Moreover, due to the capability of access restriction on smart contract functions, `Update_St()` function of SC_{UL} is only can be called by SC_{Sub}, i.e., it verifies: $msg.sender == SC_{AB}.getAddr("sub")$. In which `"sub"` is a predefined identifier for SC_{Sub} in SC_{AB}. So, due to the immutability of SC_{AB}, and having the signature of the sender in each transaction, it is impossible for A to change the user's subscription status in SC_{UL}.

- ***Scenario 7: Tampering user data-*** Adversary user U aims to alter his data stored in IPFS.

  Analysis: As shown in Fig. 6.1, the user's data can be updated in SC_{Sub}. On the other hand, one of the entities that have access to the user's data is the user himself. So, the adversary users can alter their data in their contracts and IPFS. To address the data alteration issue, we used the modifiers on SC_U to limit the entities that can alter the data. For this purpose, firstly we define an `owner` in the user's contract, who is eligible to update the `CID` in the user's contract. SC_{Sub} handles the definition of `owner` and delegates the write permission to it after verifying the role of the requester. For the role of `user` in the system, this delegation would not be assigned. Finally, `Update()` function in SC_U verifies: `msg.sender == owner`. So, U doesn't have the alteration right to modify the data.

– ***Scenario 8: Tampering smart contract addresses-*** Adversary A aims to alter the address of agreed smart contracts in SC_{AB} to advertise forged contracts.

Analysis: As mentioned before, to defeat the maintainability defect of smart contracts, SC_{AB} is designed to store the address of handler contracts (e.g., SC_{Sub}, SC_{AAC}, etc.). Modification of existing addresses in this contract can breach all security assumptions of the system. To resolve this problem, we used the intrinsic access control mechanism in smart contracts. It means we defined a list of `owners` who are eligible to execute `Update()` function in SC_{AB} (i.e., a `modifier OnlyOwners` is defined in `Update()` function that verifies the validity of the transaction sender).

8.3.1.4 Denial of services

A Denial of Service (DoS) attack means making a resource (e.g., data, device, service) inaccessible to a legitimate user.

– ***Scenario 9: Single point of failure-*** One reason that can result in the vulnerability of the system against DoS attacks is having a single point of failure either in data storing (i.e., centralized storage), service provisioning (i.e., centralized server), or process management.

Analysis: As shown in Fig. 6.1, Fig. 6.2, and Fig. 8.4, in the proposed architecture, subscription, registration, and mobility management are handled by smart contract instead of centralized servers. This means there are several nodes in the system to receive a request (transaction), validate it based on the policies in the smart contracts, reach a consensus on the validation result, update the ledger and send the result to the caller entity/smart contract. In this process, the failure of a single node does not have a significant effect on the functionality of the whole system. Rather than distributed servers, in the proposed method IPFS is used for the data and policy management procedure. So, we can state that there is no single point of failure regarding database access and the system is resistant to the DoS threat.

8.3.1.5 User privacy

User privacy can be violated in case of any leakage in their data, identity, and location [30,31,236]. One of the intrinsic characteristics of Blockchain is its *metadata privacy-*

preserving [27]. It means the real-world identities of the senders and receivers of transactions (i.e., users and providers in our use case) are masked using a random address. In our proposed method, the user privacy requirements are addressed as follows:

- None of the user's identifiable data, such as $SUPI$ in current 5G networks, is transmitted, in clear text, during connection requests.
- None of the user's personally identifiable information is stored in Blockchain or transmitted in transactions.
- The user information is stored in IPFS using a hybrid cryptosystem method, that only allows legitimate clients (i.e., user and MNO) to have access to these data.
- Blockchain is implemented in the core network, so, outside intruders do not have access to any of the internal Blockchain transactions. They only can actively/passively eavesdrop on the message passing between the user and RAN, which are encrypted, or other security solutions are adapted for that.
- The *content* privacy is not offered in Blockchain. It means that if any PII data is sent in the transaction in its body, it would be accessible to all participating nodes. To overcome this problem, we stored the minimum required non-PII data to provide connectivity and payment for the user.

8.3.2 Blockchain-based threats: Maintainability of smart contracts

Due to Chen et. al [221] since smart contracts are not updatable after deployment, there are two defects in using smart contracts regarding maintainability:

- ***Scenario 10: Hardcoded addresses-*** Having hard-codded addresses in smart contracts results in the non-flexibility of the system in case of finding any vulnerability or changing the business needs.

  Analysis: There are several controller smart contracts that may need to be updated based on business needs or patching vulnerabilities. To manage the maintainability of these contracts we defined SC_{AB} which stores the latest addresses of the controller contracts in the system. So, instead of hard coding

the addresses in other contracts, we use SC_{AB} as a reference to find the contract address. Note that, calling `getAddress()` function of SC_{AB} will not change any state of the Blockchain (i.e., in Ethereum referred to as `pure` function), so, it doesn't have a significant effect on the latency of the system.

– ***Scenario 11: Missing Interrupter-*** When bugs are detected in the smart contracts by the attackers if these smart contracts are used for payment (i.e., `payable`), the attacker is able to steal money.

Analysis: In the proposed method, SC_{AAC} an SC_U are `payable` type contract. So, to avoid the missing interrupter defect, in both contracts `selfdestruct()` function is defined to force the contract to kill itself in case of accruing an attack (i.e., smart contracts contain an interrupter on suicide function). Then the money would be transmitted to the owner's wallet (or given address).

8.4 Summary

In this chapter, we evaluated the performance of the proposed system regarding scalability, storage, and performance of different network functions (i.e., user subscription, authentication, and key agreement, access control; mobility management, session management, billing, mobile number, and profile porting, and subscription termination). The results of the evaluation show that the system is scalable, and based on the network requirements, its performance and security level are adjustable. Moreover, security analysis shows that the system is resilient against common threads for communication. In addition, the obstacles and limitations of real-world implementation of the novel architecture regarding latency, scalability, standardization, storage requirements, and incentives for different parties are discussed.

Chapter 9

Conclusion and Future Work

Contents

9.1 Summary

In this work, we first provided a futuristic scenario by explaining the expectations of different entities of the cellular network ecosystem. This helps us to define the validity domain of the proposed work. **MNOs in the future cellular networks** would be able to *collaborate* with other service providers, connectivity providers, and other MNOs in a secure, distributed, trusted, automated, non-stand alone and scalable environment. In this futuristic scenario, the **small-scale enterprise or other connectivity providers** would also be able to enter the cellular network market, participate in the system expansion and improvement, and can benefit from it to serve their users and increase their revenue by minimum investment in a short time and with a low agreement and operational cost. Moreover, **Service and content providers** will also be able to build a collaboration with MNOs and other providers, on top of existing capabilities of cellular networks such as authentication, access control, etc. to provide services for the users and other businesses and decrease their costs and operational complexities. Finally, this collaboration can result in improving the **end-user's** welfare and satisfaction by broadening the coverage, decreasing service costs, etc.

Considering this motivating use case for the beyond 5G networks, we derived the following requirements that need to be addressed:

R1 To increase the automation level of the IT procedure, contract/agreement, network functions, etc. handling to overcome the ever-growing complexity of next-generation networks and increase the collaboration among entities.

R2 To provide trust in a distrustful environment of unlimited numbers of service providers, connectivity providers, and MNOs which aim to collaborate in areas broader than their current businesses regarding services, regulation, payment, etc.

R3 To decentralize the network architecture avoiding putting the whole process, management, and authority loads in the hand of one entity. This requirement aims to overcome the management of the complexity of the next-generation networks, as well as improve security and performance.

R4 To guarantee system scalability regarding the increasing number of users, collaborators, providers, etc. in the system. Here, scalability means that by in-

creasing the number of users in the system, its performance is not experiencing high deviation from normal situations.

R5 To provide a high level of security regarding confidentiality, integrity, non-repudiation, etc. Moreover, dealing with the user's PII data needs to provide a high level of privacy.

R6 For an agile migration to a new ecosystem, its compatibility with legacy systems, hardware, software, and architectures is the first requirement. Moreover, in standardized ecosystems, such as cellular networks, being compliant with the standards is a vital issue.

In a high-level abstract, in the conventional stand-alone MNO architecture, not only there is no collaboration solution among entities, but also both user and control plane procedures are handled by one MNO (i.e., using centralized authority, with centralized IT procedure, and in a centralized architecture), which can not address the aforementioned requirements. Moreover, from *business perspective*, the existing architecture suffers from 1) high operation and installation costs for MNOs and providers to build stand-alone cellular networks, which can negatively affect their revenue-to-cost proportion, 2) negative environmental effects and energy consumption due to non-mutualization of reusable resources, 3) lack of collaboration between content/service providers and MNOs that impose the cost of innovation and network expansion to the MNOs, and 4) contract management complexity. Apart from the business-related challenges, the centralized architecture of the cellular network can bring availability, security, privacy and scalability issues.

In this regard, we profit from the unique features of Blockchain technology to propose a novel distributed architecture for core networks and cellular network-related services beyond 5G and 6G to provide a multi-actor cellular network system. The proposed architecture combines centralized/decentralized and distributed solutions to introduce a semi-distributed cellular network architecture in which the authority of the procedures is distributed among entities and the entities can cooperate more automated and trusted. Apart from business-related collaboration, the IT procedure can also be executed by several parties in a trustful environment. This solution can potentially bring many contributions to the definition of future 6G cellular networks. The proposed architecture focuses on collaboration possibilities among MNOs, connectivity providers, and service providers by migrating different independent procedures on MNOs, such as user subscription, identity management, user registration in

the cellular networks, key-agreement procedures, mobility management, and billing processes, to a distributed system empowered by Blockchain consensus and its intrinsic security features.

As defined in [237], Blockchain technology can provide more secure and reliable communication in current cellular networks. Moreover, it allows various entities to securely share and access data or resources. So, the main beneficial features of Blockchain are its trustworthiness and reliability in a distrusted environment which is achieved by the transparency of the transaction validation, the consensus among nodes, and the openness of the smart contracts for validation by all parties, the distributed nature of the Blockchain-based systems, and the immutability of the transactions, data, and policies after being written in the system. The proposed system can bring several benefits for different parties. From **an MNO perspective**, first, they would be able to provide more collaborations with other entities to decrease the costs by mutualization, decrease the environmental effects of connectivity providing infrastructure, decrease the complexity of the IT procedure by migration of user access/mobility management, and billing to Blockchain. Moreover, new business opportunities would be other incentives for MNOs to bring new solutions to serve more users, develop innovative ideas in collaboration with other entities, etc. **Service providers, vendors, and connectivity providers** can either outsource different functionality (e.g., authentication and access control, identity management, billing) to a distributed system or provide these services (plus connectivity) in the cellular network market and profit from its revenue. Furthermore, decreasing the IT procedure and management costs in the businesses would lead to providing novel and innovative services/products for the user (i.e., the enterprises can invest their assets in novel products and services). **The users** would experience broader coverage in their geographical area. Furthermore, the market makes it possible for providers to introduce more competitive services with competitive prices. So, the users can freely choose the service which is more satisfactory for them.

Apart from the incentives provided by the proposed architecture, we conclude that this method can address the majority of derived requirements. Indeed, there are several limitations that we will discuss on them in the next sections. In summary, the proposed distributed core network architecture can, fully or partially, address the requirements as follows:

R1 handling different functionalities such as core network functions, contract management in the scale of businesses, managing the collaboration between entities,

billing procedure, etc., and collaboration/contract management using smart contracts can increase automation in the system while decreasing the manual process time.

R2 Blockchain and smart contracts provide non-repudiation, execute consensus procedures, and provide transparency in the validation of the requests. Using these features, this technology brings trust between entities in distrustful environments.

R3 Migrating functionalities of the core network that are currently handled by a central authority to the Blockchain, brings decentralization of authority and operation to the ecosystem. So, the system availability is expected to be higher as well.

R4 The proposed method provides the opportunity for collaboration between an unlimited number of collaborators. Regarding the increasing number of users, the Blockchain scalability issue comes up. In the lab scale, the scalability of the system regarding the number of users is not visible, but for real-world implementation, some discussions are provided in the next section. this problem s not an issue

R5 In different entities of the proposed system such as authentication procedure, access control, identity management, and storing the user's identity in the system.

R6 The complete scenario of the proposed method which migrates the network functions to Blockchain-based systems is a clean-slate proposal for the beyond 5G which is not compliant with the 3GPP standard. However, there are other standard-compliant scenarios to benefit from Blockchain technology in the current cellular networks.

To introduce a multi-actor collaborative mobile connectivity system, this work brings several contributions such as 1) providing a comprehensive study of the existing challenges in current cellular networks and the Blockchain's role in addressing the challenges, 2) proposing a novel Blockchain-based architecture to introduce a multi-actor collaboration system **among different actors of cellular network ecosystem**, and 3) proposing a new user profile management and mobile number and profile porting solution to introduce a multi-actor collaboration system **among MNOs and authorization bodies**.

To evaluate the proposed architecture, we assessed the scalability and storage requirements of different network functions in several scenarios. The evaluation indicates that the proposed method is scalable in the scale of our experiments. Moreover, the security analysis is also provided to explain how the proposed method and function are able to resist different network or Blockchain-based attacks.

Indeed, the **real-world implementation** of the proposed architecture needs the answers to the following questions rather than technological issues: First, it should be defined 1) Who are the owners of the Blockchain? In other words, who is responsible for governing the Blockchain, its rules, incentives, etc.? 2) Which entities participate in securing the Blockchain, including storing the ledger, validating the transactions, and participating in the consensus procedure? 3) How the trust among MNOs, external entities, and users would be addressed?

The main actors of the proposed system are users, MNO, µOs, connectivity providers, and service providers. In the proposed method the underlying Blockchain can be implemented as a consortium among actors except for the users. Since, the users' processing powers, storage, and resources are limited, their participation in the consensus, not only can be inefficient but also provide new attack vectors and security breaches. The Blockchain's configuration and requirements can be shaped based on the **local regulatory rules(in the territory of the country)**, service needs, participants' requirements, and the trust level between entities. So, we propose to have the following setting for real-world implementation. **The Blockchain itself should be a consortium** among MNO and external entities, but some smart contracts such as SC_{ExE}, SC_{AB}, SC_{CNE}, and SC_{Sub} that has the role of regulation and connection management, should be deployed and owned by the MNO.

For the second question, as the Blockchain is a consortium among MNO and external entities, they are the entities who participate in consensus and keep the latest version of the ledger. Note that, the storage and scalability problem of this approach is discussed in the following subsections.

Regarding trust in the system, all external entities are registered in the system and identified by the MNO. So, we can claim that a minimal level of trust exists in the ecosystem. Moreover, regarding the trust between the proposed architecture and the user, all subscription, registration, and handover procedures are handled by smart contracts as a distributed trusted party.

The following chapter aims to provide several discussions on the limitations of the method regarding the intrinsic problems of the Blockchain and the solutions to ad-

dress these limitations. Moreover, we will provide some proposals for future directions.

9.2 Limitations and Discussions

9.2.1 System scalability

As explained in [238], Blockchain has a trilemma of features named Scalability, Security, and Decentralization, which can not be entirely delivered altogether. Due to the significance of all three parameters in the cellular network use case and knowing that security and decentralization are provided in the proposed architecture, scalability is the third parameter that needs to be addressed.

The main goal of increasing the scalability is to provide higher throughput while growing the number of concurrent transactions. Generally, there are two main dimensions of Blockchain scalability, namely *horizontal* and *vertical* [239]. Horizontal scalability refers to the capability of Blockchain to increase the throughput (or at least not to degrade it) by adding new nodes, while vertical scalability aims to enhance the capabilities of participating nodes to achieve higher throughput [240]. The first dimension is highly dependent on the consensus model, so we avoid discussing that. To address the vertical scalability of the Blockchain, we propose two approaches:

- *Designing a tailor-made Blockchain for cellular network*: It would be interesting and encouraging for the Blockchain community researchers to design a Blockchain with specific consensus models, block sizes, transaction fees, block times, incentives, and other specifications to make it possible validating the higher number of transactions in a given time. This solution can be beneficial to address the needs of all parties in the cellular network ecosystem.

- *Chain sharding:* Although the aforementioned solution is beneficial in several aspects, it needs lots of research and proof before application. For instance, providing high reliability and security of a novel consensus model is not straightforward and needs very precise analysis. Another novel solution in this regard is *sharding the chain* horizontally to distribute the transaction loads among shards [241]. After the validation of transactions in the shards and generating the blocks, a smart contract would be utilized to merge the shard blocks to the main chain [56]. This solution increases the throughput and decreases the

storage usage even by using the pre-examined and approved consensus models such as PoS, PBFT, etc. Several sharding solutions are recently proposed that state the feasibility of this method. For instance, RapidChain [242] increased the throughput to $7380TpS$ in comparison with $15-20$ in Ethereum with 4000 participating nodes and 250 shards. Another solution is proposed by Dang et al. [243] that reaches $3000TpS$. Finally, Meepo [244] is a sharded consortium Blockchain that reaches $120,000TpS$. These statistics state that sharding is a promising solution to increase the scalability [245] to an acceptable level for cellular network needs.

– Another improvement in latency and scalability can be done by *separating low latency and high throughput networks* among Blockchain entities to support the **high connectivity** and **decrease the bandwidth overhead**. This relay network can be used to update the ledger with the minimum delay and consequently, increase the system throughput. For instance, FIBRE is a real-world example of a block relay network.

9.2.2 System Storage requirement

Although it provides many unprecedented opportunities, Blockchain technology needs a huge amount of storage in its full nodes to keep the updated ledger and provide security. Currently, the proposed method generates one to three transactions that are all stored in full-node storage. So, to address the storage complexity defect of Blockchain, the following solutions can be applied:

– *Chain sharding:* Rather than its benefits for scalability (See the section 9.2.1), chain sharding can be highly beneficial in case of storage requirements. In this technique, each shard functions independently of the other shards (i.e., it has its own block validator, number of input transactions, and storage requirements). So, in each shard, the participating nodes are required to keep the transactions of their own shard, which can decrease the storage complexity. For instance, Rapidchain [242] can decrease the storage usage by 16 times in comparison with non-sharded chains.

– Optimize the number of transactions and store their content in cloud-based external storage (and store their access URL in Blockchain) in the future steps. So, the transaction number and their size in the Blockchain would be reduced, which results in decreasing the overall required repository.

However, in the proposed system, using IPFS to store the non-transaction data outside of the Blockchain environment, can decrease the storage requirement.

9.2.3 Compatibility with legacy systems

As shown in Fig. 7.1, among three possibilities of system implementation, Fig. 7.1 (b) is the fully distributed solution which indeed, is not fully compatible with the existing architecture. However, several software changes can make its implementation feasible. For instance, the hardware and software of the RAN part of the system remain intact (including the Central Unit (CU) and Distributed Unit (DU) entities in the Cloud-RAN architecture [246]). So, there is no need to change in RAN part. The only requirement is to design a gateway to send the requests to the Blockchain (similar to Fig. 7.1 (a)), instead of the existing core network. In this case, the new architecture can function along with the existing cellular networks (to support, e.g., collaboration among different technologies and providing real-time emergency connections in which using the Blockchain is not suitable). The most compatible implementation of the method is Fig. 7.1 (c) in which many functionalities of the core network and RAN remain intact.

Moreover, it is important to mention that the new model's *standardization* needs to be considered for next-generation networks by the standardization bodies. However, in this work, the building blocks of the open cellular network are provided in a high-level abstract and based on market/enterprise needs, update capacity, and use cases it is possible to use different implementation scenarios for full or partial deployment of the architecture.

9.3 Future directions

Following some future directions are listed to improve the proposed method, add new features, and provide a more comprehensive solution:

- ***Network function migration:*** This work proposes a distributed architecture for the core network and its network functions. Progressively migrating different functionalities of existing open-source private cellular networks (e.g., OpenAirInterface, Magma, etc.) is proposed for the first step in the future. Indeed, in order to be compliant with the standard, full implementation of the method is not possible for time being.

– ***Designing a tailor-made Blockchain for multi-actor cellular network system*** can be highly beneficial to provide the requirements of this ecosystem such as low latency in consensus procedure, scalability, and storage efficiency. Indeed to address these requirements, the previously mentioned solutions such as chain sharding, off-chain storage, etc. would be highly useful.

9.4 Last words

I wish to end the chapter by providing a short non-technical description of my research design about how we reached this idea from the beginning of the work.

The starting point of this work was to find a new solution for authentication and access control in the cellular network which can address the existing problems and provide a new business model in this ecosystem. After investigating this subject, we found some challenges, in the existing ecosystem which could be addressed by Blockchain technology. Indeed, at that time, and for the application layers' use cases, the requirements of the system were different (e.g., latency was not a concern). So, we proposed the idea of *"service provisioning in cellular networks using Blockchain technology"*.

While investigating more Blockchain-based access control methods, their opportunities, the requirements of the next-generation networks, the new types of private cellular networks, etc. we reached the point that collaboration is critical in next-generation networks and it is vital to provide proper services based on the requirements of the use-cases. So, we noticed that the Blockchain's opportunities can be extended to the other connection functionalities of the cellular networks So, we proposed the idea of *"designing the over architecture for the mobile network operators beyond 5G"*.

While working on this subject, and trying to provide a solution for subscriber management in the cellular network, we found another use case regarding the management of the user's profile in MNOs (i.e., the idea of mobile number portability) that can be improved to not only port the user's phone number but also since the user's identity can be securely shared among MNOs, the users profile is also can be ported. Thus, we ended up proposing the idea of *"mobile number and profile porting beyond 5G"*.

References

[1] "Mobile Vs. Desktop Internet Usage (Latest 2021 Data)." [Online]. Available: https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics

[2] "What percentage of internet traffic is mobile?" [Online]. Available: https://www.oberlo.com/statistics/mobile-internet-traffic

[3] "Mobile subscriptions forecast – Mobility Report," Nov. 2021, last Modified: 2021-11-30T06:30:56+00:00. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-subscriptions-outlook

[4] "Global mobile trends 2021- navigating covid-19 and beyond," Global System for Mobile Communications Association, Tech. Rep., 12 2020. [Online]. Available: https://data.gsmaintelligence.com/api-web/v2/research-file-download?file=141220-Global-Mobile-Trends.pdf&id=58621970

[5] "Cisco annual internet report(2018–2023)," Cisco, Tech. Rep., 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf

[6] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.

[7] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, "6g wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, 2019.

[8] M. Tahir, M. H. Habaebi, M. Dabbagh, A. Mughees, A. Ahad, and K. I. Ahmed, "A review on application of blockchain in 5g and beyond networks: Taxonomy, field-trials, challenges and opportunities," *IEEE Access*, vol. 8, pp. 115 876–115 904, 2020.

[9] Z. Luo, S. Fu, M. Theis, S. Hasan, S. Ratnasamy, and S. Shenker, "Democratizing cellular access with cellbricks," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 626–640.

[10] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5g and beyond networks: A state of the art survey," *Journal of Network and Computer Applications*, vol. 166, p. 102693, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520301673

[11] "Netcracker | From Telco to NetCo and ServCo." [Online]. Available: https://www.netcracker.com/blog/view-all/from-telco-to-netco-and-servco.html

[12] "Technical specification group services and system aspects; network architecture (release 13)," European Telecommunications Standards Institute (ETSI), Tech. Rep., 03 2016.

[13] B. Saha, M. Martínez-García, S. N. Bhattacharya, and R. Joshi, "Overcoming choice inertia through social interaction—an agent-based study of mobile subscription decision," *Games*, vol. 13, no. 3, p. 47, 2022.

[14] "Compettition policy in the digital age: A practical handbook," Global System for Mobile Communications Association, Tech. Rep., 08 2015. [Online]. Available: https://www.gsma.com/publicpolicy/wp-content/uploads/2016/04/Competition-Policy-Handbook.pdf

[15] "Together we are unstoppable," T-Mobile, Tech. Rep., 09 2021. [Online]. Available: https://s24.q4cdn.com/400059132/files/doc_financials/2021/q3/NG_TMUS-09_30_2021-ER-Tables-vFinal.pdf

[16] "Financial and operational trends," AT&T, Tech. Rep., 10 2021. [Online]. Available: https://investors.att.com/~/media/Files/A/ATT-IR-V2/financial-reports/quarterly-earnings/2021/q321/ATT_3Q21_Earnings_Deck/T_3Q21_Trending_Schedule.pdf

[17] "Financial and operating information," Verizon, Tech. Rep., 10 2021. [Online]. Available: https://www.verizon.com/about/investors/quarterly-reports/3q-2021-earnings-conference-call-webcast

[18] M. Motta, *Competition policy: theory and practice*. Cambridge University Press, 2004.

[19] G. Koltay and S. Lorincz, "Industry concentration and competition policy," p. 5, 2021. [Online]. Available: https://ec.europa.eu/competition-policy/system/files/2021-11/Competition%20Policy%20Brief%20-%20Industry%20concentration%20and%20competition%20policy%202-2021_1.pdf

[20] A. H. Khan, N. U. Hassan, C. Yuen, J. Zhao, D. Niyato, Y. Zhang, and H. V. Poor, "Blockchain and 6g: The future of secure and ubiquitous communication," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 194–201, 2021.

[21] T. Hewa, G. Gür, A. Kalla, M. Ylianttila, A. Bracken, and M. Liyanage, "The role of blockchain in 6g: Challenges, opportunities and research directions," *2020 2nd 6G Wireless Summit (6G SUMMIT)*, pp. 1–5, 2020.

[22] Q. Ni, Z. Linfeng, X. Zhu, and I. Ali, "A novel design method of high throughput blockchain for 6g networks: Performance analysis and optimization model," *IEEE Internet of Things Journal*, 2022.

[23] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, "Internet of things and its applications: A comprehensive survey," *Symmetry*, vol. 12, no. 10, p. 1674, 2020.

[24] M. U. A. Siddiqui, F. Qamar, M. Tayyab, M. N. Hindia, Q. N. Nguyen, and R. Hassan, "Mobility management issues and solutions in 5g-and-beyond networks: A comprehensive review," *Electronics*, vol. 11, no. 9, p. 1366, 2022.

[25] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5g network architecture," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, 2014.

[26] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, "Learning radio resource management in rans: Framework, opportunities, and challenges," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 138–145, 2018.

[27] K. Yue, Y. Zhang, Y. Chen, Y. Li, L. Zhao, C. Rong, and L. Chen, "A survey of decentralizing applications via blockchain: The 5g and beyond perspective," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2191–2217, 2021.

[28] T. Maksymyuk, J. Gazda, L. Han, and M. Jo, "Blockchain-based intelligent network management for 5g and beyond," in *2019 3rd International Conference on Advanced Information and Communications Technologies (AICT)*. IEEE, 2019, pp. 36–39.

[29] C. Benzaid and T. Taleb, "Ai-driven zero touch network and service management in 5g and beyond: Challenges and research directions," *IEEE Network*, vol. 34, no. 2, pp. 186–194, 2020.

[30] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2019.

[31] H. Khan and K. M. Martin, "A survey of subscription privacy on the 5g radio interface-the past, present and future," *Journal of Information Security and Applications*, vol. 53, p. 102537, 2020.

[32] "SDN and NFV: What are the features and benefits?" [Online]. Available: https://techbeacon.com/enterprise-it/sdn-nfv-do-enterprise-benefits-add

[33] J. Costa-Requena, J. L. Santos, V. F. Guasch, K. Ahokas, G. Premsankar, S. Luukkainen, O. L. Pérez, M. U. Itzazelaia, I. Ahmad, M. Liyanage, *et al.*, "Sdn and nfv integration in generalized mobile network architecture," in *2015 European conference on networks and communications (EuCNC)*. IEEE, 2015, pp. 154–158.

[34] S. Wijethilaka and M. Liyanage, "Survey on network slicing for internet of things realization in 5g networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 957–994, 2021.

[35] S. Yrjölä *et al.*, "Decentralized 6g business models," *Proceedings of the 6G Wirel. Summit, Levi, Finland*, pp. 5–7, 2019.

[36] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5g: Challenges and opportunities," *IEEE Internet Computing*, vol. 21, no. 5, pp. 20–27, 2017.

[37] A. Nag, A. Kalla, and M. Liyanage, "Blockchain-over-optical networks: A trusted virtual network function (vnf) management proposition for 5g optical networks," in *2019 Asia Communications and Photonics Conference (ACP)*. IEEE, 2019, pp. 1–3.

[38] C. Benzaïd, T. Taleb, and M. Z. Farooqi, "Trust in 5g and beyond networks," *IEEE Network*, vol. 35, no. 3, pp. 212–222, 2021.

[39] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[40] I. Bashir, *Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained*. Packt Publishing Ltd, 2018.

[41] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[42] Y. Zhao, J. Zhao, W. Zhai, S. Sun, D. Niyato, and K.-Y. Lam, "A survey of 6g wireless communications: Emerging technologies," in *Future of Information and Communication Conference*. Springer, 2021, pp. 150–170.

[43] H. H. H. Mahmoud, A. A. Amer, and T. Ismail, "6g: A comprehensive survey on technologies, applications, challenges, and research problems," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 4, p. e4233, 2021.

[44] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 858–880, 2018.

[45] "5G and blockchain: The building blocks of a shared economy," Oct. 2019, last Modified: 2020-04-23T05:27:17+00:00. [Online]. Available: https://www.ericsson.com/en/blog/2019/10/5g-blockchain-shared-economy

[46] N. Bozic, G. Pujolle, and S. Secci, "Securing virtual machine orchestration with blockchains," in *2017 1st Cyber Security in Networking Conference (CSNet)*. IEEE, 2017, pp. 1–8.

[47] A. Shahraki, M. Abbasi, M. Piran, M. Chen, S. Cui, *et al.*, "A comprehensive survey on 6g networks: Applications, core services, enabling technologies, and future challenges," *arXiv preprint arXiv:2101.12475*, 2021.

[48] Y. Lu and X. Zheng, "6g: A survey on technologies, scenarios, challenges, and the related issues," *Journal of Industrial Information Integration*, p. 100158, 2020.

[49] P. Porambage, G. Gür, D. P. M. Osorio, M. Liyanage, A. Gurtov, and M. Ylianttila, "The roadmap to 6g security and privacy," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1094–1122, 2021.

[50] M. Belotti, N. Božić, G. Pujolle, and S. Secci, "A vademecum on blockchain technologies: When, which, and how," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3796–3838, 2019, publisher: IEEE.

[51] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[52] N. Kannengießer, S. Lins, T. Dehling, and A. Sunyaev, "Trade-offs between distributed ledger technology characteristics," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–37, 2020, publisher: ACM New York, NY, USA.

[53] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," p. 9.

[54] "Secure Property Titles with Owner Authority | Satoshi Nakamoto Institute." [Online]. Available: https://nakamotoinstitute.org/secure-property-titles/

[55] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020, publisher: Elsevier.

[56] "Ethereum Whitepaper." [Online]. Available: https://ethereum.org

[57] "Solidity." [Online]. Available: https://docs.soliditylang.org/en/v0.8.17/

[58] I. Sergey, A. Kumar, and A. Hobor, "Scilla: a Smart Contract Intermediate-Level LAnguage," *arXiv:1801.00687 [cs]*, Jan. 2018, arXiv: 1801.00687. [Online]. Available: http://arxiv.org/abs/1801.00687

[59] "DagCoin: a cryptocurrency without blocks," Sept. 2015. [Online]. Available: https://bitslog.com/2015/09/11/dagcoin/

[60] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: a fast and scalable cryptocurrency protocol." *IACR Cryptol. ePrint Arch.*, vol. 2016, no. 1159, 2016.

[61] K. Gai, Z. Hu, L. Zhu, R. Wang, and Z. Zhang, "Blockchain meets dag: A blockdag consensus mechanism," in *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 2020, pp. 110–125.

[62] Q. Zhu, S. W. Loke, R. Trujillo-Rasua, F. Jiang, and Y. Xiang, "Applications of distributed ledger technologies to the internet of things: A survey," *ACM computing surveys (CSUR)*, vol. 52, no. 6, pp. 1–34, 2019, publisher: ACM New York, NY, USA.

[63] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, A. S. M. Kayes, M. Alazab, and P. Watters, "A Comparative Analysis of Distributed Ledger Technology Platforms," *IEEE Access*, vol. 7, pp. 167 930–167 943, 2019, conference Name: IEEE Access.

[64] A. López Vivar, A. T. Castedo, A. L. Sandoval Orozco, and L. J. García Villalba, "An analysis of smart contracts security threats alongside existing solutions," *Entropy*, vol. 22, no. 2, p. 203, 2020.

[65] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*. IEEE, 2017, pp. 557–564.

[66] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2018, publisher: IEEE.

[67] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018, publisher: Inderscience Publishers (IEL).

[68] A. H. Mohsin, A. A. Zaidan, B. B. Zaidan, O. S. Albahri, A. S. Albahri, M. A. Alsalem, and K. I. Mohammed, "Blockchain authentication of network applications: Taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions," *Computer Standards & Interfaces*, vol. 64, pp. 41–60, 2019, publisher: Elsevier.

[69] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 858–880, 2018, publisher: IEEE.

[70] S. Y. Lim, P. T. Fotsing, A. Almasri, O. Musa, M. L. M. Kiah, T. F. Ang, and R. Ismail, "Blockchain technology the identity management and authentication service disruptor: a survey," *Int. J. Adv. Sci. Eng. Inf. Technol*, vol. 8, no. 4-2, pp. 1735–1745, 2018.

[71] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020, publisher: IEEE.

[72] M. Li, C. Lal, M. Conti, and D. Hu, "Lechain: A blockchain-based lawful evidence management scheme for digital forensics," *Future Generation Computer Systems*, vol. 115, pp. 406–420, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X1933167X

[73] G. Kumar, R. Saha, C. Lal, and M. Conti, "Internet-of-forensic (iof): A blockchain based digital forensics framework for iot applications," *Future Generation Computer Systems*, vol. 120, pp. 13–25, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21000686

[74] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," in *Concurrency: the Works of Leslie Lamport*. New York, NY, USA: Association for Computing Machinery, Oct. 2019, pp. 203–226. [Online]. Available: https://doi.org/10.1145/3335772.3335936

[75] V. Gramoli, "From blockchain consensus back to Byzantine consensus," *Future Generation Computer Systems*, vol. 107, pp. 760–769, 2020, publisher: Elsevier.

[76] P. Siano, G. De Marco, A. Rolán, and V. Loia, "A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3454–3466, 2019, publisher: IEEE.

[77] I. Homoliak, S. Venugopalan, D. Reijsbergen, Q. Hum, R. Schumi, and P. Szalachowski, "The Security Reference Architecture for Blockchains: Toward a Standardized Model for Studying Vulnerabilities, Threats, and Defenses," *IEEE Communications Surveys Tutorials*, vol. 23, no. 1, pp. 341–390, 2021, conference Name: IEEE Communications Surveys Tutorials.

[78] L. Lamport *et al.*, "Paxos made simple," *ACM Sigact News*, vol. 32, no. 4, pp. 18–25, 2001.

[79] L. Ismail and H. Materwala, "A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions," *Symmetry*, vol. 11, no. 10, p. 1198, 2019, publisher: Multidisciplinary Digital Publishing Institute.

[80] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Annual international cryptology conference*. Springer, 1992, pp. 139–147.

[81] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in *Secure information networks*. Springer, 1999, pp. 258–272.

[82] S. Popov, "The tangle," 2016. [Online]. Available: https://iotatoken.com/IOTA\char'_Whitepaper.pdf

[83] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," p. 6.

[84] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019, conference Name: IEEE Access.

[85] S. Wan, M. Li, G. Liu, and C. Wang, "Recent advances in consensus protocols for blockchain: a survey," *Wireless networks*, vol. 26, no. 8, pp. 5579–5593, 2020, publisher: Springer.

[86] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain," p. 11.

[87] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10 127–10 149, 2019, publisher: IEEE.

[88] "Clique PoA protocol & Rinkeby PoA testnet · Issue #225 · ethereum/EIPs." [Online]. Available: https://github.com/ethereum/EIPs/issues/225

[89] "poanetwork/website." [Online]. Available: https://github.com/poanetwork/website/blob/master/for-users/whitepaper/poadao-v1/proof-of-authority.md

[90] "The Second Coming of Blockchain." [Online]. Available: https://www.intel.com/content/www/us/en/develop/blogs/the-second-coming-of-blockchain.html

[91] "Intel® Software Guard Extensions." [Online]. Available: https://www.intel.com/content/www/us/en/develop/topics/software-guard-extensions.html

[92] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted Execution Environment: What It is, and What It is Not," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, Aug. 2015, pp. 57–64.

[93] 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.

[94] T. NEM, "Nem technical reference," *URL https://nemplatform.com/wp-content/uploads/2020/05/NEM_techRef.pdf*, 2018.

[95] "Delegated Proof of Stake (DPOS) — BitShares Documentation documentation." [Online]. Available: https://how.bitshares.works/en/master/technology/dpos.html

[96] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, 1999, pp. 173–186, issue: 1999.

[97] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, 2014, pp. 305–319.

[98] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*. Springer, 2017, pp. 357–388.

[99] "What is a Mobile Network Operator (MNO)? - Definition from Techopedia." [Online]. Available: http://www.techopedia.com/definition/27804/mobile-network-operator-mno

[100] S. Rommer, P. Hedman, M. Olsson, L. Frid, S. Sultana, and C. Mulligan, *5G Core Networks: Powering Digitalization*. Academic Press, 2019.

[101] "Introducing 3GPP." [Online]. Available: https://www.3gpp.org/about-us/introducing-3gpp

[102] M. Ivezic, "Introduction to 5G Core Service-Based Architecture (SBA) Components," Aug. 2020. [Online]. Available: https://5g.security/5g-technology/5g-core-sba-components-architecture/

[103] "5g; system architecture for the 5g system (5gs)-(3gpp ts 23.501 version 15.12.0 release 15)," European Telecommunications Standards Institute (ETSI), Tech. Rep., 01 2021.

[104] "5g; 5g system; unified data repository services; stage 3 (3gpp ts 29.504 version 15.3.0 release 15)," 3GPP, Tech. Rep., 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/129500_129599/129504/15.03.00_60/ts_129504v150300p.pdf

[105] "5g; 5g system; usage of the unified data repository services for subscription data; stage 3 (3gpp ts 29.505 version 16.3.0 release 16)," 3GPP, Tech. Rep., 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/129500_129599/129505/16.03.00_60/ts_129505v160300p.pdf

[106] "5g; security architecture and procedures for 5g system (3gpp ts 33.501 version 15.6.0 release 15)," 3GPP, Tech. Rep., 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/133500_133599/133501/15.06.00_60/ts_133501v150600p.pdf

[107] "Security in 5g specifications, controls in 3gpp security specifications (5g sa)," ENISA, Tech. Rep., 2021. [Online]. Available: https://www.enisa.europa.eu/publications/security-in-5g-specifications/at_download/fullReport

[108] Q. Tang, O. Ermis, C. D. Nguyen, A. De Oliveira, and A. Hirtzig, "A systematic analysis of 5g networks with a focus on 5g core security," *IEEE Access*, vol. 10, pp. 18 298–18 319, 2022.

[109] "5g; system architecture for the 5g system (5gs)(3gpp ts 23.501 version 16.6.0 release 16)," 3GPP, Tech. Rep., 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf

[110] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations," National Institute of Standards and Technology, Tech. Rep. NIST SP 800-162, Jan. 2014. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-162.pdf

[111] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," in *2012 IEEE Symposium on Security and Privacy*, May 2012, pp. 553–567, iSSN: 2375-1207.

[112] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004, publisher: IEEE.

[113] J. M. Stewart, E. Tittel, and M. Chapple, *CISSP: Certified information systems security professional study guide*. John Wiley & Sons, 2008.

[114] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, no. 1, p. 1, 2018, publisher: Multidisciplinary Digital Publishing Institute.

[115] D. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-based access control*. Artech House, 2003.

[116] O. Standard, "extensible access control markup language (xacml) version 3.0," 2013.

[117] B. W. Lampson, "Protection," *ACM SIGOPS Operating Systems Review*, vol. 8, no. 1, pp. 18–24, 1974.

[118] E. Bertin, D. Hussein, C. Sengul, and V. Frey, "Access control in the internet of things: a survey of existing approaches and open research questions," *Annals of Telecommunications*, vol. 74, no. 7, pp. 375–388, 2019.

[119] H. M. Levy, *Capability-based computer systems*. Digital Press, 2014. [Online]. Available: https://homes.cs.washington.edu/~levy/capabook/

[120] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A Survey on Access Control in the Age of Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, June 2020, conference Name: IEEE Internet of Things Journal.

[121] P. Samarati and S. C. de Vimercati, "Access control: Policies, models, and mechanisms," in *International School on Foundations of Security Analysis and Design*. Springer, 2000, pp. 137–196.

[122] D. E. Denning, "A lattice model of secure information flow," *Communications of the ACM*, vol. 19, no. 5, pp. 236–243, 1976.

[123] R. S. Sandhu, "Role-based access control," in *Advances in computers*. Elsevier, 1998, vol. 46, pp. 237–286.

[124] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[125] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 89–98.

[126] B. Sumitra, C. R. Pethuru, and M. Misbahuddin, "A Survey of Cloud Authentication Attacks and Solution Approaches," in *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), An ISO 3297:2007, ISSN (online): 23209801, ISSN (Print): 2320-9798, Volume 2, Issue 10*, 2014.

[127] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *Journal of Network and Computer Applications*, vol. 116, pp. 42–52, 2018, publisher: Elsevier.

[128] L. Zhang, H. Li, L. Sun, Z. Shi, and Y. He, "Poster: towards fully distributed user authentication with blockchain," in *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 2017, pp. 202–203.

[129] G. Deep, R. Mohana, A. Nayyar, P. Sanjeevikumar, and E. Hossain, "Authentication protocol for cloud databases using blockchain mechanism," *Sensors*, vol. 19, no. 20, p. 4444, 2019, publisher: Multidisciplinary Digital Publishing Institute.

[130] "NIST Special Publication 800-63B." [Online]. Available: /sp800-63b.html

[131] J. Kohl, C. Neuman, *et al.*, "The kerberos network authentication service (v5)," RFC 1510, september, Tech. Rep., 1993.

[132] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013, publisher: IEEE.

[133] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016, publisher: IEEE.

[134] A. Jesudoss and N. Subramaniam, "A survey on authentication attacks and countermeasures in a distributed environment," *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 5, no. 2, pp. 71–77, 2014.

[135] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, May 2018.

[136] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.

[137] R. John, J. P. Cherian, and J. J. Kizhakkethottam, "A survey of techniques to prevent sybil attacks," in *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, Feb. 2015, pp. 1–6.

[138] L. Tamilselvan and D. V. Sankaranarayanan, "Prevention of impersonation attack in wireless mobile ad hoc networks," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 7, no. 3, pp. 118–123, 2007, publisher: Citeseer.

[139] N. Abdullah, A. Hakansson, and E. Moradian, "Blockchain based approach to enhance big data authentication in distributed environment," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2017, pp. 887–892.

[140] "ION – Booting up the network," June 2020, section: Identity Standards Blog. [Online]. Available: https://techcommunity.microsoft.com/t5/identity-standards-blog/ion-booting-up-the-network/ba-p/1441552

[141] "Identity protocol v1 - Bitcoin Wiki." [Online]. Available: https://en.bitcoin.it/wiki/Identity\char'_protocol\char'_v1

[142] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," *Computer Science Review*, vol. 30, pp. 80–86, 2018.

[143] K. Baqer, D. Y. Huang, D. McCoy, and N. Weaver, "Stressing out: Bitcoin "stress testing"," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 3–18.

[144] K. Bicakci, D. Unal, N. Ascioglu, and O. Adalier, "Mobile authentication secure against man-in-the-middle attacks," *Procedia Computer Science*, vol. 34, pp. 323–329, 2014.

[145] Z. Cui, X. Fei, S. Zhang, X. Cai, Y. Cao, W. Zhang, and J. Chen, "A hybrid blockchain-based identity authentication scheme for multi-wsn," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 241–251, 2020.

[146] L. Xiong, F. Li, S. Zeng, T. Peng, and Z. Liu, "A blockchain-based privacy-awareness authentication scheme with efficient revocation for multi-server architectures," *IEEE Access*, vol. 7, pp. 125 840–125 853, 2019.

[147] A. Z. Ourad, B. Belgacem, and K. Salah, "Using blockchain for iot access control and authentication management," in *International Conference on Internet of Things*. Springer, 2018, pp. 150–164.

[148] A. Yakubov, W. Shbair, N. Khan, C. Medinger, J. Hilger, *et al.*, "Blockpgp: A blockchain-based framework for pgp key servers," *International Journal of Networking and Computing*, vol. 10, no. 1, pp. 1–24, 2020.

[149] B. Alotaibi, "Utilizing blockchain to overcome cyber security concerns in the internet of things: A review," *IEEE Sensors Journal*, vol. 19, no. 23, pp. 10 953–10 971, 2019, publisher: IEEE.

[150] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A Survey on Security and Privacy Issues of Bitcoin," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018, conference Name: IEEE Communications Surveys Tutorials.

[151] K. Alachkar and D. Gaastra, *Blockchain-based Sybil Attack Mitigation: A Case Study of the I2P Network*. Semanticscholar Seattle, Washington, 2018.

[152] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2794–2830, 2019, publisher: IEEE.

[153] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[154] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 2663–2668.

[155] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020, publisher: IEEE.

[156] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*, 1970, pp. 281–285.

[157] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.

[158] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE communications surveys & tutorials*, vol. 16, no. 2, pp. 1024–1049, 2013.

[159] M. Conti, M. Hassan, and C. Lal, "Blockauth: Blockchain based distributed producer authentication in icn," *Computer Networks*, vol. 164, p. 106888, 2019.

[160] Z. Luo, S. Fu, M. Theis, S. Hasan, S. Ratnasamy, and S. Shenker, "Democratizing Cellular Access with CellBricks (Extended Version)," p. 17.

[161] "magma/magma," Sept. 2021, original-date: 2019-02-15T04:46:24Z. [Online]. Available: https://github.com/magma/magma

[162] S. Hasan, A. Padmanabhan, B. Davie, J. Rexford, U. Kozat, H. Gatewood, S. Sanadhya, N. Yurchenko, T. Al-Khasib, O. Batalla, *et al.*, "Building flexible, low-cost wireless access networks with magma," *arXiv preprint arXiv:2209.10001*, 2022.

[163] "lorawan-on-helium | Helium Documentation." [Online]. Available: https://docs.helium.com/lorawan-on-helium

[164] A. Haleem, "Episode Two: The Path To 5G," Apr. 2021. [Online]. Available: https://blog.helium.com/episode-two-the-path-to-5g-3f704a58661

[165] X. Ling, Y. Le, J. Wang, and Z. Ding, "Hash access: Trustworthy grant-free iot access enabled by blockchain radio access networks," *IEEE Network*, vol. 34, no. 1, pp. 54–61, 2020.

[166] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, and K.-K. R. Choo, "Blockchain-enabled authentication handover with efficient privacy protection in sdn-based 5g networks," *IEEE Transactions on Network Science and Engineering*, 2019.

[167] Y. Zhang, R. Deng, E. Bertino, and D. Zheng, "Robust and universal seamless handover authentication in 5g hetnets," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[168] M. T. Hammi, P. Bellot, and A. Serhrouchni, "Bctrust: A decentralized authentication blockchain-based mechanism," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[169] H. Lee and M. Ma, "Blockchain-based mobility management for 5g," *Future Generation Computer Systems*, vol. 110, pp. 638–646, 2020.

[170] D. C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, "UPORT: A PLATFORM FOR SELF-SOVEREIGN IDENTITY," p. 17.

[171] "A decentralized, open source solution for digital identity and access management." [Online]. Available: https://jolocom.io/wp-content/uploads/2019/12/Jolocom-Whitepaper-v2.1-A-Decentralized-Open-Source-Solution-for-Digital-Identity-and-Access-Management.pdf

[172] A. Tobin, D. Reed, F. P. J. Windley, and S. Foundation, "The Inevitable Rise of Self-Sovereign Identity," p. 24, 2017.

[173] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 103 059–103 079, 2019.

[174] "Travel identity of the future." [Online]. Available: https://blockchainlab.com/pdf/2016-05-00-idm-ShoCard-travel-identity-of-the-future.pdf

[175] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: Design and implementation of a global naming system with blockchains," *Last visited on*, vol. 25, no. 2, 2016.

[176] J.-H. Lee, "Bidaas: Blockchain based id as a service," *IEEE Access*, vol. 6, pp. 2274–2278, 2017.

[177] J.-H. Huh and K. Seo, "Blockchain-based mobile fingerprint verification and automatic log-in platform for future computing," *The Journal of Supercomputing*, vol. 75, no. 6, pp. 3123–3139, 2019.

[178] L. Wu, X. Du, W. Wang, and B. Lin, "An out-of-band authentication scheme for internet of things using blockchain technology," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 769–773.

[179] A. Mohsin, A. Zaidan, B. Zaidan, O. Albahri, A. Albahri, M. Alsalem, and K. Mohammed, "Based blockchain-pso-aes techniques in finger vein biometrics: A novel verification secure framework for patient authentication," *Computer Standards & Interfaces*, vol. 66, p. 103343, 2019.

[180] H.-W. Kim and Y.-S. Jeong, "Secure authentication-management human-centric scheme for trusting personal resource information on mobile cloud computing with blockchain," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–13, 2018.

[181] K. Xue, X. Luo, Y. Ma, J. Li, J. Liu, and D. S. Wei, "A distributed authentication scheme based on smart contract for roaming service in mobile vehicular networks," *IEEE Transactions on Vehicular Technology*, 2022.

[182] T. Sanda and H. Inaba, "Proposal of new authentication method in wi-fi access using bitcoin 2.0," in *2016 IEEE 5th Global Conference on Consumer Electronics*. IEEE, 2016, pp. 1–5.

[183] Y. Niu, L. Wei, C. Zhang, J. Liu, and Y. Fang, "An anonymous and accountable authentication scheme for wi-fi hotspot access with the bitcoin blockchain," in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2017, pp. 1–6.

[184] G. Ali, N. Ahmad, Y. Cao, M. Asif, H. Cruickshank, and Q. E. Ali, "Blockchain based permission delegation and access control in internet of things (baci)," *Computers & Security*, vol. 86, pp. 318–334, 2019.

[185] R. Xu, Y. Chen, E. Blasch, and G. Chen, "Blendcac: A smart contract enabled decentralized capability-based access control mechanism for the iot," *Computers*, vol. 7, no. 3, p. 39, 2018.

[186] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, "Exploiting smart contracts for capability-based access control in the internet of things," *Sensors*, vol. 20, no. 6, p. 1793, 2020.

[187] S. Dramé-Maigné, M. Laurent, and L. Castillo, "Distributed access control solution for the iot based on multi-endorsed attributes and smart contracts," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1582–1587.

[188] O. J. A. Pinno, A. R. A. Gregio, and L. C. De Bona, "Controlchain: Blockchain as a central enabler for access control authorizations in the iot," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[189] J. Zhang, Y. Yang, X. Liu, and J. Ma, "An efficient blockchain-based hierarchical data sharing for healthcare internet of things," *IEEE Transactions on Industrial Informatics*, 2022.

[190] X. Qin, Y. Huang, Z. Yang, and X. Li, "An access control scheme with fine-grained time constrained attributes based on smart contract and trapdoor," in *2019 26th International Conference on Telecommunications (ICT)*. IEEE, 2019, pp. 249–253.

[191] S. Alansari, F. Paci, and V. Sassone, "A distributed access control system for cloud federations," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 2131–2136.

[192] 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.

[193] Y. Le, X. Ling, J. Wang, and Z. Ding, "Prototype design and test of blockchain radio access network," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[194] X. Ling, Y. Le, J. Wang, Z. Ding, and X. Gao, "Practical modeling and analysis of blockchain radio access network," *IEEE Transactions on Communications*, 2020.

[195] K. Fan, Y. Ren, Y. Wang, H. Li, and Y. Yang, "Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5g," *IET communications*, vol. 12, no. 5, pp. 527–532, 2018.

[196] Q. Lyu, Y. Qi, X. Zhang, H. Liu, Q. Wang, and N. Zheng, "Sbac: A secure blockchain-based access control framework for information-centric networking," *Journal of Network and Computer Applications*, vol. 149, p. 102444, 2020.

[197] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for iot devices," *Applied Sciences*, vol. 10, no. 2, p. 488, 2020.

[198] T. Sultana, A. Ghaffar, M. Azeem, Z. Abubaker, M. U. Gurmani, and N. Javaid, "Data sharing system integrating access control based on smart contracts for iot," in *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. Springer, 2019, pp. 863–874.

[199] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2018.

[200] G. D. Putra, V. Dedeoglu, S. S. Kanhere, R. Jurdak, and A. Ignjatovic, "Trust-based blockchain authorization for iot," *IEEE Transactions on Network and Service Management*, 2021.

[201] H. Liu, D. Han, and D. Li, "Fabric-iot: A blockchain-based access control system in iot," *IEEE Access*, vol. 8, pp. 18 207–18 218, 2020.

[202] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for iot," *IEEE Access*, vol. 7, pp. 38 431–38 441, 2019.

[203] M. Yutaka, Y. Zhang, M. Sasabe, and S. Kasahara, "Using ethereum blockchain for distributed attribute-based access control in the internet of things," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[204] B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, "Iot passport: A blockchain-based trust framework for collaborative internet-of-things," in *Proceedings of the 24th ACM symposium on access control models and technologies*, 2019, pp. 83–92.

[205] D. Hwang, J. Choi, and K.-H. Kim, "Dynamic access control scheme for iot devices using blockchain," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 713–715.

[206] O. Alphand, M. Amoretti, T. Claeys, S. Dall'Asta, A. Duda, G. Ferrari, F. Rousseau, B. Tourancheau, L. Veltri, and F. Zanichelli, "Iotchain: A blockchain security architecture for the internet of things," in *2018 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2018, pp. 1–6.

[207] O. Novo, "Scalable access management in iot using blockchain: A performance evaluation," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4694–4701, 2018.

[208] A. R. Rajput, Q. Li, M. T. Ahvanooey, and I. Masood, "Eacms: Emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84 304–84 317, 2019.

[209] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Bedgehealth: A decentralized architecture for edge-based iomt networks using blockchain," *IEEE Internet of Things Journal*, 2021.

[210] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for large-scale internet of things data storage and protection," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 762–771, 2018.

[211] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao, and K. Yu, "Authprivacychain: A blockchain-based access control framework with privacy protection in cloud," *IEEE Access*, vol. 8, pp. 70 604–70 615, 2020.

[212] Y. Xiao, N. Zhang, J. Li, W. Lou, and Y. T. Hou, "Privacyguard: Enforcing private data usage control with blockchain and attested off-chain contract execution," in *European Symposium on Research in Computer Security*. Springer, 2020, pp. 610–629.

[213] Y. Zhu, Y. Qin, G. Gan, Y. Shuai, and W. C.-C. Chu, "Tbac: transaction-based access control on blockchain for resource sharing with cryptographically decentralized authorization," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2018, pp. 535–544.

[214] S. Wang, X. Wang, and Y. Zhang, "A secure cloud storage framework with access control based on blockchain," *IEEE Access*, vol. 7, pp. 112 713–112 725, 2019.

[215] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," p. 11.

[216] icoadmin, "What is IFPS? The hard drive for Blockchain." Aug. 2020. [Online]. Available: https://icommunity.io/en/what-is-ifps-the-hard-drive-for-blockchain/

[217] U. Fazil, "IPFS: A Distributed File Store," Feb. 2019. [Online]. Available: https://medium.com/block360-labs/ipfs-a-distributed-file-store-533cda4c6047

[218] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-based, decentralized access control for ipfs," in *2018 Ieee international conference on internet of things (iThings) and ieee green computing and communications (GreenCom) and ieee cyber, physical and social computing (CPSCom) and ieee smart data (SmartData)*. IEEE, 2018, pp. 1499–1506.

[219] P. Merson, "Data model as an architectural view," CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, Tech. Rep., 2009.

[220] P. P.-S. Chen, "The entity-relationship model—toward a unified view of data," *ACM transactions on database systems (TODS)*, vol. 1, no. 1, pp. 9–36, 1976.

[221] J. Chen, X. Xia, D. Lo, J. Grundy, X. Luo, and T. Chen, "Defining smart contract defects on ethereum," *IEEE Transactions on Software Engineering*, 2020.

[222] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "The keccak sha-3 submission," STMicroelectronics, 2NXP Semiconductors, Tech. Rep., 2011. [Online]. Available: https://keccak.team/index.html

[223] E. Rescorla *et al.*, "Diffie-hellman key agreement method," 1999. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2631

[224] "Countries with Mobile Number Portability - XConnect," July 2019, section: Blogs. [Online]. Available: https://www.xconnect.net/countries-with-mobile-number-portability/

[225] Y.-B. Lin, I. Chlamtac, and H.-C. Yu, "Mobile number portability," *IEEE network*, vol. 17, no. 5, pp. 8–16, 2003.

[226] D. Krishnaswamy, K. Chauhan, A. Bhatnagar, S. Jha, S. Srivastava, D. Bhamrah, and M. Prasad, "The design of a mobile number portability system on a permissioned private blockchain platform," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2019, pp. 90–94.

[227] "New mobile operator, mnp setup guide," MNP OSG, Tech. Rep., 2016. [Online]. Available: https://www.mnposg.org.uk/Main_Documents/New\%20Mobile\%20Operator\%20Guide\%201.1.pdf

[228] "Mobile number portability – review of the porting process," OfCom, Tech. Rep., 2009. [Online]. Available: https://www.ofcom.org.uk/__data/assets/pdf_file/0023/41765/mnpcondoc.pdf

[229] "Enisa threat landscape for 5g networks, threat assessment for the fifth generation of mobile telecommunications networks (5g)," ENISA, Tech. Rep., 2020. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-for-5g-networks/

[230] "OpenAirInterface –An Upstream Project for OPNFV – OpenAirInterface." [Online]. Available: https://openairinterface.org/community/whitepapers/openairinterface-an-upstream-project-for-opnfv/

[231] "Towards Open Cellular Ecosystem – OpenAirInterface." [Online]. Available: https://openairinterface.org/getting-started/openairinterface-an-open-cellular-ecosystem/

[232] M. Schäffer, M. Di Angelo, and G. Salzer, "Performance and scalability of private ethereum blockchains," in *International Conference on Business Process Management*. Springer, 2019, pp. 103–118.

[233] P. W. Eklund and R. Beck, "Factors that impact blockchain scalability," in *Proceedings of the 11th international conference on management of digital ecosystems*, 2019, pp. 126–133.

[234] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Network*, vol. 33, no. 5, pp. 166–173, 2019.

[235] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, "Exploring the attack surface of blockchain: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1977–2008, 2020.

[236] "5g; security architecture and procedures for 5g system (3gpp ts 33.501 version 15.2.0 release 15)," European Telecommunications Standards Institute (ETSI), Tech. Rep., 10 2018.

[237] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, "Security for 5g and beyond," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3682–3722, 2019.

[238] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: A comprehensive survey," *IEEE Access*, vol. 8, pp. 125 244–125 262, 2020.

[239] M. H. Nasir, J. Arshad, M. M. Khan, M. Fatima, K. Salah, and R. Jayaraman, "Scalable blockchains—a systematic review," *Future Generation Computer Systems*, vol. 126, pp. 136–162, 2022.

[240] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *Ieee Access*, vol. 8, pp. 16 440–16 455, 2020.

[241] G. Kaur and C. Gandhi, "Scalability in blockchain: Challenges and solutions," in *Handbook of Research on Blockchain Technology*. Elsevier, 2020, pp. 373–406.

[242] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 931–948.

[243] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proceedings of the 2019 international conference on management of data*, 2019, pp. 123–140.

[244] P. Zheng, Q. Xu, Z. Zheng, Z. Zhou, Y. Yan, and H. Zhang, "Meepo: Sharded consortium blockchain," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 1847–1852.

[245] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "Sok: Sharding on blockchain," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 41–61.

[246] "Radio Access Network (RAN) Hardware and Software." [Online]. Available: https://www.sdxcentral.com/5g/ran/definitions/radio-access-network-ran-hardware-software/

[247] Y. Wang, S. Cai, C. Lin, Z. Chen, T. Wang, Z. Gao, and C. Zhou, "Study of blockchains's consensus mechanism based on credit," *IEEE Access*, vol. 7, pp. 10 224–10 231, 2019, publisher: IEEE.

[248] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Blockchain and machine learning for communications and networking systems," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1392–1431, 2020, publisher: IEEE.

[249] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, "A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–32, 2020, publisher: ACM New York, NY, USA.

[250] S. M. H. Bamakan, A. Motavali, and A. Babaei Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, p. 113385, Sept. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420302098

[251] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, "Exploring the Attack Surface of Blockchain: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1977–2008, 2020, conference Name: IEEE Communications Surveys Tutorials.

[252] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Lsb: A lightweight scalable blockchain for iot security and anonymity," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 180–197, 2019.

[253] S. K. Lo, Y. Liu, S. Y. Chia, X. Xu, Q. Lu, L. Zhu, and H. Ning, "Analysis of blockchain solutions for iot: A systematic literature review," *IEEE Access*, vol. 7, pp. 58 822–58 835, 2019.

[254] "We (Finally) Built Eris!" [Online]. Available: https://blog.erisindustries.com/products/2015/08/19/v010-release/

[255] A. Menezes, P. Van Oorschot, and S. Vanstone, "Chapter 12 key establishment protocols," *Handbook of Applied Cryptography*, pp. 489–541, 1997.

[256] M. A. Islam and S. Madria, "A permissioned blockchain based access control system for iot," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 469–476.

[257] N. Tapas, F. Longo, G. Merlino, and A. Puliafito, "Experimenting with smart contracts for access control and delegation in iot," *Future Generation Computer Systems*, vol. 111, pp. 324–338, 2020.

List of figures

List of tables

Appendix A

Comparison of consensus models

The aforementioned consensus models are listed and compared in Table A.1, based on the following parameters:

- *BFT:* The maximum tolerable rate of Byzantine nodes in the system [51, 247];
- *Scalability:* The system's ability to tolerate increasing numbers of nodes [248];
- *Throughput/ transaction rate:* The average number of transactions validated in one second [247];
- *Recourse consumption:* The amunt of resources needed for a method's operation [65, 69]; and
- *Recourse type:* The types of resources needed to run each method by a specific node (e.g., computational power, reputation, stake).

Table A.1: Comparison of Existing Consensus Models [40, 51, 75–78]

Type	***Feature Method***	***BFT***	***Scalability***	***Throughput***	***Resource consumption***	***Resource type***	***Example***
Compute-intensive	PoW	49%($N/2$)	High	Low	High	Computational power	Bitcoin, Ethereum
Capability-based	PoS	49%($N/2$)	High	Low	Moderate	Stake	Ethereum, peerCoin, Nxt
	PoA [86]	49%($N/2$)	High	Moderate	Low	Stake and Reputation	VeChain
	PoET	-	-	High	Moderate	Time	Hyperledger Sawtooth
	PoI [249, 250]	49%($N/2$)	High	High	Low	Importance mark	NEM
	Ouroboros	33%($N/3$)	High	-	Low	Stake	Cardano
Voting-based	DPoS	49%($N/2$)	High	Moderate	Moderate	Reputation	EOS, Nano
	Raft [251]	49%($N/2$)	Low	Higher than PBFT	Moderate	Time	Quorum
	PBFT	33% ($N/3$)	Low	High	Low	None	Hyperledger Fabric

Appendix B

Comparison of DLT-based authentication methods

Table B.1: Comparison of existing authentication methods based on DLT

Auth. Type	***App.***	***Refs.***	***App. Env.***	***Auth. step***	***BC Plat.***	***Cons. Model***	***Pros***	***Cons***
Possession-based	Distributed database	[176]	*Communication networks*- ID generation	– Recording and generating user identification	Not mentioned	PBFT	– Proposing novel idea on IDaaS – Does not use pre-registered user information	– Does not protect user privacy – No security analysis
		[167]	*Communication networks*- handover	– Recording user's identification – Recording authentication logs	Not implemented		– Provides seamless handover – Low storage complexity – Perfect forward secrecy – Prevents Replay, password cracking, DoS/DDoS, and Sybil attacks	– None identified to date
		[166]	*Communication networks*- handover	– Recording user's identification	Ethereum	DPoS	– Provides acceptable handover delay – Protects user privacy – Resistant against DoS/DDoS, Spoofing, Link-ability and Numb attacks	– None identified to date

Auth. Type	*App.*	*Refs.*	*App. Env.*	*Auth. step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
Possession-based	Distributed database	[168]	*IoT (WSN)*-Mobility management	– Recording user's identification	Ethereum	Not mentioned	– Cryptanalysis is not possible – Energy efficient – Resistant against password cracking, DoS/DDoS, and Replay attacks	– System performance is not evaluated – Low scalability based on the context – Not sufficient security analysis
		[159]	*ICN*- Mobility management	– Recording user's identification	Not mentioned	Time based model [252]	– Resistant against password cracking, DoS/DDoS, Replay, Sybil, prefix hijacking, deppending, and packet discarding attacks – The administrator cannot falsify the node's reputation	– Single authorization server can be a single point of failure – Not sufficient security analysis
		[129]	*Cloud*- Access to resources	– Recording user's identification	Not implemented		– Mechanism is robust and secure – Resistant against major DLT attacks – Resistant against password cracking, DoS/DDoS, MitM, Replay and Spoofing attacks	– System performance is not evaluated

Auth. Type	*App.*	*Refs.*	*App. Env.*	*Auth. step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
Possession-based	Distributed DB & verification	[182]	*Communication-* Connection to the internet	– Recording user's identification – User verification – Token issuing	Bitcoin	PoW	– Provides mutual authentication – Does not need user information – Protects user privacy [68]	– Uses Bitcoin address as encryption key – No security analysis
		[181]	*Communication-* Mobility management	– Recording roaming session key – Token issuing	Ethereum	PoW	– Provides mutual authentication – Resistant against MitM, Replay and modification attacks	– None identified to date
		[147]	*IoT-* Access to IoT sensor	– Recording user's identification – User verification – Token issuing	Ethereum	PoW [253]	– Modification of the signed authentication message is impossible – High availability and scalability – Resistant against password cracking, MitM and Replay attacks	– High computational cost – Not sufficient security analysis
		[169]	*Communication networks-* user mobility management	– Key handover by mining	Bitcoin	PoW	– Resilient against password cracking, MitM, Spoofing, desynchronization and rogue base station attacks	– None identified to date

Auth. Type	*App.*	*Refs.*	*App. Env.*	*Auth. step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
Know-ledge-based	Distributed database	[183]	*Communication*-Wi-Fi hotspot access	– Recording user's identification – User verification	Bitcoin	PoW	– Provides accountability & anonymity – Provides suggestions for mutual authentication (without detail)	– Not sufficient security analysis
Inherent-based	Distributed DB	[177]	*IoT*-Access to IoT sensors	– Recording user's identification	Bitcoin	PoW	– Guarantees tamper-free credentials – Prevents data leakage – Uses biometric authentication	– High resource consumption – Not sufficient security analysis
multi-factor	Distributed database	[178]	*IoT*-Access to IoT sensors	– Recording identification data – Recording authentication log	Eris [254]	PoS (Eris)	– Low resource consumption – Multi-factor authentication – Uses out-of-band authentication – Resistant against password cracking	– Single point of failure – Not sufficient security analysis
		[179]	*Smart helthcare*-Access to medical record via IoT	– User verification	Not mentioned	Not mentioned	– Protects user privacy – Resistant against password cracking and Spoofing attacks	– Not sufficient analysis on Blockchain size and scalability – Not sufficient security analysis

Auth. Type	*App.*	*Refs.*	*App. Env.*	*Auth. step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
multi-factor	Distributed database	[146]	*All use-cases*-single authentication	– Recording user's identification	Bitcoin	Ouroboros [98]	– Mutual authentication and privacy – Supports forward secrecy [255] – Resistant against password cracking, MitM, Replay and Spoofing attacks	– None identified to date
		[180]	*Cloud*-Access to the resources	– Recording user's identification	Not implemented		– Impossibility of data falsification – Uses Multi-factor authentication – Resistant against MitM	– Single point of failure (master node) – Not sufficient security analysis

Appendix C

Comparison of DLT-based Access control methods

Table C.1: Comparison of existing access control methods based on DLT

AC Type	***App.***	***Refs.***	***App. Env.***	***AC step***	***BC Plat.***	***Cons. Model***	***Pros***	***Cons***
Generic (support all mechanisms)	DDB*	[184]	*IoT/ Smart City-* right delegation	– Policy Storing – Stores access logs	Hyperledger Fabric	PBFT	– Low resource consumption – High efficiency	– Saves huge amounts of data in Blockchain (low scalability)
		[189]	*Healthcare-* Data sharing	– Policy storing – Recording logs	Ethereum	PoS	– Low time consumption	– Non identified to date
	AC Mng.**[165]		*Communication-* Resource access	– Medium access validation	Ethereum	PoD	– Prevents rogue devices from exhibiting selfish behaviors	– The method inherits the problems of [192–194]
	DDB & AC Mng.***	[192–194]	*Communication-* Resource access	– Policy storing – Access validation	Ethereum	PoD [253]	– High efficiency	– Low scalability – Low service quality
		[206]	*IoT-* Access to IoT sensor	– Policy storing – Token issuing	Ethereum	Proof-of-Possession (PoP)	– Uses encrypted storage – Protects user privacy – Resilient to DoS and MitM attacks	– Depends on an intermediary entity for key distribution [?]

AC Type	*App.*	*Refs.*	*App. Env.*	*AC step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
Generic (support all mechanisms)	DDB & AC Mng.***	[211]	*Cloud*- Access to cloud resource	– Permission storing – Access validation, – Access log recording	EOS (Kylin/ Jungle)	Not mentioned	– Protects user privacy – Resistant against internal attacks – Confidentiality, integrity, availability, authenticity, and accountability are provided	– Does not provide any security solution for secret key sharing
		[197, 198]	*IoT*- Data sharing	– Policy storing – Access validation – Access log recording	Ethereum	Not mentioned	– High reliability – Trustworthy system	– Does not protect user privacy
		[209]	*Smart healthcare*- Data sharing	– Policy storing – Access validation – Access log recording	Hyperledger Fabric	PBFT	– Data privacy	– None identified to date
		[208]	*Smart healthcare*- Data sharing	– Policy storing – Access validation – Access log recording	Hyperledger Fabric	Not mentioned	– Data privacy	– Insufficient security analysis

AC Type	*App.*	*Refs.*	*App. Env.*	*AC step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
DDB		[190]	*Cloud*-Data sharing	– Policy storing	Ethereum	Not mentioned	– Low computational cost – ABE provides confidentiality	– The CA is a single point of failure
		[191]	*Cloud*-data and resource sharing	– Storing of policies and attributes	Proposed as a framework		– Provides secure resource sharing – Protects data and user's privacy	– Low scalability
		[188]	*IoT*- Access to IoT sensor	– Policy and credential storing – Access logs recording	Not mentioned		– Compatible with a wide range of access control mechanisms – High scalability	– Four Blockchains must be used [?] – Efficiency is not proven [?] – Access engine is SPoF
ABAC	AC Mng.	[196]	*ICN*-Data sharing	– Access validation	Ethereum	PoS	– Multi-level content access – Resistant against Cache poisoning, DDoS and MitM attacks	– Low scalability [?]
		[187]	*IoT*- Access to IoT sensor	– Access validation	Not mentioned		– Multiple domain access – Scalable and flexible – Mitigates reply attacks	– High time consumption – Information leakage is not addressed – Lack of implementation details

AC Type	*App.*	*Refs.*	*App. Env.*	*AC step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
ABAC	DDB & AC Mng.	[199]	*IoT*- Data sharing	– Policy storing – Access validation – Access log recording	Ethereum	PoW	– Implements trustworthy access control for IoT systems using smart contracts	– Does not protect user privacy – High transaction fees [?, 197] – Limited environment attributes
		[202]	*IoT*- Access to IoT sensor	– ID storing – Access verification	Hyperledger Fabric	PBFT	– Avoids data tampering – Lightweight calculation	– High message passing in the network – No efficient consensus [197]
		[200]	*IoT*- data access	– Policy storing – Access verification	Rinkbey	PoA	– Provides flexible access control – Resistant against self-promoting, and Ballot-stuffing attacks.	– Low scalability – Attribute authority is a single point of failure
		[201]	*IoT*- Access to IoT sensor's data	– Stores policy & URL – Access verification	Hyperledger Fabric	Kafka	– Lightweight computation – Dynamic permissions	– Performance is not proven – Low scalability
		[213]	*Cloud*- Cloud resource sharing	– Rule storing – Access validation – Access log recording	Not mentioned		– Supports flexible and diverse permission management	– Performance is not proven

AC Type	*App.*	*Refs.*	*App. Env.*	*AC step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
ABAC — DDB & AC Mng.		[212]	*Cloud*-Data sharing	– Policy & log storing – Access validation	Ethereum	PoA	– Protects user and data privacy – Provides secure payment	– Requires specific hardware [**?**]
		[256]	*IoT*- Access to IoT device	– Access validation – Access log recording	Hyperledger fabric	PBFT	– High scalability – High performance	– High latency
		[203]	*IoT*- Access to IoT device	– Policy storing – Access validation	Ethereum	Not mentioned	– Acceptable scalability	– High latency for access control – Does not protect user privacy
		[195]	*5G*-Data sharing	– Policy storing – Access validation	Not mentioned		– Protects providers' privacy – Provides forward secrecy	– Low scalability (because of block size limitation)
		[204]	*IoT*-Providing trust	– Policy storing – Access validation	Not mentioned		– Secure interactions [**?**] – Uses an incentive mechanism	– No implementation – No analysis
CapBAC — DDB		[185]	*IoT*- Access right delegation	– Policy storing – Storing capabilities	Ethereum	PoW	– High scalability – Hierarchical delegation	– A subject can only obtain rights from one subject [186] – High resource consumption

AC Type	*App.*	*Refs.*	*App. Env.*	*AC step*	*BC Plat.*	*Cons. Model*	*Pros*	*Cons*
CapBAC	DDB	[186]	*IoT*- Access right delegation	– Records capability tokens & access matrix	Ethereum	Not mentioned	– Supports multi-delegation [?] – No limitation in right delegation	– Tokens are stored in Blockchain with no encryption
RBAC	DDB & AC Mng.	[257]	*IoT*- Access delegation	– Access validation	Ethereum	PoW	– High fault tolerance – Time efficiency	– High energy consumption
		[205]	*IoT*- Access to IoT device's data	– Policy storing – Access validation	No implementation		– Dynamic access control policy generation	– No implementation
ACL	DDB	[210]	*Smart healthcare*- Access to EHR	– Policy storing – Access validation	Not implemented		– High traceability, and scalability – Certificate-less cryptography	– Lack of details on its implementation in Blockchain
	DDB & AC Mng.	[207]	*IoT*- Access to wireless sensors	– ACL storing	Ethereum	Not mentioned	– Low energy consumption – High scalability – Low latency	– Low performance because of RPC – Single management hub [197]

* DLT is used as only Distributed DataBase.
** DLT is used as a solution for ACcess Management.
*** DLT is used as both Distributed DataBase and a solution for ACcess Management.

Appendix D

User experienced latency

In this chapter, we provide the user experience latency for the service provisioning scenario in which the user's access to the service and payment is done in smart contracts (see Fig. 6.6 and Fig. 6.7).
User's experienced latency of the proposed method for private use cases is provided in Figure D.1 regarding the different scenarios and the aforementioned performance indicators. Each bar in the figure indicates how the user experienced latency is decomposed to different times (i.e., $T_{net}, T_{dapp}, T_{fn}$).
Note that, the utilized configuration for this analysis is applicable in private or semi-private use cases when one/several companies govern the Blockchain and have the right to participate in consensus procedure and write into the Blockchain. What is special in this kind of use-cases is the level of security requirement (i.e., to protect the system against the intrinsic attacks of Blockchain, such as 51% attack [**?**]) in the system. Since in these use cases, the participating nodes are already authenticated and are known in the system, there is a minimum level of trust (i.e., the simple consensus procedure is sufficient for these networks). To simulate the simple consensus procedure, we opt for a minimum value for the `block time` which defines the complexity of the consensus procedure. As is shown in Figure D.1, the latency of function execution (i.e., the average of 10 times of function execution) to provide the user's access to the system is very low compared to the network and DApp latency. Moreover, the user's experienced latency for the access control procedure is around $3s$ which is well comparable with the existing centralized systems. Another important point to mention is the real latency of the DApp, which has a significant impact on the user's real experience time. Hence, in the real implementation of our method, many of the extra procedures in the DApp do not need to be executed. As a result, the value of T_{total} can significantly decrease.
Figure D.2 provides the performance analysis for public use cases, in which the

Blockchain governance is not in the hands of several organizations, and every micro business or even users can use it as well. Under this scenario, we cannot assume any trust level. The actors need to protect the system against Blockchain attacks [235], by enforcing some more complex and secure consensus procedures. To simulate a complex consensus procedure, we choose the `block time` as 5 s that is near to public networks. As is shown in Figure D.1, in this case, the latency of function execution to provide the user's access to the system is a very high portion of the user's real experienced latency. To have a more secure network, we assume that the nodes of the network can update their ledger after two or three `block time`. Hence, when we select `block time` as $5s$, the minimum time by which the user can see the result of her transaction is $3 \times BlockTime$ which in our case (i.e., in Ethereum with PoW consensus model) it is $15s$. Indeed, this latency is higher than the expectation of the user and the experienced latency in centralized systems.

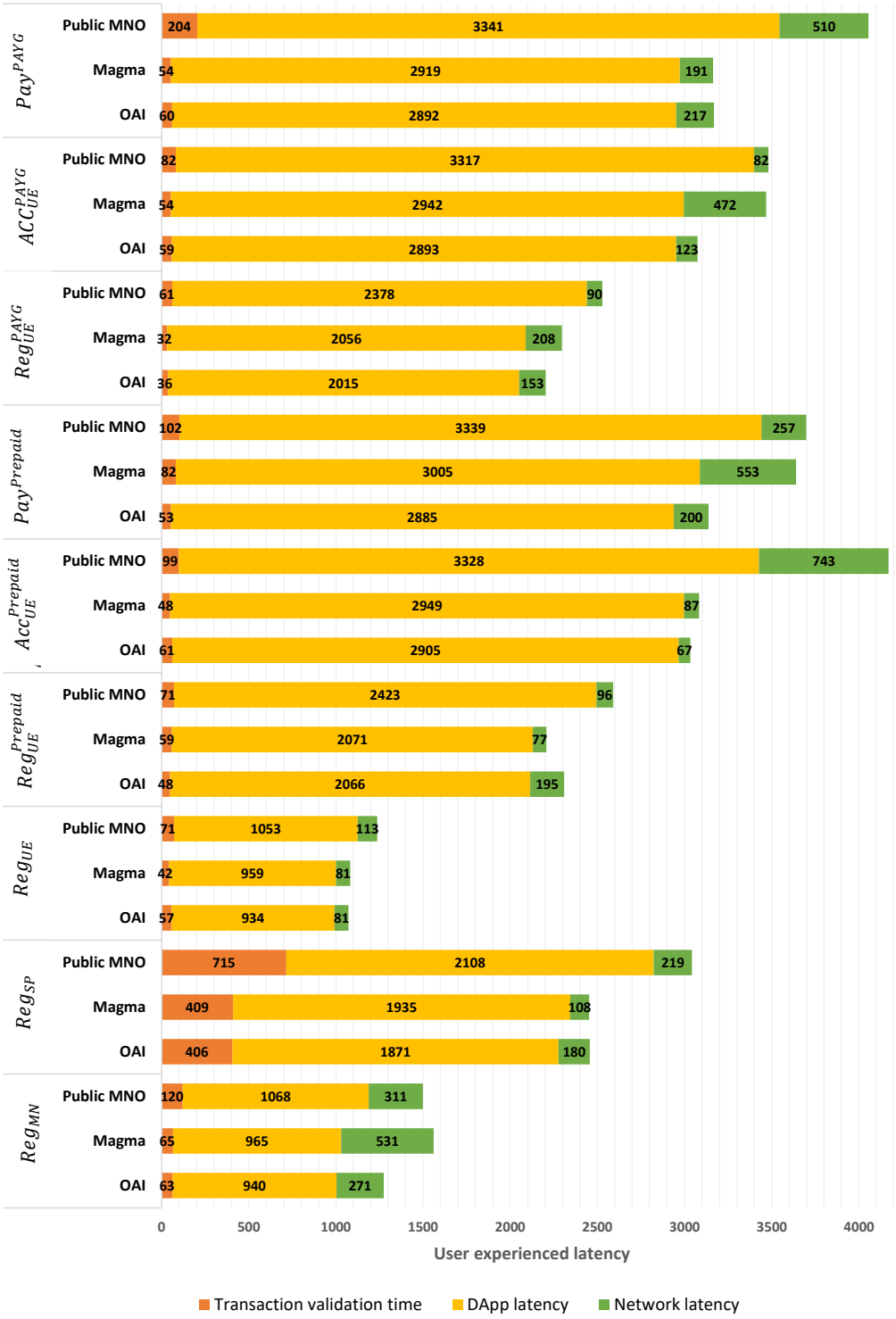


Figure D.1: The user experienced latency of the system for the private or consortium *dlt use-case, with low-security and high throughput requirements. Each bar in the figure represents the latency experienced by the user, and it is made up of Network latency (T_{net}), DApp latency (T_{dapp}), and Transaction validation latency (T_{fn}).

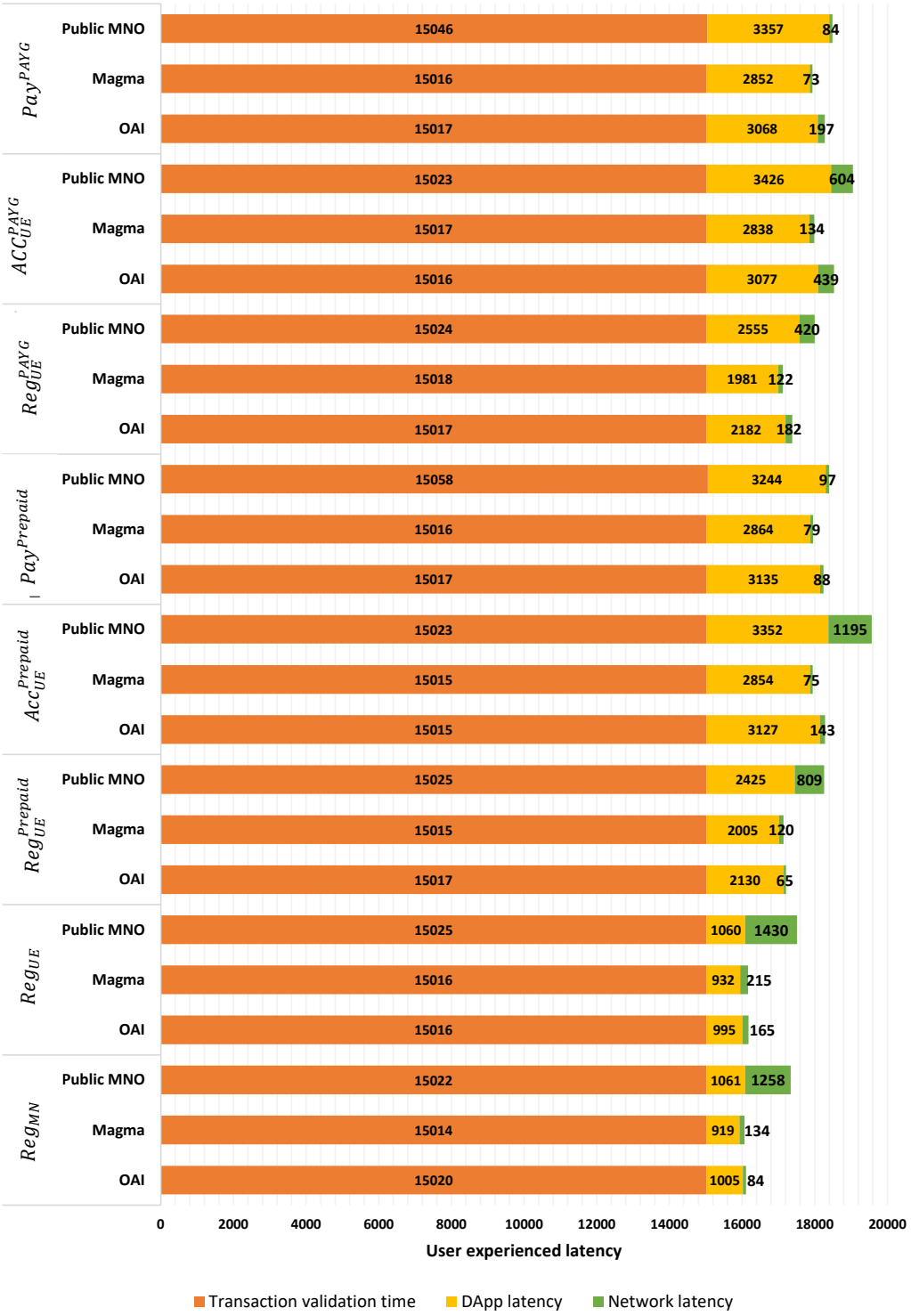


Figure D.2: The latency of the system for public Blockchain use-case with high-security requirements. Each bar in the figure represents the latency experienced by the user, and it is made up of Network latency (T_{net}), DApp latency (T_{dapp}), and Transaction validation latency (T_{fn}).

Titre : Une nouvelle architecture basée sur la Blockchain pour les opérateurs de réseaux mobiles : au-delà de la 5G

Mots clés : Système de connectivité multi-acteurs, Blockchain, Business model, Sécurité.

Résumé : Les Opérateurs de Réseaux Mobiles (ORM) fournissent à chaque instant de la connectivité à des milliards d'utilisateurs, en se basant sur des architectures centralisées dont certains des principes fondateurs ont été conçus en standardisation dans les 1980 et 1990. Cela a des impacts sur les coûts de mutualisation, ainsi que sur la consommation d'énergie et l'impact environnemental. Cela complexifie également la mise en place de modèles d'affaire plus collaboratifs avec d'autres fournisseurs. De plus, la complexité des réseaux 5G et au-delà de la 5G peut dépasser la capacité d'un ORM à gérer le coût et la complexité de la connexion pour un grand nombre d'éléments interconnectés. Enfin, de par leur centralisation, les architectures ORM existantes peuvent être sujettes à des risques techniques et à des vulnérabilités. Bien que les systèmes actuels soient fonctionnels et performants, une étude de principes d'architecture alternatifs, basés sur les acquis des systèmes distribués, semble importante à réaliser dans la perspective de l'après 5G est de la 6G. C'est l'objet de ce travail doctoral. Relever ces défis n'est pas une démarche simple. Cependant, nous croyons qu'il y a un intérêt à proposer au sein de la communauté de la recherche en télécommunications une approche nouvelle, en repartant des besoins et non des architectures existantes. Premièrement, nous proposons une étude approfondie des défis existant dans les réseaux cellulaires actuels concernant les aspects commerciaux et de collaboration. Les résultats de cette étude nous ont amenés à proposer deux contributions principales. Notre première contribution concerne la coopération entre les différents acteurs de l'écosystème des réseaux cellulaires. Notre deuxième contribution concerne la collaboration entre ORM pour la gestion des identités et des profils. Les résultats de l'évaluation montrent que le système est suffisamment évolutif en termes de nombre d'acteurs et de collaborateurs, et qu'en fonction des exigences du réseau, ses performances et son niveau de sécurité sont réglables. De plus, l'analyse de sécurité montre que le système est résilient aux fils conducteurs de communication courants. Enfin, les obstacles et les limites de la mise en œuvre réelle de la nouvelle architecture en termes de latence, d'évolutivité, de standardisation, d'exigences de stockage et d'incitations pour les différentes parties sont discutés.

Title : A novel Blockchain-based Architecture for Mobile Network Operators: Beyond 5G

Keywords : Multi-actor connectivity system, Blockchain, Business model, Security.

Abstract : Mobile Network Operators (MNOs) provide connectivity to billions of users at all times, based on centralized architectures, some of whose founding principles were designed as standardization in the 1980s and 1990s. This has impacts on mutualization costs, as well as on energy consumption and environmental impact. This also complicates the implementation of more collaborative business models with other providers, or even with business customers. Moreover, the complexity of 5G and beyond 5G networks may surpass the capability of one MNO to manage the cost and the complexity of connection. Finally, due to their centralization, existing MNO architectures can be subject to technical risks and vulnerabilities. Although the current systems are functional and efficient, a study of alternative architecture principles, based on the achievements of distributed systems, seems important to carry out in the perspective of after 5G and 6G. This is the subject of this doctoral work. Addressing these challenges is not a straightforward journey. However, we believe that there is an interest in proposing within the telecommunications research community a new approach, starting from needs and not from existing architectures. First, we propose a comprehensive study of the challenges existing in current cellular networks regarding the business and collaboration aspects The results of this study led us to propose two main contributions. Our first contribution concerns the cooperation between the different actors of the cellular network ecosystem. Our second contribution concerns the collaboration between MNOs for the management of identities and profiles. The evaluation results show that the system is scalable enough regarding the number of actors and collaborators, and based on the network requirements, its performance and security level are adjustable. Moreover, security analysis shows that the system is resilient against common threads for communication. Finally, the obstacles and limitations of real-world implementation of the novel architecture regarding latency, scalability, standardization, storage requirements, and incentives for different parties are discussed.

Institut Polytechnique de Paris
91120 Palaiseau, France