UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingenieros Informáticos

DECENTRALIZED SYSTEMS FOR THE PROTECTION AND PORTABILITY OF PERSONAL DATA

TESIS DOCTORAL

Mirko Zichichi
Licenciado en Informática

2023

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos

PhD Thesis

Decentralized Systems for the Protection and Portability of Personal Data

Author: Mirko Zichichi

Supervisors: Dr. Víctor Rodríguez Doncel
Dr. Stefano Ferretti

January, 2023

Tribunal nombrado por el Sr. Rector Magfco. de la Universidad Politécnica de Madrid, el día __ de __ de ____

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Suplente:

Suplente:

Realizado el acto de defensa y lectura de la Tesis el día __ de __ de ____ en la Escuela Técnica Superior de Ingenieros Informáticos

Calificación: ____

EL PRESIDENTE VOCAL 1 VOCAL 2

VOCAL 3 EL SECRETARIO

Thanks to all the people who supported me during the creation of this work. I am particularly grateful to both my supervisors and all the people involved in the LAST-JD-RIoE program, the members of the Analysis of Networks and Simulation Research Group and of the Legal Blockchain Lab at the University of Bologna, and the members of the Ontology Engineering Group at the Universidad Politécnica de Madrid.

Acknowledgments - Agradecimientos

This work received funding from the European Union Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Actions – Innovative Training Networks European Joint Doctorate with grant agreement No. 814177 Law Science and Technology Joint Doctorate - Rights of the Internet of Everything.

–

Este trabajo ha recibido financiación del programa de investigación e innovación de la Unión Europea Horizon 2020 en el marco de las Marie Sklodowska-Curie Actions - Innovative Training Networks European Joint Doctorate con el acuerdo de subvención No. 814177 Law Science and Technology Joint Doctorate - Rights of the Internet of Everything.

Abstract

The transformation introduced by information communication technologies in the last decades significantly impacts the economy and society concerning the digital representation of one's identity. The economics of exploiting personal information is assisted by the more pervasive nature of today's digital world: data are at the center of this transformation, and individuals are the primary sources of information and the ones most affected by it. The General Data Protection Regulation (GDPR) is having a significant impact on the protection of personal data. It has been designed for European Union (EU) citizens to help promote a view in favor of the interests of individuals instead of large corporations. However, at a large scale, there need to be more dedicated technologies that can help companies comply with GDPR (and similar regulations) while enabling people to exercise their rights. We argue that such a dedicated solution must address two main issues: the need for more transparency towards individuals regarding the management of their personal information and their often hindered ability to access and make interoperable personal data in a way that the exercise of one's rights would result in straightforward and not excessively burdensome.

In this work, we make the first step toward these two objectives by designing a user-centered model for managing personal data, where storage is decoupled from the data management logic. We aim to provide a system that helps to push personal data management towards the individual's control, i.e., a personal information management system (PIMS). By using distributed storage and decentralized computing networks to control online services, users' personal information could be shifted towards those directly concerned, i.e., the data subjects. The use of Distributed Ledger Technologies (DLTs) and Decentralized File Storage (DFS) as an implementation of decentralized systems is of paramount importance in this case.

The structure of this dissertation follows an incremental approach to describing a set of decentralized systems and models that revolves around personal data and their subjects. Each chapter of this dissertation builds up the previous one and discusses the technical implementation of a system and its relation with the corresponding regulations. Indeed, most of the time, implementations based on decentralized systems clash with laws such as GDPR. We refer to the EU regulatory framework, including GDPR, eIDAS, and Data Governance Act, to build our final system architecture's functional and non-functional drivers.

In our PIMS design, personal data is kept in a Personal Data Space (PDS) consisting of encrypted personal data referring to the subject stored in a DFS. We use a decentralized indexing system to guarantee the integrity, verifiability, linkability, searchability, and indexing of the encrypted personal data stored in the PDS. We follow the approach to reference data and their content on a DLT, i.e., on-chain hash pointers. Then, we associate to such hash pointer reference a keyword set that is exploited to lookup for specific kinds of contents. On top of that, a network of authorization servers acts as a data intermediary to provide access to potential data recipients. Access to the data stored on a PDS can be allowed by the data holder through smart contracts. These maintain a data structure to record eligible data recipients, i.e., those to whom to issue the keys needed to access the encrypted data. Also, in this case, the GDPR tensions with DLTs drove the architecture to be multi-DLT. Authorization servers use a "tightly controlled" permissioned DLT to handle personal data, while a permissionless "audit" DLT is used for system security and only holds non-personal data.

After describing the design of a decentralized PIMS, we focus on enriching the expressiveness of the access control mechanism through privacy policies. These let data subjects and/or holders express privacy policies aligned with the GDPR legal bases to be enforced through smart contracts. We use a set of Semantic Web technologies and standards for this aim. Finally, we present a Self-Sovereign Identity (SSI) model for providing, requesting, and obtaining qualified data to negotiate and/or execute electronic transactions with a focus on the legal and operational context.

Resumen

La transformación introducida por las tecnologías de la información y las comunicaciones en las últimas décadas repercute significativamente en la economía y la sociedad en lo que respecta a la representación digital de la identidad personal. La explotación económica de la información personal se ve favorecida por la naturaleza omnipresente del mundo digital actual: los datos están en el centro de esta transformación, y los individuos son las principales fuentes de información y los más afectados por ella. El Reglamento General de Protección de Datos (RGPD) está teniendo un impacto significativo en la protección de los datos personales. Ha sido diseñado para los ciudadanos de la Unión Europea (UE) con el fin de ayudar a promover una visión a favor de los intereses de los individuos en lugar de las grandes corporaciones. Sin embargo, a gran escala, es necesario que existan más tecnologías dedicadas que puedan ayudar a las empresas a cumplir con el RGPD (y reglamentos similares) al tiempo que permiten a las personas ejercer sus derechos. Sostenemos que una solución específica de este tipo debe abordar dos cuestiones principales: la necesidad de una mayor transparencia hacia las personas en lo que respecta a la gestión de su información personal y su capacidad, a menudo obstaculizada, de acceder a los datos personales y hacerlos interoperables de forma que el ejercicio de los derechos propios resulte sencillo y no excesivamente oneroso.

En este trabajo, damos el primer paso hacia estos dos objetivos diseñando un modelo de gestión de datos personales centrado en el usuario, en el que el almacenamiento se desvincula de la lógica de gestión de datos. Pretendemos ofrecer un sistema que ayude a impulsar la gestión de datos personales hacia el control del individuo, es decir, un sistema de gestión de información personal (PIMS). Al utilizar el almacenamiento distribuido y las redes de computación descentralizadas para controlar los servicios online, la información personal de los usuarios podría desplazarse hacia

los directamente interesados, es decir, los interesados. El uso de las *Distributed Ledger Technologies* (DLT) y del *Decentralized File Storage* (DFS) como implementación de sistemas descentralizados es de vital importancia en este caso.

La estructura de esta disertación sigue un método incremental para describir un conjunto de sistemas y modelos descentralizados que gira en torno a los datos personales y sus sujetos. Cada capítulo de esta disertación se basa en el anterior y analiza la implementación técnica de un sistema y su relación con la normativa correspondiente. De hecho, la mayoría de las veces, las implementaciones basadas en sistemas descentralizados chocan con leyes como el RGPD. Nos referimos al marco normativo de la UE, incluidos el RGPD, el eIDAS y la Data Governance Act, para construir los impulsores funcionales y no funcionales de nuestra arquitectura final del sistema.

En nuestro diseño de PIMS, los datos personales se guardan en un Espacio de Datos Personales (PDS) que consiste en datos personales cifrados referentes al sujeto almacenados en un DFS. Utilizamos un sistema de indexación descentralizado para garantizar la integridad, verificabilidad, vinculabilidad, capacidad de búsqueda e indexación de los datos personales cifrados almacenados en el PDS. Seguimos el enfoque para referenciar datos y su contenido en una DLT, es decir, *on-chain hash pointers*. A continuación, asociamos a dicha referencia de *hash pointer* un conjunto de palabras clave que se explota para buscar tipos específicos de contenidos. Además, una red de servidores de autorización actúa como intermediaria para facilitar el acceso a los posibles destinatarios de los datos. El acceso a los datos almacenados en un PDS puede ser permitido por el titular de los datos a través de *smart contracts*. Estos mantienen una estructura de datos para registrar los destinatarios de datos elegibles, es decir, aquellos a quienes expedir las claves necesarias para acceder a los datos cifrados. Además, en este caso, las tensiones del RGPD con las DLT impulsaron a que la arquitectura fuera *multi-DLT*. Los servidores de autorización utilizan una *permissioned DLT* "estrechamente controlada" para manejar datos personales, mientras que

una *"audit" permissionless DLT* se utiliza para la seguridad del sistema y sólo contiene datos no personales.

Tras describir el diseño de un PIMS descentralizado, nos centramos en enriquecer la expresividad del mecanismo de control de acceso mediante políticas de privacidad. Éstas permiten a los titulares y/o sujetos de los datos expresar políticas de privacidad alineadas con las bases legales del RGPD que se aplicarán a través de *smart contracts*. Para ello utilizamos un conjunto de tecnologías y estándares de la Web Semántica. Por último, presentamos un modelo de *Self-Sovereign Identity* (SSI) para proporcionar, solicitar y obtener datos cualificados para negociar y/o ejecutar transacciones electrónicas con un enfoque en el contexto legal y operativo.

Contents

List of Figures

List of Tables

List of Abbreviations

ACL	Access Control List
AKN	AKoma Ntoso
CEL	Contract Expression Language
CNIL	Commission Nationale de l'Informatique et des Libertes
DAO	Decentralized Autonomous Organization
dApp	Decentralized Application
DAG	Directed Acyclic Graph
DFS	Decentralized File Storage
DGA	Data Governance Act
DHT	Distributed Hash Table
DID	Decentralized IDentifier
DLT	Distributed Ledger Technology
DPA	Data Processing Agreement
DPV	Data Privacy Vocabulary
EBSI	European Blockchain Services Infrastructure
EDPB	European Data Protection Board
eIDAS	electronic IDentification Authentication and Signature
ENISA	European Union Agency for Cybersecurity
EU	European Union
GDPR	General Data Protection Regulation
IAPP	International Association of Privacy Professionals
ICT	Information Communication Technology
IPFS	InterPlanetary File System
IRI	Internationalized Resource Identifier
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
IoP	Internet of People

IoT	Internet of Things
MCO	Media Contract Ontology
MPEG	Moving Pictures Expert Group
NFT	Non Fungible Token
OSN	Online Social Network
OSP	Online Service Provider
P2P	Peer-to-Peer
PDS	Personal Data Space
PET	Privacy-Enhancing Technology
PIMS	Personal Information Management System
PoW	Proof-of-Work
RDF	Resource Description Framework
REL	Rights Expression Language
SCM	Smart Contract for Media
SSI	Self-Sovereign Identity
TPRE	Treshold Proxy Re-Encryption
UML	Unified Model Language
URI	Universal Resource Idendifier
USA	United States of America
VC	Verifiable Credential

Part I

INTRODUCTION

Chapter 1

Introduction

The issue of personal data protection is currently one of the most discussed and controversial, given the proliferation of data abuse and leakage scandals involving the world's largest companies. The transformation introduced by digital Information Communication Technologies (ICTs) has had and is having a significant impact on the economy and society. The economics of exploiting personal information is assisted by the more pervasive nature of today's digital world: data are at the center of this transformation, and individuals are the primary sources of information and the ones most affected by these. However, current platform-focused data management threatens individuals' control over their personal information, which is proactively collected and stored by many different services and firms. This information enables organizations to provide personalized or more valuable services in digital and physical spaces. Still, it could also have potentially damaging consequences for the privacy and autonomy of users and society at large. Individuals are the everyday users of applications and services provided by Online Service Providers (OSPs), Online Social Networks (OSNs), or firms operating in the vast world of the new ICTs. Everyday Internet services are leveraged by their users to provide them with real-time information based on data sensing or tracking. Smartphones and other personal devices act as gateways, being their users' proxies in the digital world.

However, these proxies usually do not provide appropriate direct or indirect control for individuals to exercise over their data. It is conditioned by the centralized platform-based personal information management techniques, which are then concentrated in a few OSPs to explore, filter, and obtain data of interest (Pariser, 2011). The lack of more direct control by interested parties over their data is of increasing concern. As a result, enacting targeted regulations is critical, and their effects impact the digital lives of many individuals and the operations of many companies. The General Data Protection Regulation (GDPR) (European Parliament, 2016) is the principal example, designed for European Union (EU) citizens to help promote a view in favor of the interests of individuals instead of large corporations. The GDPR conveys data control by imposing several accountability measures on the responsible actors and assigning a set of rights to individuals, i.e., "natural persons should have control of their own personal data" (Recital 7). Even if the GDPR had and is having a significant impact on the protection of personal data, at a large scale, dedicated technologies can help companies comply with GDPR (and similar regulations) while enabling individuals to *fully* exercise their rights. This aspect mainly addresses two main issues: the lack of transparency towards individuals about managing their personal information and their often hindered ability to access and make interoperable personal data. From the technological point of view, there is a need to place (again) individuals at the center and to relieve the absence of technical instruments and standards that make the exercise of one's rights simple and not excessively burdensome.

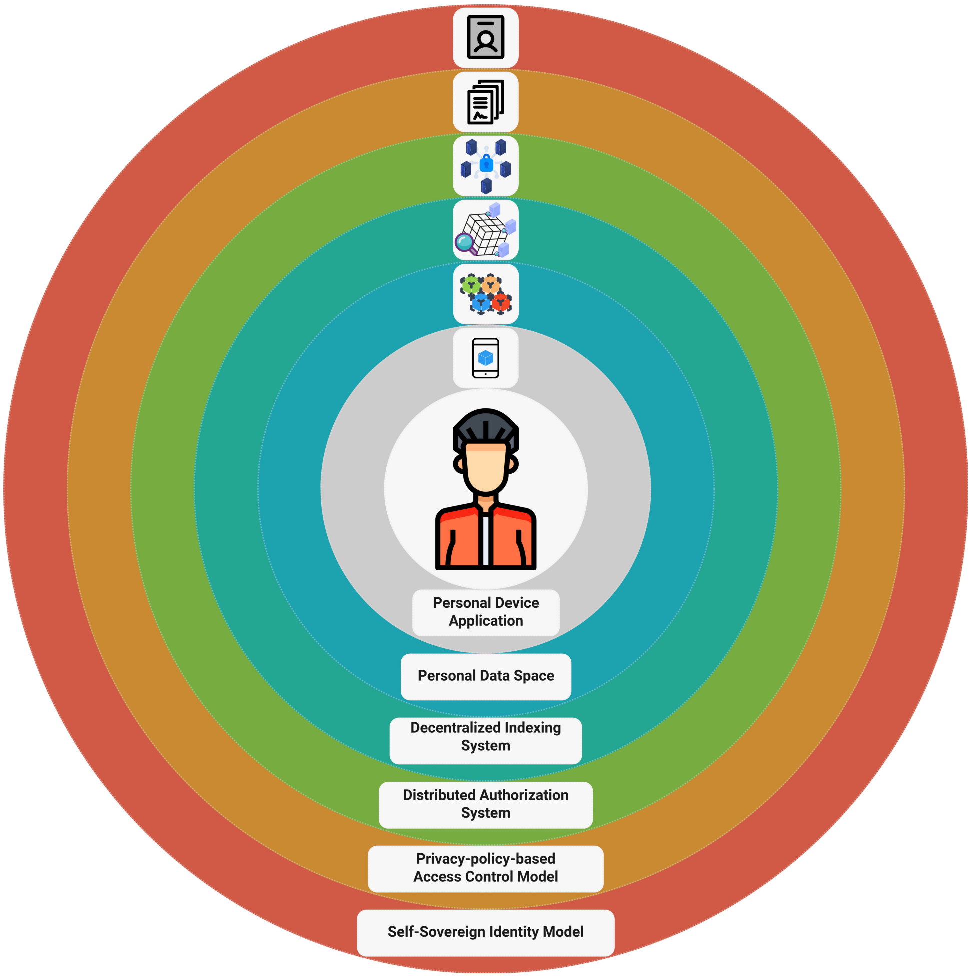
The first step toward these two aims might be to rely on the use of user-centered models for managing personal data, where storage is decoupled from the data management logic and the latter is pushed towards the individual, i.e., a personal information management system (PIMS) (EDPB, 2016; ENISA, 2021). PIMS provides technology-backed mechanisms for individuals to mediate, monitor, and control how their data is accessed, used, or shared, with the purported goal of empowering individuals concerning their data (Janssen and Singh, 2022b). This vision is not only beneficial for the privacy needs of the individual but also for building a single data market that capitalizes on the data interoperability in data spaces for the social good and the development of data markets (European Commission, 2020; Janssen et al., 2020). A new paradigm may arise by ensuring that the user is at the center of data management models, similar to the paradigm involving devices that communicate among themselves in the Internet of Things (IoT). The Internet of People (IoP) is a logical continuation at a higher level of abstraction than what we comprehend the Internet to be, where the individual and his or her relationships with other individuals, ICTs systems, and the world around them constitute the essence of the network itself. The IoP paradigm leverages the new ICTs devices and services to place their users at the heart of the data management design (Conti and Passarella, 2018). Ultimately, the IoP paradigm can help individuals regain their digital sense of ease and foster the development of a new "digital identity" that is truly representative of individuals' intentions.

Self-Sovereign Identity (SSI) is a rising paradigm that brings forward the concept that the user must have complete control over his or her data, being able to store them and decide how much, which, and with whom to share them (Giannopoulou and Wang, 2021). A person's identity does not correspond to a universal concept, but generally, it follows the definitions of different legal, technological or institutional contexts, especially when it comes to digital identity. The continuous flow of subjective views, personal tastes, private details, and intimate experiences consists of data points contributing to the online construction of one's identity. This digital identity, therefore, must be preserved because it will affect the individual, both online and offline, i.e., *onlife* (Floridi, 2014). SSI stems from the evolution of identity management systems, from centralized to federated to user-centered, and prioritizes user autonomy through ten fundamental principles: existence, control, access, transparency, persistence, portability, interoperability, consent, minimization, and protection (Christopher Allen, 2016). A candidate solution that can serve as a basis for developing such SSI is the use of decentralized systems (European Parliament, 2017; Giannopoulou, 2020). In systems theory, a system is decentralized when lower-level components operate on local information to accomplish global goals. Such a system operates through the emergent behavior of its component parts rather than as a result of the influence of a centralized part (Wikipedia community, 2022). In a more technical definition, such systems are decentralized, meaning that their architecture is such that it tries to avoid single points of failure. In the context of SSI, by promoting distributed storage and decentralized computing networks, the ability of the centralized strongholds to use, control, and transfer users' personal information could be shifted towards those that are directly concerned, i.e., the data subjects. Each user of such systems can be associated with a digital space containing personal data that will be used to attend to the data access requests coming from data providers and data consumers. Bringing together regulations and decentralized architectures benefits individuals with the ability to record their data in some interoperable personal data spaces (PDS) (European Commission, 2020), guarantees a path towards data sovereignty, and enables users to control what personal data they want to share (European Parliament, 2017; Giannopoulou, 2020).

PIMS and PDS can be built through decentralized systems and, possibly, through transparent data management offered by OSPs and OSNs, enabling them to demonstrate their compliance easily. The use of Distributed Ledger Technologies (DLTs) and Decentralized File Storage (DFS) in implementing decentralized systems is of paramount importance in this case. DLTs provide the technological guarantees for trusted data management and sharing, as they can offer a fully auditable decentralized access control policy management and evaluation (Maesa et al., 2019). In view of the GDPR, this makes it possible to check whether the involved actors comply with the regulation or not. DLTs embody the principles of control, security, and transparency that enable users to identify themselves while controlling what personal data they want to share (European Parliament, 2017). This kind of technology enables the decentralized execution of immutable instructions, i.e., smart contracts, that can prevent unauthorized access to personal data and ensure compliance with agreed contractual terms for making data available. As concerns DFS, its combined use with DLT allows for overcoming the typical scalability and privacy issues of the latter while maintaining the benefits of decentralization (Politou et al., 2020). In practice, DFS can be leveraged for storing personal data outside the DLT, i.e., by means of "off-chain" storage.

1.1 Structure of the document

The structure of this dissertation follows an incremental approach to describing a set of decentralized systems and models that revolves around personal data and their subjects. Figure 1.1 provides an overall view of the layered conceptual model for the thesis structure. Each layer represents a specific set of systems and models that enable the enactment of a particular behavior for data storing and sharing. They are all centered around the data subjects (or holders) of such systems, and each outer layer depends on its inner layers. Each chapter of the thesis will be focused around a layer and will discuss its technical implementation and relation with the correspondent regulations (except for the personal device application layer, which is discussed throughout all chapters).


Personal Device Application
Personal Data Space
Decentralized Indexing System
Distributed Authorization System
Privacy-policy-based Access Control Model
Self-Sovereign Identity Model

Personal Device Application is transversal to all the other layers; thus, it can be considered as a vertical component. However, it is still the first layer that the final user interacts with. In the document, the personal device application (described in Chapter 4), the Personal Data Space (described in Chapter 4), the decentralized indexing system (described in Chapter 5), the distributed authorization system (described in Chapter 6) are all components or sub-systems of the decentralized PIMS. The privacy-policy-based access control model (described in Chapter 7) and Self-Sovereign Identity model (described in Chapter 8) are not directly "comparable" to the other components, as they are not vital PIMS' components and cannot work as standalone systems.

Figure 1.1: The layers representing the structure of this work and the relation between the proposed decentralized systems and models.

Part I: INTRODUCTION

Chapters 2 and 3 describe state-of-the-art related to current personal data exploitation practices and the research methodology of this work.

Part II: DECENTRALIZED PERSONAL INFORMATION MANAGEMENT SYSTEM

Chapter 4 describes the interdisciplinary analysis of technical and non-technical drivers for the design of a PDS and its implementation based on a DFS.

Chapter 5 describes the decentralized indexing architecture and design of a hypercube Distributed Hash Table on top of DLTs and/or DFS for the execution of keyword-based queries.

Chapter 6 describes the final PIMS based on a multi-DLT GDPR-compliant design, including a distributed personal data access authorization mechanism through smart contracts.

Part III: PRIVACY POLICY, SELF-SOVEREIGN IDENTITY & USE CASES

Chapter 7 describes the design of a privacy-policy-based access control layer to place on top of the smart contract distributed authorization.

Chapter 8 describes the design of an SSI model for verifying the authenticity of some claims in digital interactions, where information about an entity, be it a physical or digital object or an identity or a contextual document, has to be shared with third parties.

1.2 Publications

1.2.1 Journal contributions

- M. Zichichi, S. Ferretti, and G. D'Angelo, *"A Framework Based on Distributed Ledger Technologies for Data Management and Services in Intelligent Transportation Systems,"* IEEE Access, pp. 100384–100402. IEEE, 2020.

- M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, *"Complex Queries Over Decentralised Systems for Geodata Retrieval,"* IET Networks, pp. 1–16. Wiley, 2022.

- M. Zichichi, S. Ferretti, G. D'Angelo, and V. Rodríguez-Doncel, *"Data Governance through a Multi-DLT Architecture in View of the GDPR,"* Cluster Computing, pp. 1–28. Springer Nature, 2022.

- M. Zichichi, S. Ferretti, and V. Rodríguez-Doncel, *"Decentralized Personal Data Marketplaces : How Participation in a DAO can Support the Production of Citizen-Generated Data,"* Sensors, pp. 1–31. MDPI, 2022.

1.2.2 International Standard contributions

- ISO/IEC JTC 1/SC 29 Technical Committee, *ISO/IEC 21000-23 Information technology — Multimedia framework (MPEG-21) — Part 23: Smart Contracts for Media.* 2022.

1.2.3 Book Chapters

- M. Zichichi, S. Ferretti, and G. D'Angelo, Handbook on Blockchain, *ch. Blockchain-based Data Management for Smart Transportation*, pp. 1–29. Springer Nature, 2022.

1.2.4 Conference Proceedings

- M. Zichichi, S. Ferretti, and G. D'Angelo, *"A Distributed Ledger Based Infrastructure for Smart Transportation System and Social Good,"* in Proc. of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), pp. 1–6, IEEE, 2020.

- M. Zichichi, S. Ferretti, and G. D'Angelo, *"On the Efficiency of Decentralized File Storage for Personal Information Management Systems,"* in Proc. of the 25th IEEE Symposium on Computers and Communications (ISCC), pp. 1–6, IEEE, 2020.

- M. Zichichi, S. Ferretti, G. D'Angelo, and V. Rodríguez-Doncel, *"Personal Data Access Control through Distributed Authorization,"* in Proc. of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), pp. 1–4, IEEE, 2020.

- M. Zichichi, S. Ferretti, and G. D'Angelo, *"Movo : a dApp for DLT-based Smart Mobility,"* in Proc. of the 30th IEEE International Conference on Computer Communications and Networks (ICCCN), pp. 1–6, IEEE, 2021.

- M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, *"Towards Decentralized Complex Queries over Distributed Ledgers : a Data Marketplace Use-case,"* in Proc. of the 30th IEEE International Conference on Computer Communications and Networks (ICCCN), pp. 1–6, IEEE, 2021.

- L. Serena, M. Zichichi, G. D'Angelo, and S. Ferretti, *"Simulation of Dissemination Strategies on Temporal Networks,"* in Proc. of the 2021 Annual Modeling and Simulation Conference (ANNSIM), pp. 1–12, Society for Modeling and Simulation International (SCS), 2021.

- M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, *"Governing Decentralized Complex Queries through a DAO,"* in Proc. of the Conference on Information Technology for Social Good (GoodIT), pp. 1–6, ACM, 2021.

- L. Serena, M. Zichichi, G. D'Angelo, and S. Ferretti, *"Simulation of Hybrid Edge Computing Architectures,"* in Proc. of the 2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), pp. 1–8, IEEE, 2021.

- L. Yu, M. Zichichi, R. Markovich, and A. Najjar, *"Argumentation in Trust Services Within a Blockchain Environment,"* in Proc. of the 33rd Benelux Conference on Artificial Intelligence (BNAIC) and the 30th Belgian Dutch Conference on Machine Learning (BENELEARN), pp. 1–21, 2021.

- L. Yu, M. Zichichi, R. Markovich, and A. Najjar, *"Enhancing Trust in Trust Services : Towards an Intelligent Human-input-based Blockchain Oracle (IHiBO),"* in Proc. of the 55th Hawaii International Conference on System Sciences (HICSS), pp. 1–10, 2022.

- L. Yu, M. Zichichi, R. Markovich, and A. Najjar, *"Intelligent Human-input-based Blockchain Oracle (IHiBO),"* in Proc. of the 14th International Conference on Agents and Artificial Intelligence (ICAART), pp. 1–12, SCITEPRESS, 2022.

- N. Pocher and M. Zichichi, *"Towards CBDC-based Machine-to-Machine Payments in Consumer IoT,"* in Proc. of the 37th ACM/SIGAPP Symposium on Applied Computing (SAC), pp. 1–8, 2022.

- M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, *"Incentivized Data Mules Based on State-Channels,"* in Proc. of the 2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 1–2, 2022.

- A. Li, L. Serena, M. Zichichi, S.-K. Tang, G. D'Angelo, and S. Ferretti, *"Modelling of the Internet Computer Protocol Architecture : the Next Generation Blockchain,"* in Proc. of the 4th International Congress on Blockchain and Applications (Blockchain 22), pp. 1–10, 2022.

- L. Serena, M. Zichichi, G. D'Angelo, and S. Ferretti, *"On the Modeling of P2P Systems as Temporal Networks : a Case Study with Data Streaming,"* in Proc. of the 2022 Annual Modeling and Simulation Conference (ANNSIM), pp. 1–12, Society for Modeling and Simulation International (SCS), 2022.

- M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, *"DLT-based Data Mules for Smart Territories,"* in Proc. of the 31st International Conference on Computer Communications and Networks (ICCCN 2022), pp. 1–7, 2022.

- G. Bigini, M. Zichichi, E. Lattanzi, S. Ferretti, and G. D'Angelo, *"Decentralized Health Data Distribution : a DLT-based Architecture for Data Protection,"* in Proc. of the 5th IEEE International Conference on Blockchain (IEEE Blockchain 2022), pp. 1–7, 2022.

- S. Casale-Brunet, M. Zichichi, L. Hutchinson, M. Mattavelli, and S. Ferretti, *"The Impact of NFT Profile Pictures within Social Network Communities,"* in Proc. of the Conference on Information Technology for Social Good (GoodIT), pp. 1–11, ACM, 2022.

- L. Serena, A. Li, M. Zichichi, G. D'Angelo, S. Ferretti, and S.-K. Tang, *"Simulation of the Internet Computer Protocol : the Next Generation Multi-blockchain Architecture,"* in Proc. of the 2022 IEEE/ACM 26th International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2022), pp. 1–8, IEEE, 2022

- M. Rossini, M. Zichichi, and S. Ferretti, *"Smart Contracts Vulnerability Classification Through Deep Learning,"* in Proc. of the 20th ACM Conference on Embedded Networked Sensor Systems (SenSys 2022), pp. 1–2, ACM, 2022.

- G. Bigini, M. Zichichi, E. Lattanzi, S. Ferretti, and G. D'Angelo, *"On the Decentralization of Health Systems for Data Availability: a DLT-based Architecture,"* in Proc. of the 2023 IEEE 20th Annual Consumer Communications & Networking Conference (CCNC), pp. 1–6, IEEE, 2023.

1.2.5 Workshop Proceedings

- M. Zichichi, V. Rodríguez-Doncel, and S. Ferretti, *"The Use of Decentralized and Semantic Web Technologies for Personal Data Protection and Interoperability,"* in AI Approaches to the Complexity of Legal Systems XI-XII, pp. 328–335, Springer International Publishing, 2021. GDPR Compliance - Theories, Techniques, Tools, co-located with the 32nd International Conference on Legal Knowledge and Information Systems (JURIX), December 2019, Madrid Spain.

- M. Zichichi, *"Location Privacy and Inference in Online Social Networks,"* in Proc. of the Seventh JURIX 2019 Doctoral Consortium, co-located with the 32nd International Conference on Legal Knowledge and Information Systems (JURIX), no. 2598 in CEUR Workshop Proceedings, (Aachen), pp. 1–10, 2019.

- M. Zichichi, M. Contu, S. Ferretti, and V. Rodríguez-Doncel, *"Ensuring Personal Data Anonymity in Data Marketplaces through Sensing-as-a-service and Distributed Ledger,"* in Proc. of the 3rd Distributed Ledger Technology Workshop, co-located with ITASEC 2020, no. 2580 in CEUR Workshop Proceedings, (Aachen), pp. 1–16, 2020.

- M. Zichichi, S. Ferretti, and G. D'Angelo, *"Are Distributed Ledger Technologies Ready for Intelligent Transportation Systems?,"* in Proc. of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock), co-located with the 26th Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 1–6, ACM, 2020.

- B. Distefano, N. Pocher, and M. Zichichi, *"MOATcoin : Exploring Challenges and Legal Implications of Smart Contracts Through a Gamelike DApp Experiment,"* in Proc. of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock), co-located with the 26th Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 1–6, ACM, 2020.

1.2.6 Submitted contributions

- (SUBMITTED) M. Zichichi, and V. Rodríguez-Doncel, *"Encoding of Media Value Chain Processes Through Blockchains and MPEG-21 Smart Contracts for Media,"* IEEE Multimedia, pp. 1-11. IEEE, 2022.

- (SUBMITTED) N. Pocher, M. Zichichi, F. Merizzi, M.Z. Shafiq, and S. Ferretti, *"Detecting Anomalous Cryptocurrency Transactions: an AML/CFT Application of Machine Learning-based Forensics,"* Electronic Markets, pp. 1-32. Springer Nature, 2022.

- (SUBMITTED) M. Zichichi, G. D'Angelo, S. Ferretti, and M. Marzolla, *"Accountable Clouds through Blockchain,"* Journal of Network and Computer Applications, pp. 1-39. Elsevier, 2022.

- (SUBMITTED) M. Zichichi, L.Serena, S. Ferretti, and G. D'Angelo, *"InDaMul: Incentivized Data Mules for Opportunistic Networking Through Smart Contracts and Decentralized Systems,"* Distributed Ledger Technologies: Research and Practice, pp. 1-25. ACM, 2022.

1.3 Scholarships

This work has been supported by Marie Sklodowska-Curie Actions – Innovative Training Networks European Joint Doctorate with grant agreement No. 814177 Law Science and Technology Joint Doctorate - Rights of the Internet of Everything.

1.4 Research stays

01.11.19 – 30.04.20 (6 months)	Research stay at the **University of Bologna, Bologna, Italy**.
01.05.20 – 31.10.20 (6 months)	Research stay at the **University of Turin, Turin, Italy**.

01.11.20 – 30.04.21 (6 months) Traineeship at **Bitnomos S.r.l., Bologna, Italy**.

1.5 Software

The software produced during the development of this work is indexed in Github Zichichi (2022b) and stored in the following repositories.

- M. Zichichi (2019). Data and scripts for IOTA vehicular scenario. DOI: `10.5281/zenodo.4572578`
- M. Zichichi (2020). Data and scripts for IPFS and Sia vehicular scenario. DOI: `10.5281/zenodo.3552198`
- C. Giansante, and M. Zichichi (2021). Hypercube DHT Simulation. DOI: `10.5281/zenodo.6548266`
- F. La Piana, A. Leurini, D. Tropea, and M. Zichichi (2021). Hypercube DHT implementation ans vehicular scenario. DOI: `10.5281/zenodo.5810396`
- M. Zichichi, and J. Sparber (2021). Personal data decentralized access control tests. DOI: `10.5281/zenodo.4572552`
- M. Zichichi (2021). Rust implementation of the umbral threshold proxy re-encryption scheme. DOI: `10.5281/zenodo.6548260`
- M. Zichichi (2022). Implementation of a dao that governs the exchange of data with an aggregator. DOI: `10.5281/zenodo.6548262`
- M. Zichichi (2021). Privacy Policy Generator from Smart Contracts. `github.com/miker83z/desp3d-policy-mco-generator`
- M. Zichichi (2021). Privacy Policy Parser for Smart Contracts. `github.com/miker83z/desp3d-policy-mco-parser`
- M. Zichichi (2022). Privacy Policy Smart Contract Templates. `github.com/miker83z/desp3d-policy-web3-templates`
- M. Zichichi (2022). Privacy-policy DLT manager: Manage policies based on MCO and DPV ontologies. DOI: `10.5281/zenodo.7132775`

- M. Zichichi (2022). Client tools to interact with Intelligible suite and Privacy Policy software. `github.com/miker83z/desp3d-client-tools`

- M. Zichichi (2021). Intelligible Decentralized Identity implementation. `github.com/miker83z/desp3d-intelligible-identity`

- M. Zichichi (2021). Intelligible Verifiable Certificate implementation. `github.com/miker83z/desp3d-intelligible-certificate`

- M. Zichichi (2022). Intelligible packages based on DID and VC. DOI: `10.5281/zenodo.7132777`

Chapter 2

State of the Art

2.1 Why should the current personal data protection and portability paradigm be changed?

The ideological and technological foundations of Web 2.0 have been exploited by social media and, in general, by online services to build platforms that enable the creation and exchange of users' generated content (Kaplan and Haenlein, 2010). Primarily by leveraging mobile personal devices, online services provide users the foundation for information dissemination, content generation, and interactive communications of the modern era. Information Communication Technologies (ICTs) and their ubiquitous computing are modifying our world by creating new realities and promoting an informational interpretation of our lives (Floridi, 2014). An example is Web 2.0, which at the start of the new millennium has favored a process that broke the boundaries between Internet consumption and participation: the users of the Web produce the data that other users consume. This great innovation has led to reduced friction in disseminating information online, with great benefits for the entire world population. However, this significant influence on informational friction brings with it great concerns about the privacy of the users who inhabit the online/offline (*onlife*) world (Floridi, 2014). Internet users act not just as content consumers but mainly as content creators. It implies that the content they share often consists of highly personal data belonging to them or their family, friends, and colleagues. Location, interests, their general behavior are all data points derived from their textual data (comments, posts), actions (sharing, reactions, likes), social network topology (friendships, following system), hyperlinks, or metadata.

It is a revolutionary fact that “new” ICTs, such as OSN platforms, do not produce most of the data they handle by themselves, as the “old” ICTs, such as broadcast-based traditional and industrial media, usually do. This kind of data concerns ICTs users’ static personal attributes and, due to the spread of mobile devices, also dynamic information extracted by their activities (Altshuler et al., 2012). Smartphones are the primary source of this information since, through their mobile computation, set of sensors, and Internet connectivity, they can measure several aspects of an individual’s physical environment. Hence the birth of a digital world, created upon peoples’ actions, interests, and desires given in input in the form of data to firms that operate ICTs on a large scale. There is a large number of vendors in the digital marketing industry whose only purpose is to collect ICTs users’ data and transform it into actionable information, i.e., create detailed profiles and user segments for prediction, attribution, and insights. Raw users’ generated data is accessed and transformed by data aggregators and brokers, processed to obtain more sophisticated forms, referenced by analytics vendors, and sold to third parties (e.g., retailers, market researchers, brands) for prediction, attribution, and insights (Acquisti et al., 2016; Banerjee, 2019).

In the following subsections, we will go into the details of the protection and portability of personal data and the privacy threats arising from the current uses of new ICTs.

2.1.1 How a piece of information can be influential to your privacy

First of all, let us clarify that throughout this work, the reference to privacy will be mainly associated with the concept of informational privacy.

> Informational privacy is an individual’s freedom from informational interference or intrusion achieved by a restriction on facts about him or her that are unknown or unknowable.

At the basis of this vision of Floridi (2014), we find the vision of Westin (1967), stating that privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others. Generally speaking, the privacy threat associated with Web-2.0-based services is that, although many users have some information they keep private, they are not aware that a significant part of information about them is generated from other information sources (Acquisti et al., 2016; Forbrukerrådet, 2020; Kamleitner and Mitchell, 2019). It makes each individual less free from informational interference. It creates a lack of control to determine information about them is (possibly) communicated to others.

The reason for this lies in the foundations of the current ICTs structure. The exploitation, i.e., economics, of personal data is helped by the more pervasive nature of today's digital world. When fundamental aspects of one's life are recreated online, his or her "digital twin" can be depicted not only by using his or her information but also by others thanks to social networks (Forbrukerrådet, 2020). Thus, it becomes easier to understand one's activity choice and lifestyle patterns (Hasan et al., 2016) and then to make intrusive recommendations using this information (Bothorel et al., 2018; Partridge and Price, 2009). In practice, informational interference techniques can reduce some data protection mechanisms to render these almost ineffective. For instance, adding "side information", even with a small amount of background data, most anonymous or pseudo-anonymous datasets regarding users' online platforms interaction can be de-anonymized (Ma et al., 2010). De Montjoye et al. (2013) provide a study that shows how just four approximate location data points are sufficient to identify an individual in 95% of cases. When personal data are enriched with Point of Interest, people's activities can be inferred (He et al., 2019). Home and work location information are usually the first (and the easiest) to be inferred (Pontes et al., 2012). Then, simply by knowing these two locations, it is possible to recognize one's activity patterns through his or her peers (Phithakkitnukoon et al., 2010) or his or her friends (Cho et al., 2011). When social media site information comes along, it can only get better (or worse, depending on the point of view). Qian et al. (2016) use knowledge graphs to combine background knowledge and anonymous OSNs data to identify individuals and discover their personal attributes. OSNs often track and collect the location of their users when providing their services. This monitoring can continue even when users are not logged in or have never used those services. Sadilek et al. (2012) show how to infer social links, i.e., friendship in OSNs, considering the patterns in link formation, the content of users' messages, and their location. Bonneau et al. (2009) demonstrate that eight public social links are enough to infer the entirety of your social circle. Jurgens (2013) shows how, by exploiting only a small number of initial locations, it is possible to infer fine-grained users' location even when they keep their location data private, but their friends do not. Indeed, it is not enough for an individual to fully protect his or her activity information if it is possible to obtain “co-location" information from his or her friends (Olteanu et al., 2014). Co-location may consist of data, e.g., a friend’s picture or message posted on the OSNs (Ajao et al., 2015), and metadata, e.g., two users connecting to the OSNs with the same IP address or spatiotemporal correlations in OSNs streams (Yamaguchi et al., 2014).

Sensitive personal data, such as location data combined with other data (Keßler and McKenzie, 2018), are substantially different from the rest of personal data. The ability to track individuals’ locations and movements and to combine this data with other metadata and background knowledge allows first and third-party companies to make inferences such as, for instance, visiting a church weekly (i.e., religious affinity) or attending climate strikes(i.e., political views). Data protection thus becomes crucial, as it concerns the vast majority of the population, who are often unaware of how the underlying ICTs work and how most sensitive information can be deduced simply by using other information obtained from them.

The above inference techniques demonstrate that the “nothing to hide” approach to privacy, often raised by some people, is fundamentally flawed for many reasons: the main one is that everyone has information they want to keep private. However, many do not know that such information can be deduced from other data sources generated from someone else (Kamleitner and Mitchell, 2019). Citing Floridi (2014) in his work on how ICTs affect our sense of self and interaction with the world, he defines ourselves as informational organisms, mutually connected and embedded in an informational environment, i.e., the *infosphere*. The informational organism, i.e., *inforg*, is a set of points obtained by interacting with other organisms that are natural agents, e.g., family, friends, strangers, or artificial agents, e.g., the same digital ICTs that gather these data points. In the infosphere, individuals are de-individualized, re-identified as crossing points of many “kinds of”, and then treated like a commodity and sold or bought in the advertising market (Zuboff, 2019). Acquiring personal information enables large firms and organizations operating in the digital world to provide personalized or more valuable services in digital and physical spaces. However, it could also have potentially harmful consequences for the privacy and autonomy of users and society at large. Lack of privacy control, for instance, leads an individual to be thrown into a “filter bubble" that can affect his ability to choose how he wants to live, simply because the companies that build this bubble choose which options he can be aware of (Pariser, 2011). On a social level, this scheme can lead to a deeper polarization and manipulation of society (Cadwalladr and Graham-Harrison, 2018; Christl et al., 2017) and to "geoslavery" in the case of location information (Dobson and Fisher, 2003). After being categorized through personalities, predispositions, and secret desires, each consumer's digital twin is bought and sold on a vast market that operates largely outside his sphere, namely the digital marketing and the adtech industry. All to persuade individuals to buy particular products or to act in a certain way (Forbrukerrådet, 2020).

2.1.2 The legal support to personal data protection and portability

The General Data Protection Regulation (GDPR) was enacted into law in 2016 (European Parliament, 2016) to protect the personal data of European Union (EU) citizens and to allow the free movement of such data within the EU. According to Article 4(1), personal data are "*any information relating to an identified or identifiable natural person; (...) identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the (...) identity of that natural person*". The GDPR builds upon, or better "runs in parallel to", the Privacy and Electronic Communications Directive (ePrivacy Directive) (European Parliament, 2002) that applies to the data protection and privacy in electronic communications networks and services for the EU citizens. The ePrivacy Directive includes language requiring providers to secure the data they carry by taking "appropriate technical and organizational measures to safeguard the security of its services". Generally, it regulates how third parties collect consent to access information stored on individuals' devices. After the 2009 amendment, it explicitly deals with web cookies, requiring the user's consent for processing. The GDPR, on the other hand, conveys control to the data subject, i.e., any natural person identified or identifiable by the kind of data defined above, by imposing several accountability measures on the actor responsible for the data processing and by assigning a set of rights to subjects, i.e., as "natural persons should have control of their own personal data" (Recital 7). The data controller, i.e., the natural or legal person, public authority, agency, or other body which, alone or jointly with others, determines the purposes and means of processing personal data, plays a central role in the interactions between the various interested parties. Indeed, they are being called into action by the data subjects for the exercise of their rights, and they are rendered liable in the event of a violation of the rules by the data processors, i.e., the body which processes personal data on behalf of the controller. Processors have their obligations under the GDPR, although they ultimately report to the data controller. Within this framework, because of increased technological complexities and multiple data-exploiting business practices, it is becoming harder for ICTs users to gain control over their data. Individual control, particularly concerning one's person, has been described as a reflection of fundamental values such as autonomy, privacy, and human dignity. In regards to this, the GDPR sets first some legal obligations about data processing: (i) data must be processed lawfully, fairly, and transparently; (ii) data must be collected for specified, explicit, and legitimate purposes only (purpose limitation); (iii) data must be limited to data required for the entity's defined purposes (data minimization); (iv) data must be accurate and up-to-date. The idea of control over personal data, then, comes to the front in the provisions of six legal bases (Article 6(1)(a)) for data processing:

i *Consent*, the data subject has given consent to the processing of his or her data for one or more specific purposes.

ii *Performance of a contract*, the data processing activity is necessary to enter into or perform a contract with the data subject.

iii *Legal requirement*, the processing activity is necessary for a legal obligation, such as information security, employment, or consumer transaction law.

iv *Vital interest*, the processing activity that could be necessary to save someone's life.

v *Public interest*, the processing activity for a task carried out in the public interest or in the exercise of official authority vested in the controller.

vi *Legitimate interest*, the processing activity of data subjects' data in a way they would reasonably expect and which would have a minimal impact on their privacy.

2.1.2.1 Consent

The most relevant legal basis consists of the data subject's consent to data processing. Consent must relate both to the use being made of the data and to the entity making such use. In order to be valid, it must be freely given, specific, informed, and unambiguous, involving some form of clear affirmative action from the data subject. It means subjects must fully understand that they are giving consent to use their data for some purpose.

- *Freely given and specific*, the controller must not condition access to the service to the user's consent to processing personal data for purposes other than the service provision. The request for consent must be presented in a manner clearly distinguishable from other issues, in a comprehensible and easily accessible form, using clear and simple language.
- *Informed and unambiguous*, information related to consent must be provided to the data subject in a concise, transparent, intelligible and easily accessible form, using clear and plain language, in particular for any information addressed specifically to a child.
- *Explicit*, the controller should obtain verbal or written confirmation about the specific processing

According to the European Data Protection Board (EDPB) and European Data Protection Supervisor (EDPS) (Article 29 Working Party, 2014b): *"opt-in consent would almost always be required* [...] *for tracking and profiling for purposes of direct marketing, behavioral advertisement, location-based advertising or tracking-based digital market research*".

2.1.2.2 Data Subject Rights

The GDPR grants data subjects several rights regarding their personal data, which they can exercise under particular conditions and with certain exceptions. Compliance with the GDPR thus means, among other things, allowing the exercise of these rights.

- *Right to be Informed*, the data controller must provide the data subject with concise and transparent information relating to the processing of his or her personal data for collecting it.

- *Right to Access*, the data subjects have the right to request access to their data in order to ensure that they are processed correctly, e.g., requesting information about the purpose of the processing, the identity of the data recipient, the existence and logic of automatic data processing.

- *Right to Rectification*, the right to rectification obliges the data controller to facilitate the exercise of the data subject's right to correct, rectify, or complete the personal data concerning him or her.

- *Right to be Forgotten*, the data controller has an obligation to erase personal data without undue delay and to inform controllers who are processing the data about the subject's request for erasure.

- *Right to Data Portability*, enables the data subject to receive personal data which have been provided to a controller (in a structured, commonly used, and machine-readable format) in order to be able to transmit it to another controller. Two categories of data are considered in this case:

  i where the processing is based on the consent or it is necessary to perform a contract to which the data subject is a party;

  ii where the data subject provides the data indirectly by using the services or devices that retrieve the data.

  This right is closely related to the data subject access right and is also in line with fostering the Single Digital Market strategy and enhancing competition between EU countries.

- *The Right to Withdraw Consent*, the data controller must facilitate the exercise of the right to withdraw consent, and the withdrawal may be expressed as conditional, timely, or territorially limiting.

- *The Right to Object*, the data subject has the right to object to the processing of personal data concerning him or her when it is carried out in the public interests or the legitimate interest of the data controller or third party at any time on the grounds that relate to his or her particular situation.

- *The Right to Object to Automated Decision-Making*, the data subject may refuse the automated processing of personal data concerning him or her whenever this processing leads to a decision that affects or produces legal effects on him or her.

- *The Right to Restriction of Processing*, the data controller has an obligation to restrict the processing of a subject's data when: (i) the subject requests the verification of the data accuracy; (ii) the data has been unlawfully processed; (iii) to establish, exercise or defend a legal claim; (iv) the subject objected to the controller the processing and the controller considers that its legitimate grounds override the subject's ones.

2.1.2.3 The GDPR impact on businesses

The GDPR has had (and is having) a worldwide impact on establishing how to promote a view in favor of the interests of individuals as opposed to large companies and corporations (Li et al., 2019). For instance, it has been followed by other regulations around the world, such as the California Consumer Privacy Act (California State Legislature, 2020) in the USA.

In economic terms, however, it can be argued that the GDPR affects the options available to firms to collect the data they need for their operations and the resulting ability to achieve economies of scale in data analysis (Gal and Aviv, 2020). Ensuring the lawfulness of data processing, such as obtaining each data subject's explicit and informed consent for all the specific uses of the data pertaining to him or her, is costly, and large and diversified data controllers enjoy an advantage. Moreover, a data controller is liable to the data subject to ensure that her data are used only under his or her rights. Thus, the costs imposed by this requirement may include ongoing monitoring, screening, and auditing of the processing performed by a data receiver.

The declared intention of the GDPR is not to prevent the exploitation of personal data but to ensure that such exploitation is performed in accordance with the data subjects. However, this approach has a direct impact on business activities for (Ziegler et al., 2019): (i) *risk management*, the necessity to better control the risks related to personal data protection and the exposition to GDPR-related sanctions and penalties; (ii) *data subject rights ownership and control*, the design and implement systems with the data subjects at the core of the model; (iii) *purpose consistency*, when the controller wants to substantially extend the use of the collected data, it should collect a complementary consent; (iv) *data transfer to third parties*, firms must map, manage, monitor, and control the way they process and share data; (v) *cross-border transfer*, the requirement to control cross-border data transfers toward non-trusted countries.

2.1.3 Privacy, data protection, and the user control

There is an essential distinction between privacy and data protection that would be limited to the discussion in this work but that has been discussed extensively in other studies (Kokott and Sobotta, 2013; Westin, 1967; Zuboff, 2019). Assigning a value to informational privacy is different from the protection of the actual personal information related to the individual making the assignment. Privacy controls are mainly in the hands of individuals and the system's users. However, privacy also depends on the protection of personal data, which, on the contrary, is primarily the responsibility of the entity controlling the data, i.e., entities operating in the ICTs digital world. From the point of view of the user of an ICT system, being a data subject, it is possible to distinguish it into three types of personal data (Pangrazio and Selwyn, 2019):

i *volunteered*, data that users give to ICTs systems they are using in exchange for an often "free" service and that may be unconsciously disclosed;

ii *observed*, data that ICTs systems extract from their users by monitoring them;

iii *inferred*, data that ICTs systems obtained by processing the last two types created often beyond their users' knowledge.

These types of personal data are moved through three main links along the data value chain: collection, processing, and use of data-generated information and knowledge (Gal and Aviv, 2020). The collection is the extraction of the data and its "datafication", i.e., the recording, aggregation, and organization of information. Processing consists of optimizing, cleaning, parsing, or combining different datasets to organize the data for future extractions and to find correlations. It can transform the raw data into information and can create knowledge. Finally, data use means employing data-based information or knowledge for prediction and decision-making in relevant markets.

How does GDPR enhance ICTs systems users' control over personal data in this data value chain?

The GDPR, indeed, according to the principle of accountability, imposes the obligation to adopt and account for protection measures to the data controller. This one needs to ensure that data is protected and that the level of privacy its users have set is implemented. Therefore, the question that brings up the central issue here is: even if the data controller makes sure to adopt an “adequate” response in proportion to the assessment it has made of the level of risk to the rights and freedoms of the data subject, is the latter able to determine the level of privacy concerning the personal data being protected by the former? The answer to this question requires two layers of analysis: a surface layer and a deeper layer.

2.1.3.1 Privacy at the surface layer

The first layer comprises the interface methods with which users interact to assess privacy levels and determine the level at which they want to set their privacy. Interfaces here consist of the hardware and software tools that inform and make users decide on actions that have direct consequences on data protection and indirect consequences on data privacy. We specifically refer to smartphone apps, browsers, websites, and similar. In the context of the GDPR, consent is the one that is usually leveraged in these cases in order to process collected data. However, the general problem here is that users often do not seem to think about the consequences of providing (or refusing) consent but, instead, consent whenever they are confronted with a request (Custers et al., 2013). Users generally interface with informed consent through privacy notices (e.g., cookie notices) and user control options at the operating system level. However, these are ineffective for users because they are presented in different and inconsistent ways across services and platforms; worse, most are not GDPR compliant (Mehrnezhad, 2020). Many of the current notices implementations offer no meaningful choice to users. For example, in the case of third-party cookies, a more appropriate implementation would require service providers to use consent notices that would effectively result in less than 0.1% of users consenting to their use (Utz et al., 2019). Cookies, in particular, can assume the form of personal data and are essential by themselves because they have become the backbone of a vast market infrastructure based on their ability to transform information about users’ online behavior into data assets (Mellet and Beauvisage, 2020).

In their work, Van Ooijen and Vrabec (2019) identify three stages in consent-based data processing: (i) the information receiving stage, (ii) the approval and primary use stage, and (iii) the secondary data use (reuse) stage.

In the first stage, the threats to users' control are given by the fact that, even if data collectors may provide individuals' information employing a data use policy, these have difficulties in cognitively processing such information. As a result of the rapid development of technology, such policies are becoming more time-consuming and more complex, resulting in increased pressure on the cognitive functioning of individuals. Moreover, this approach fails to address the problem of information complexity, as it needs to explain the real implications of automated decision-making for an individual. What the GDPR guarantees, with the right to explanation, is an *ex-ante* motivation that merely refers to the system's functionality. Icons may be more successful in mitigating informational complexity, but there is a risk that they may worsen the problem of bias in decision-making (Rossi and Palmirani, 2020).

The threats to users' control at the second stage, i.e., the approval and primary use stage, are steered by subtle changes in the context wherein consent is requested, such as system architectures based on default settings. These can unconsciously steer users' behavior in a phenomenon coined "the malleability of privacy preferences" (Acquisti et al., 2016). Consumers generally prefer and choose the option marked as the default when presented with several choice options. GDPR addresses these threats by validating consent only on the presupposition that a data subject has fully understood the consequences of his or her approval. However, this must be implemented in a way that indeed empowers individuals.

Finally, in the third stage identified by Van Ooijen and Vrabec (2019), the threats over users' control are given by the limited scope of the right to access and portability for individuals. The authors foresee the use of electronic data platforms where individuals can manage their own data. This argument is further analyzed in the following Sections.

2.1.3.2 Privacy at the deeper layer

The second layer of analysis, which is deeper than the previous one, interests the relationship between a user and the information itself in terms of information complexity and privacy perception. In perceiving privacy when they disclose personal information, users come across a privacy paradox that most of the time is not in their favor: while the attitude of users is to profess their need for privacy, in their behavior, most of them remain consumers of the same technologies that collect their personal data (Norberg et al., 2007). Two resolutions can be attributed to this: firstly, the fact that attitudes (e.g., the attitude of practicing high privacy awareness) are usually expressed generically, while behaviors (e.g., the actual data disclosure act) are more specific and contextual (Fishbein and Ajzen, 1977); secondly, users engage in a mental trade-off between privacy concerns and disclosure benefits, performing a "privacy calculus" (Laufer and Wolfe, 1977). When consumers are asked to provide personal information to companies, they will disclose their information based on a decision made after a risk-benefit analysis, i.e., the privacy calculus, analogously to estimating the perceived value. Xu et al. (2011) define this perceived value of information disclosure as the individual's overall assessment of the utility of information disclosure based on perceptions of privacy risks incurred and benefits received. However, two main challenges hinder the correct estimation of this value. First, there is a problem of information overload that stands in the way of a correct estimation in the privacy calculus. This is because users need to consider all the information that is made available in the collector's data use policies, together with the vast amount of information spread across different devices, media, and services. This richness of information threatens the ability and motivation of individuals to examine the critical details needed to make informed privacy decisions (Van Ooijen and Vrabec, 2019). Secondly, a problem of information complexity arises (Acquisti et al., 2016).

Most ITCs users need to learn the sophistication of how they can be tracked or to be aware of possible alternative solutions to their privacy concerns, e.g., the use of privacy-enhancing technologies (PETs).
It results in information asymmetries between the user and the data collector.

Taking as an example a specific kind of sensitive personal data, i.e., location data, the perception of location privacy falls under the same assumptions of the privacy calculus. In particular, the determination of location privacy can assume a numerical quantity, as shown by Shokri et al. (2011), based on the idea that users' privacy and the success of an "adversary" in his location-inference attacks are two sides of the same coin. The authors quantify location privacy as the error of the adversary in estimating the actual user's location information (given an attack model of reference). In a more consumer-oriented definition, location privacy consists of the user's ability to regulate external audiences' access to information about his or her current or past locations (Banerjee, 2019). This view is in line with Westin's and IAPP's definitions of information privacy, i.e., based on the assumption that "privacy is not the opposite of sharing– rather, it is control over sharing" (Acquisti et al., 2016).

2.1.3.3 A solution for the long (and perhaps also short) run

This Sub-Section builds upon the significant issue related to the informational privacy discussed up until now and serves as a bridge to the following Sections. In the following, we will analyze state of the art for a possible solution to approach personal data protection and portability. The primary analysis and studies in the state-of-the-art focus on describing the so-called "decentralized" or "Web3" systems that can benefit users' privacy. Distributed Ledger Technologies (DLT) no longer need to be introduced. Its ledger, distributed among a network of nodes, and the decentralized protocol eliminate the need for a trusted authority and replace it with a system of publicly verifiable evidence. This technology provides the means for disintermediation as it increases confidence in the functioning of its particular system and indirectly reduces the need for trust in the system (De Filippi et al., 2020). The "Web3" or Web 3.0 comes along trying to exploit the advantages that decentralized system might provide in order to build upon Web 2.0, a version of the Internet in which users are truly sovereign over their data and actions, e.g., by owning the unique piece of information that might enact an operation such as a private key. From the perspective of individuals, these technologies help to move computing applications, data, and services towards the edge of the IoP, i.e., closer to them, as personal devices compose the frontiers of such a network of devices. For many scholars, DLTs, combined with decentralized identity mechanisms, could become the necessary building blocks for the decentralized Internet of the future (Kondova and Erbguth, 2020; Lopez and Farooq, 2020; Lopez et al., 2019).

2.2 Why decentralized (permissionless) systems should NOT be used to implement this change?

As we have seen in this Chapter, the general flaw we are trying to address in this work is the centralization of power in a few organizations. This can be due to many factors, such as the spontaneous emergence of hierarchies in natural systems (Bakos et al., 2021) or market dynamics' effect, such as preferential attachment and manifestation power law manifestation (Lopez et al., 2019). Privacy is our main concern here, but this flaw regards many more reasons, such as censorship and anti-competition issues. In general, the main risk is the existence of a single point of failure. In systems theory, a system is decentralized when lower-level components operate on local information to accomplish global goals. Such a system operates through the emergent behavior of its component parts rather than as a result of the influence of a centralized part (Wikipedia community, 2022).

> In a more technical definition, such systems are decentralized, meaning that their architecture is such that it tries to avoid single points of failure.

With the advent of Bitcoin (Nakamoto, 2009) as the first system providing a Peer-to-Peer (P2P) cryptocurrency, there has been a new wave of development of decentralized systems to combat the single point of failure issue. The first widely used systems of this type date back to the start of the new millennium, making it possible to share and exchange resources such as text files, music, and video, e.g., BitTorrent, Emule, Gnutella, and Napster. Generally speaking, P2P systems usually run on top of an already existing network, like the Internet. The underlying overlay network can be represented as a graph where the nodes are the "peers," and the edges connect peers directly communicating. Two prominent aspects related to the functioning of a P2P system are (i) how the overlay network is built and maintained and (ii) how messages are exchanged among peers (Serena et al., 2020). Hybrid P2P systems might employ some servers for coordination, while pure P2P architectures do not rely on any centralized entity (Backx et al., 2002). Bitcoin is a DLT built on top of a pure P2P system, with the primary objective of storing transactions of assets in the form of so-called cryptocurrency. To achieve decentralized verification of each block, the entire DLT is replicated among all nodes forming the P2P system. Each peer is randomly connected to others, and transactions are disseminated across the entire network. Each node then independently verifies the transactions received to ensure their consistency and avoid attacks, e.g., double spending of Bitcoins. Different types of DLTs provide different implementations of the ledger that store transactions; however, in blockchains, transactions are collected in blocks, and each block contains the hash of the previous one. Nakamoto revolutionized the decentralized systems' world (and even finance) by introducing a consensus mechanism to ensure all the copies of the ledger are the same for all peers (Nakamoto, 2009). All the other peers accept a block only if it "solves a puzzle". To make it solving this "crypto-puzzle," it is required intensive computation work that consumes time, i.e., at each cycle try and attach a different nonce (i.e., a random number) to the block until the execution of a hash function on the block together with the nonce returns a string prefixed with X zeros. This is a general description of Bitcoin's Proof-of-Work (PoW) consensus mechanisms (also called mining). Moreover, the X is changed dynamically to make this crypto-puzzle more difficult.

Nakamoto's is, in fact, a revolution since many industries and financial sectors have been influenced by it. In addition, it has also brought optimism for incentivizing participation models, resource contribution, and consensus that could provide a substrate for a decentralized Internet, i.e., the Web3. While there are many different types of DLT, each built with fundamentally different design decisions, the overarching value proposition of DLT and blockchains is that they can operate securely without any centralized control. However, this permissionless way of operating a sizeable decentralized system has also brought many criticisms.

Several authors argue that the central problematic aspect of DLTs is their core notion of being trustless (De Filippi et al., 2020; Finocchiaro and Bomprezzi, 2020; Waldo, 2019). In many ways, their attempt to replace trust with code, i.e., the source code that implements the consensus algorithm, makes these DLTs less trustworthy than non-blockchain systems. Indeed, it can be argued that many permissionless DLTs are not truly decentralized, and their inevitable centralization is detrimental because it is essentially emergent and ill-defined (Schneier, 2019, 2022). In such systems, the need for trusted intermediaries is still present, and they often have more power and less oversight than non-blockchain systems. As we will see in the following Chapters, governance and regulations are needed. A critical part of permissionless DLTs is that anyone can participate in the consensus mechanism. Thus, due to its distributed nature, there are no reference points for placing trust. Schneier (2019) argues that non-DLT systems are based on other general mechanisms that humans use to incentivize trustworthy behavior and that make consensus mechanisms unnecessary: morals, reputation, institutions, and security mechanisms. Morals make a person act in a trustworthy way based on decisions taken individually, while reputation is based on the influence of one's social group. Institutions have rules and laws that induce people to behave according to the group norm, while security mechanisms such as door locks are employed to fulfill the gaps of the previous three mechanisms. What DLTs' consensus mechanism does, is to shift some of the trust in people and institutions to trust in the technology (Schneier, 2019). When that trust turns out to be misplaced, there is no recourse. For example, if an individual is the sole holder of a private key used to unlock Bitcoin funds and this one is lost, then there is no remedy. In this case, Scheiner wonders whether it is better to trust a technology or a (or a group of) person(s).

By being open access and fully distributed, a permissionless environment may not be able to incentivize participants to adequately provide functions like quality control or coordination of system development and evolution (Bakos et al., 2021). Centralization emerges de facto because of the investment of expertise, reputation, time, or money of the hierarchy of developers or organizations required to enable open access and decentralized control. The higher the costs, the fewer the people that want to participate (Bakos et al., 2021). Permissionless consensus mechanisms require trust in the various members who execute it, i.e., miners or validators (depending on the consensus algorithm) or in their governance. Verifying that these members are not cheating on the hashing of a block is easy. The more extensive and diverse the group of validators or governance, the less likely anyone is to collude. However, even assuming that the group that computes the blocks is trusted, as is its governance, it is necessary to trust the developers of the software used to manage the blocks, ledgers, and all. For this regard, Trail of Bits (2022) investigates these matters intending to show how a subset of participants can gain centralized control over the entire DLT. Their work is based on 6 measures of centrality:

- **Authoritative centrality** - *What is the minimum number of entities necessary to disrupt the system?* The Nakamoto coefficient represents the reply to such a question, and the closer this value is to one, the more centralized the system. It has been shown that for the most used permissionless DLTs, such as Bitcoin and Ethereum, the Nakamoto coefficient is relatively low. While it might be prohibitively expensive attacks such DLTs for individuals, competing technologies and nation-states might have the requisite resources.

- **Consensus centrality** - *To what extent is the source of consensus centralized?* The Bitcoin PoW consensus mechanism is currently not executed by single computers or machines owned by an individual but rather by so-called mining pools that aggregate several individual computing powers into one for the same objective, i.e., solving the crypto-puzzle. Only the four most popular mining pools constitute over 51% of Bitcoin's computing power. This paves the way for the most impactful attacks possible to the DLT (Ye et al., 2018). Moreover, each mining pool operates its proprietary centralized protocol and interacts with the public Bitcoin network only through a gateway node, making them an easy target for single point of failure attacks.

- **Motivational centrality** - *How are participants disincentivized from acting maliciously?* The possibility of attacks, such as the 51% attack, shows how most Bitcoin nodes are incentivized to behave dishonestly. Kwon et al. (2019) have shown that there is no known way to create a permissionless DLT that is immune to malicious nodes without having a trusted centralized third party.

- **Topological centrality** - *How resistant is the consensus network to disruption?* Through empirical estimates of the degree distribution, the authors show that a dense subnetwork of public nodes in the P2P DLT permissionless network is largely responsible for reaching consensus and communicating with nodes executing the consensus mechanism.

- **Network centrality** - *Are the nodes sufficiently geographically dispersed such that they are uniformly distributed across the Internet?* Permissionless DLTs such as Bitcoin can suffer arbitrarily degradation or denial of services to any node because, in the past 5 years, 60% of all Bitcoin traffic has traversed just three Internet Service Providers. Moreover, these or any third party on the network route between nodes can observe and choose to drop any messages they wish, also when the Tor protocol is employed (Biryukov and Pustogarov, 2015).

- **Software centrality**: *To what extent is the safety of the DLT dependent on the security of the software on which it runs?* Each DLT has a privileged set of entities that can potentially modify past transactions: software developers and maintainers. They represent a centralized point of trust in the system, susceptible to targeted attacks; for example, there are currently four active contributors who have access to modify the Bitcoin Core code base. Software bugs can lead to consensus errors and change the state of the blockchain. In this case, trust in software development is to accept the immutability of the ledger and believe that the programmers have not introduced a bug. A not-so-popular alternative is to update the code off-chain, which shares the same trust issues as a centralized approach.

Moreover, some other problems are inherent to the P2P nature of decentralized, permissionless systems. Lopez et al. (2019) argue that decentralized, permissionless blockchains remain ill-suited. The first problem they analyze is the free-riding and incentives for peers. The issue is that, generally, peers do not donate their computing, storage, and bandwidth resources for altruistic reasons, i.e., without incentives. This led to the creation of mining pools and large mining industries, making it impossible for an average user to mine with an ordinary desktop computer to receive compensation. The second problem depicted by the authors regards the security and fragility of open permissionless networks, as attackers, in this case, have economic incentives to break the system. The third problem is a performance issue due to instability, heterogeneity, and churn in the P2P network. The "fatal" issue of P2P networks is that they show high heterogeneity and high degrees of churn, i.e., nodes that dynamically come and go in the system. In DLTs, this translates into the scalability trilemma: a DLT can only address two of the three scalability, decentralization, and security challenges. The solution to this is the employment of so-called layer-two or off-chain solutions that increase the system's complexity. In fact, the fourth problem taken into consideration is system complexity and maintenance. Programming distributed systems that must be fault-tolerant, self-adjusting, and scalable is challenging.

2.3 Why (and how) decentralized systems should be used to implement this change?

If trust can be a pivotal element in making the promises of (permissionless) decentralized systems vain, at the same time, it can be that element that, if reinterpreted, can tell us why these systems could bring significant benefits in different contexts. Becker and Bodó (2021) define trust as a complex social phenomenon with interrelated individual and systemic aspects, a relational attribute between a social actor and other actors (interpersonal) and/or actors and institutions (institutional) and institutions and actors (shared expectations). The discussion on trust in DLT spans from the technological to the legal-social to the economic context. If we first need to clarify what trust means academically, its relationship with DLTs can be delineated: “does blockchain increase trust, decrease trust, make trust obsolete, or represent a shift in the nature of trust?” (Becker and Bodó, 2021). A DLT can be considered trustless or trust-free, i.e., that takes care of trust issues and frees individuals from the necessity of implementing mechanisms to signal or convey trust (Beck et al., 2016), or not wholly trustless, i.e., replacing interpersonal trust with trust in the DLT itself (miners, consensus mechanisms, nodes), software developers or new intermediaries (cryptocurrency exchanges). De Filippi et al. (2020) define this not wholly trustlessness as confidence, and thus the DLT becomes a “confidence machine” in the sense that it increases the confidence in the operation of a particular system. The DLT, thus, is reduced only indirectly. The authors argue that creating solid expectations about the correct behavior of operations performed by DLTs increases user confidence, thus eliminating the need for a centralized "trusted" authority. Therefore, confidence in DLTs ultimately depends on the proper governance of the system. This means increased confidence is intrinsically related to the degree to which the various actors involved in governance act as expected.

Both trustless and confidence visions mainly apply to the permissionless case, while permissioned DLTs are usually not considered trustless, as they afford one or more organizations in a maintaining role that need to be trusted. Nevertheless, the members of the latter kind of DLT do not necessarily trust each other, as problems such as authorization and auditing are intrinsic to permissioned DLTs. Although not fully decentralized by design, the governance structure of permissioned systems can also ensure some level of decentralization.

2.3.1 Permissioned systems come to the rescue

As seen in the previous Section, in the absence of formal checks for the underlying centralization forces of permissionless systems, centralization emerges in practice. On the other hand, permissioned decentralized systems often operate thanks to contractual agreements between the entities that implement the governance aspect for the system’s operations. Permissioned refers mainly to access to information and governance aspects. In this regard, decentralized systems can be characterized on three key dimensions

(Bakos et al., 2021): (i) architecture, which can be concentrated or distributed; (ii) access, which can be permissionless or permissioned; and (iii) control, which can be centralized or decentralized.

Even if permissioned DLTs are often compared to "not so interesting" distributed databases, when these systems are designed to streamline and convey business relations (that already require trust), they can be more effective than a non-DLT system in the transfer, clearing, and traceability of exogenous assets or rights (Bakos et al., 2021). Permissioned blockchains and the execution of smart contracts on top of them may enable trust, or, better say, confidence, between many players for the validation of contractual obligations (Lopez et al., 2019). Business relationships require trust in the operational and institutional setting, action accountability, and reputation. What permissioned DLT can bring is to achieve higher security at lower levels of trust that any single participant can be induced not to deviate from the protocol (Bakos and Halaburda, 2021). Permissioned DLT entities are identifiable outside the DLT, i.e., off-chain, and are thus subject to penalties imposed by the institutional setting.

The permissioned DLT is then **a unique framework for collaboration in competition scenarios**.

Entities competing at an overall business level can cooperate for other purposes (Bakarich and Castonguay, 2021). Unlike a traditional distributed database, no central entity manages and protects the data. Instead, all "business-competing" permissioned DLT nodes control, maintain, and guard the information posted to it, providing an additional layer of control if one of the parties attempts to alter or change previously agreed-upon information. In this way, the ledger can securely and efficiently create a tamper-proof log of sensitive activity.

The trust placed in off-chain negotiations between two or more entities, whether institutional or shared expectations, can allow for the design of consensus mechanisms where a large number of validator nodes have a say in the validation process (Bakos et al., 2021). Thus, the primary ability that (permissioned) decentralized systems can provide to their users when reinterpreted as a confidence machine is "gaining truth through the ability to share data safely" (Hardjono et al., 2019). Coming back and focusing again on the change in the current paradigm for personal data protection and portability paradigm:

- DLTs can provide a single source of verifiable truth among organizations and some level of appropriate automation of data processing.

- Organizations can arrange a form of governance to decide the distributed sources of trust and to moderate such permissioned systems.

- The authority can be distributed among many trusted actors so that compromise of one or even a few authorities does not destroy the consensus.

- Intrinsic cryptographical properties of DLTs can enable distributed safe computation and data minimization.

- The networked collaboration environment can be easily exploited for the audit and accountability of operations.

- P2P networks offer a valuable solution for data resiliency and scalability.

For Lopez et al. (2019), permissioned systems are of great value when used in environments composed by a collection of players or stakeholders that do not fully trust each other: supply chain & logistics, to trace products while different actors and companies handle them, without having to trust every node in the chain explicitly; healthcare, for the secure sharing of health data across hospitals and platforms; education, for validating of academic credentials and certifications, utilities, such as smart power grids with distributed power generation from both residential and businesses. Even if a centralized system for each case were feasible, such an approach would lack the *flexibility of the evolution and portability of members and services and interoperability with other external DLTs and/or services*.

As an example, the European Blockchain Services Infrastructure (EBSI) (European Blockchain Partnership EBP, 2022) could become a superhub for a myriad of lesser national/local networks: DLT islands will emerge around the world in different sectors (Lopez et al., 2019). The EBSI leverages a permissioned DLT where each EU member state maintains nodes to accelerate the creation of cross-border services for public administrations. The DLT enables the EBSI ecosystem to verify the information and to make services more trustworthy, changing the traditional pattern of data sharing due to its distributed nature. In there, the ledger acts as a point of truth that supports the verification of the entities involved in the transaction and the authenticity of information without requiring real-time access to the source of information.

2.3.2 Personal information management in DLTs

The main benefit of DLTs and related decentralized systems (e.g., Decentralized File Storages (DFS)) is that they provide individuals with functionalities that are not possible in traditional cloud services. In particular, they favor the creation of decentralized Personal Information Management Systems (PIMS) (EDPB, 2016), guaranteeing data sovereignty and enabling users to control what personal data they want to share. This kind of PIMS can be considered a consent management tool, personal data cooperative, or trusts that act as new neutral intermediaries in the personal data economy, built on distributed software architectures. These empower individuals with tools and means to decide at a granular level what is done with their data to provide, between many benefits, greater oversight and transparency over the data. The first step toward this can be using a new user-centered model for managing personal data, where storage is decoupled from application logic, and individuals decide what to do with their data. This model is not only beneficial for the privacy needs of the individual but would also allow data collectors (i.e., OSNs, ISPs, and companies, generally speaking) to prove their compliance with regulations. Not only that, but it could also benefit the creation of a single data market that capitalizes on the data interoperability between data spaces for the social and economic good (European Commission, 2020). Data interoperability is one of the main issues from a practical point of view, tainted by the centralization of personal data management and distribution. The current practice of data collectors is to store data in disconnected silos inaccessible to innovative services, researchers, and often to the individual who generated them. The lack of control by individuals over access to their data, therefore, joins the other privacy concerns. As a result, it is challenging for an individual to understand and manage the associated risks. The GDPR helps to promote a vision in favor of the interests of individuals instead of large corporations, i.e., “natural persons should have control of their own personal data” (Recital 7). However, from the technical side, there is an urgent need to place individuals at the center of personal data management and to relieve the absence of technical instruments and standards that make the exercise of one’s rights simple and not excessively burdensome (European Commission, 2020). For instance, Article 20 of the GDPR has the potential to allow data flow and promote competition, but instead, it translates into the possibility of switching service providers rather than re-using data.

By promoting distributed storage, decentralized computation, and specifically SSI and PIMS, the ability of centralized strongholds to exploit individuals' personal information in their favor could be transferred to those directly concerned users. Decentralized computation comes in the form of smart contracts, i.e., an immutable set of instructions whose execution is calculated deterministically by all (or several, depending on the protocol) peers in the DLT. In the following, we will devise the solutions scholars have provided for the problems discussed here.

- **DLTs as a data management and sharing tool** - Various works in the literature have the main goal of proposing DLTs-based architectures for data management to build novel smart services and to promote social good (Khelifi et al., 2018; Naz et al., 2019; Ortega et al., 2018). These solutions usually store data outside the ledger, i.e., off-chain, and involve the DLT to provide transparency in the process of access to data while simultaneously enabling users to control their data. Focusing on data trading, works such as Özyilmaz et al. (2018); Shafagh et al. (2018, 2017) propose the use of distributed technologies to trade data in IoT and smart cities, as these two scenarios seem to fit perfectly with DLTs.

- **DLTs towards GDPR compliance** - DLTs are fundamentally in contrast to some parts of the GDPR. However, some scholars present DLT-based solutions without compromising GDPR compliance, but on the contrary, trying to best leverage them as tools for data subjects. A GDPR-compliant model for tracing the personal data life cycle is proposed by Onik et al. (2019), while Ahmed et al. (2020) focus on the lack of GDPR-compliant consent management mechanisms. Several works focus mainly on health data, such as the one by Hawig et al. (2019) and by Koscina et al. (2019), that enable healthcare data exchange through a distributed architecture. The latter is part of a larger body of works related to a Horizon 2020 program, i.e., MyHealthMyData. Another Horizon 2020 program related to DLT-based architectures for personal data is the DECODE project which provides compliant tools for individuals to take control of their data and share it.

- **Smart Contract based policies** - Smart contracts are often employed to let users express their privacy policies. One of the tools developed with this focus is Zenroom (Roio, 2019), a language interpreter which reads in a natural language. Smart contracts are used to implement access control mechanisms already tested in the past to convey user consent, e.g., PROV-O and the XACML models (Davari and Bertino, 2019) or delegation models (Ahmad et al., 2017). Legal ontologies can help when translated into smart contracts (Cervone et al., 2020; Choudhury et al., 2018).

- **Non-DLT-based PIMS models** - Some solutions that are not based on the use of DLTs, but that share some features, are the Databox model (Katevas et al., 2020) and Solid (Sambra et al., 2016). The latter is a project that involves the use of distributed technologies and Semantic Web integration.

2.3.2.1 Self-Sovereign Identity

From the self-determination perspective, DLT can play a central role in implementing a personal right to identity. Self-Sovereign Identity (SSI) is currently one of the most compelling use cases of DLTs and consists of a digital identity created and managed by each person individually, without the intervention of third parties. Looking at identity as the outcome of the construction of one's personality, i.e., a set of attributes that draw a profile, brings back the considerations of the previous Sections on the need to protect geodata, metadata, and all other related personal data. Especially when it comes to digital identity, the continuous flow of subjective views, personal tastes, intimate details, and experiences consists of data points that contribute to it, affecting the individual, both online and offline, i.e., *onlife* (Floridi, 2014). SSI brings forward the concept of managing digital identity through DLT. In particular, it advocates the vision that individuals should not only be placed at the center of the digital identity management process, but they must be the rulers of their digital identity (Christopher Allen, 2016). It leads to the result that individuals must have complete control over their data and be able to store them and decide how much, which, and with whom to share them. The thesis of scholars is that DLTs can be used to empower individuals through SSI (Giannopoulou, 2020; Kondova and Erbguth, 2020). SSI stems from the evolution of identity management systems, from centralized to federated to user-centric, and prioritizes user autonomy through ten fundamental principles: existence, control, access, transparency, persistence, portability, interoperability, consent, minimalization, and protection (Christopher Allen, 2016). The SSI paradigm aims to empower the individual through solutions that can be thought complementary to regulations such as the GDPR. From what we have devised up until now, SSI and DLTs can bring to individuals the necessary tools to not only have their data protected but also enhance their correct estimation in the trade-off between privacy concerns and disclosure benefits (i.e., privacy calculus), thus leading to a favorable environment for individuals' privacy.

Only some of the architectures described in previous Sections are based on the SSI principles. Two of the leading solutions we find in the literature leverage public and/or private DLTs to implement SSI principles. The first one is, Sovrin (Sovrin Foundation, 2020), an open-source identity network built on a layered architecture involving a public permissioned DLT that only stores identity transactions, without personal data, and where only trusted institutions, i.e., banks, universities, governments, can be DLT nodes. uPort (Naik and Jenkins, 2020), on the other hand, is based on Ethereum's public permissionless DLT and provides a platform for user-centric identity and communication, consisting of a mobile app, smart contracts, and a set of open protocols for message flow. Finally, in the context of SSI, a European-wide project is gaining momentum for digital identity management and other citizens' services through the EBSI.

On the other hand, some solutions do not involve using DLTs to embody the SSI paradigm. As already cited, Solid (Sambra et al., 2016) is a PIMS where users directly act on their data. The principles are in line with SSI, i.e., users decide what data is stored, where it is stored, and who has access to it, and can also revoke access to their data. The digi.me system (World Data Exchange Company, 2022), on the other hand, focuses on providing a PIMS that first tries to get as many data points as possible from different sources. Users can import their data from different ISPs, encrypt them and store them in a personal cloud (e.g., Google Drive). Individuals can then control these data and allow other apps or services to access them through set limitations.

2.4 The tensions between DLTs and GDPR

When the GDPR was first drafted in 2012, centralized client-service network relationships strongly influenced it. That was when cloud computing was gaining momentum, while peer-to-peer technologies were losing the appeal that made them stand out in the first decade of the 2000s. The fundamental characteristics of DLTs, mainly decentralization, imply an operating environment and paradigm that makes it difficult to interpret some of the GDPR rules. Usually, no central authority determines how and where data is stored, processed, or used in any other way. According to this disintermediation model, it may be challenging to identify which participants in a DLT are the data controllers, processors, or subjects because the information does not necessarily flow linearly from users to providers (Sovrin Foundation, 2020). The tensions between the GDPR and the DLT mainly concern three issues, as stated by scholars working on this specific field:

- **Actor accountability** - As far as the GDPR is concerned, it must be possible to identify a data controller, but it is not clear whether this is possible in permissionless DLTs (Finck, 2019; Lyons et al., 2018). The concept of data controller requires further clarification to define such a decentralized data processing model. The more concentrated the control of the participating nodes on the DLT is, the more it is possible to identify the controllers. In contrast, the more diffuse the control is, the more it is not possible an identification (Sovrin Foundation, 2020).

- **Personal data processing** - So far, there is no significant consensus on what is necessary to render personal data anonymous so that the resulting output can be stored in a DLT. This argument found its basis in Recital 26 of the GDPR, which states that data becomes anonymous if it is 'reasonably likely' that no identification of a natural person can be derived (Agencia Española De Protección De Datos, 2019; Finck and Pallas, 2020).

- **Data subject rights** - Even if a data controller might be identified by their nature, DLTs hinder the implementation of a set of rights for data subjects and obligations for data controllers of the GDPR. Two, in particular, the "right to be forgotten" and the "right to rectification" (Articles 16 and 17), are the main breaking point between the DLT and GDPR due to the ledger immutability.

2.4.1 Principles of a GDPR-compliant DLT-based solution

However, although challenging, DLT-based solutions can be designed to be GDPR-compliant, provided they are developed with appropriate design features to protect personal data. State of the art includes studies by regulators such as CNIL, AEPD, and STOA, but also companies, organizations, and scholars. Best practices for a GDPR-compliant DLT-based design vary from case to case, but here we can derive some major ones:

1. Avoid storing personal data on-chain (or let the subject do it)
2. "Conscious" use of Privacy-Enhancing Technologies
3. Specialized DLT-based system that forgets
4. De-linking DLT-addresses from identities
5. Define accountability obligations among the network participants

2.4.1.1 Avoid storing personal data on-chain

A DLT-based system architecture that intends to be GDPR-compliant needs to be designed to prohibit or prevent personal data from being stored on-chain on a public permissionless DLT or a not "tightly controlled" one (Lyons et al., 2018; Sovrin Foundation, 2020). A use case where the DLT is tightly controlled is when the nodes that store the ledger are known and act as data controllers, and access to this network is permissioned. For this principle, one can refer to the multi-layered blockchain design proposed by the EU Blockchain Observatory and Forum (Lyons et al., 2018). In particular, Principle 3 suggests to save personal data needed to perform a service in a permissioned DLT and non-personal data in a permissionless DLT. This makes it easier to permissioned DLT nodes to agree on contractual terms that precisely define their roles and duties and the privacy policy toward end users. When acting as data controllers, these entities must facilitate the exercise of data subject rights (Articles 12 and 15 to 22) and cannot delegate this task to processors. They can be aided by using off-chain storage, e.g., DFS, and by appropriate data obfuscation, encryption, aggregation, and deletion techniques (discussed in the following Sub-Sections). A specification is needed at this point for data types that are typically managed and stored in DLT-based architectures:

- Hash digests stored on-chain - The result of a hash function, i.e., a digest, should be generally considered pseudonymized data and thus still in the scope of the

GDPR (Recital 26). This is the conclusion that the analysis of the Data Protection Working Party (ex Article 29 Working Party) reached (Article 29 Working Party, 2014a). There are, however, Privacy Enhancing Technologies (PETs) with stronger privacy guarantees that may resist de-anonymization and thus making pseudonymous data be considered with a shallow risk of re-identification (Finck, 2019; Finck and Pallas, 2020). For instance, a proposed approach would be to store Salted Hash Digests with solid privacy guarantees on-chain (possibly in a permissioned DLT) while encrypted data off-chain.

- DLT addresses - The addresses that represent accounts in DLTs derived from the user's public keys are also pseudonymous data. However, their use is essential to the DLT's proper functioning; thus, in the light of the data minimization principle (Article 5(1)(c)), the CNIL considers that it is not possible to minimize them further (CNIL, 2018a).

- Personal data that is submitted by the subject - This data consists of all the possible personal information that the subject may voluntarily submit to the system. This kind of data should be stored (and protected) off-chain, and the actors who handle this data should always be known: the subject or permissioned DLT nodes.

2.4.1.2 "Conscious" use of Privacy-Enhancing Technologies

When a PET enables the possibility not to store data or references on-chain, e.g., Zero-Knowledge Proof (Feige et al., 1988), then what is stored, e.g., cryptographic proofs, is outside the scope of GDPR. Whereas, in the case of minimized data (which still counts as personal data), the hash digest, for instance, "conscious" use of PET could minimize the risks to identification and thus provide sufficient protection for the legitimate interest to rely upon Article 6(1)(f) GDPR (Erbguth, 2019a; Finck and Pallas, 2020; Giannopoulou, 2020). These data protection mechanisms should be implemented in order "not to allow the data subject to be identified via 'all' 'likely' and 'reasonable' means" (Agencia Española De Protección De Datos, 2019; Finck and Pallas, 2020). For example, the conscious use of PET could be the case where a piece of data, such as a subject's age, is stored on the ledger as the result of "strong" encryption, whose result is then hashed instead of just being hashed. The risks to re-identification, however, imply also not wholly relying on encryption for on-chain anonymous data that could constitute personal data since it is not clear at all whether the still uncertain prospect of quantum computing should be taken into account (Finck, 2019; Lyons et al., 2018; Sovrin Foundation, 2020). For the scope of this work, such risk is not taken into account.

2.4.1.3 Specialized DLT-based system that forgets

A DLT that can arbitrarily delete information might seem to be the perfect solution to resolve many of the tensions that come with GDPR (Erbguth, 2019a). However, the principle of immutability is at the very core of DLTs. As the two previous subsections show, for some personal data types, a DLT such that information stored on-chain (e.g., hash pointers) can be rendered valueless works favorably to the GDPR. It may be the case of the deletion of off-chain stored (encrypted) data and the destruction of encryption keys and correlated information. However, the use of special "forgetting" DLTs may still make sense because the deletion of encryption keys, while leaving the encrypted/hashed data on-chain, does not completely adhere to the data deletion definition (Article 29 Working Party, 2014a; Finck, 2019; Finck and Pallas, 2020). A DLT operating through a protocol built to remove old transactions without affecting the security of the DLT structure would favor the right to be forgotten and, above all, the data minimization principle (CNIL, 2018a; Politou et al., 2019). In order to fulfill the data subject's rights, when the permissioned DLT nodes act as data controllers, several approaches are presented in state of the art. Two of them stand out, i.e., using the implementation of a redactable (or forgetting) DLT (Farshid et al., 2019; Florian et al., 2019) or using pruning (Finck, 2019; Politou et al., 2019). While the first approach consists of a specific case of DLT implementation, the pruning algorithm, on the other hand, can be potentially implemented in all DLTs. This algorithm, foreseen by Nakamoto in Bitcoin (Nakamoto, 2009), consists of deleting old transactions and blocks (on demand or after a predefined period) and keeping the old block headers containing the hashed version of the removed data in order to ensure the security of the blockchain. Although this technique has been judged to be weakly applicable in permissionless blockchains, pruning may provide a suitable solution for permissioned blockchain, where the operating environment is more easily controlled and regulated (Palm, 2017; Politou et al., 2019). In the case in which a DLT configuration cannot be altered in favor of pruning or redaction, the solution of storing data off-chain remains the most pleasing for GDPR compliance. Nevertheless, also, in this case, some considerations must be taken into account. The use of DFS is often considered in state-of-the-art solutions (Aiello et al., 2020; Onik et al., 2019; Wang et al., 2018). Taking IPFS (Benet, 2014) as a reference, it consists in a trustless network, where nodes can never be guaranteed to comply with a content deletion request because there is no way to verify that data has been removed from the entire network (Politou et al., 2020). A solution for data deletion could be the construction of a private IPFS network, where the participants agreed on the duties towards data subjects, thus making it possible to delete a file from all nodes of the private network. This, however, would limit the data availability and portability. Suppose one refers to a public IPFS network, on the other hand. In that case, one can take the anonymous delegated deletion of Politou et al. (2020) based on Article 17(2) of the GDPR: “the controller, [...] shall take reasonable steps, including technical measures, to inform controllers which are processing the personal data that the data subject has requested the erasure”.

2.4.1.4 De-linking DLT-addresses from identities

One important consideration must be made for DLT addresses and the data they are derived from, i.e., asymmetric keypairs of public and private keys. The use of such key pairs in digital signatures is considered a pseudonymization technique (Article 29 Working Party, 2014a), and thus this data is still under the scope of the GDPR. The use of public keys in DLTs, i.e., to derive addresses and digitally sign, cannot be considered an anonymization technique because of their high risk of being de-anonymized (Finck, 2019; Finck and Pallas, 2020). These public keys, however, are considered by the CNIL as data that “cannot be further minimized and that their retention periods are, by essence, in line with the blockchain’s duration of existence” (CNIL, 2018a). Therefore, it can be argued that their combination with a "conscious" use of PETs could potentially meet the GDPR data minimization requirements (Giannopoulou, 2020). However, several approaches can be leveraged to decrease the possibility that the owner of a public key may be identified as a natural person, and many DLTs rely on this (Christensen, 2018). For instance, one can refer to the Dual-Key Stealth Address Protocol (Courtois and Mercer, 2017) for limiting, when possible, the identification of subjects in permissioned DLTs, but also permissionless ones.

2.4.1.5 Define accountability obligations among the network participants

The most stressed recommendation in the various analyses on the GDPR compliance of DLT-based solutions is the definition of liability obligations for the participants in the system. Several suggestions embrace the shared responsibility approach (CNIL, 2018a; Rieger et al., 2019a), in which all participants in a DLT network act as joint controllers and stipulate an agreement establishing the respective responsibilities of each participant. The legal basis for personal data processing related to network participants and/or third parties is guaranteed by mutual agreements. In a permissioned DLT, it is possible to adopt extensive measures to minimize the transmission and/or storage of any personal data through the ledger, following the principles of Privacy by Design, e.g., pruning. Permissioned DLT nodes, as stated earlier, may act as joint controllers for the transactional data that they verify, store, and put on/off-chain. However, if they process a subject request for distributing personal data, DLT nodes are likely to be data processors as they are acting on behalf of a subject that acts as controller of his/her data, i.e., SSI (Edwards et al., 2019; Giannopoulou, 2020). DLT nodes can be controllers for some activities and processors for others (Lyons et al., 2018; Sovrin Foundation, 2020). For nodes of peer-to-peer networks that provide other decentralized services than DLTs, such as DFS, the discussion on liability obligations can become complex. Most of the time, these actors process encrypted data only, which is considered pseudonymous in principle and thus subject to the GDPR. In practice, using appropriate PETs, these providers handle meaningless and anonymous data without additional information (e.g., a decryption key). Thus they are likely to have no obligations as long as the data is as such. In the case of information leakage (e.g., key leakage), the risk of being identified (e.g., through data decryption) for a subject increases, and, then such data may fall into the personal data definition (Article 29 Working Party, 2014a; Finck and Pallas, 2020). Consequently, the DFS service provider's role becomes a controller role. When we talk about a "traditional" file storage service, e.g., cloud storage, it may be easy to identify an accountable actor. However, in the case of a DFS service, it may be more challenging (Politou et al., 2020). Finally, permissionless DLT nodes generally have no obligations since they might not even be aware of any information flowing (e.g., non-personal data). However, they can be considered processors if a subject issues a transaction on the permissionless DLT (CNIL, 2018a; Finck, 2019; Sovrin Foundation, 2020).

Chapter 3

Methodology

3.1 Objectives

As seen in the previous Chapter, the exploitation of personal information is assisted by the more pervasive nature of today's digital world and the lack of data subjects' direct control. On the basis of such a lack, the overall objective of this work is the following:

> To investigate methodologies and to design systems that direct the personal data control flow towards individuals in the European Union regulatory framework.

This main objective is composed of the succession of three sub-objectives:

O1. To identify the systems that can be used to de-centralize the exploitation of personal information and their legal compliance with the EU regulation.

O2. To design and implement systems that store and/or trace personal data, and that provide access to them only through policies set by data subjects or based on GDPR legal bases.

O3. To design and implement interoperable mechanisms that enable the universal identification of personal data, policies and credentials, in such a way that the data subjects can be sovereign to decide how to store and share their data.

In order to achieve these goals, the following limitations detected in the state-of-the-art had to be addressed:

L1. The lack of technical solutions that provide access to personal data and fully transparent mechanisms that are not fully controlled by a single entity (different from the data subject).

L2. Distributed systems that can contemporaneously provide interoperable data access policies and data redundancy. State-of-the-art solutions only provide one of them, or necessitate the data subject (or a delegate) to maintain a self-hosted personal data storage, leading to a single point of failure again.

L3. The majority of current proposed decentralized solutions do not always have an approach that combines law and technology, but on the other hand, gives priority to one of the two.

3.2 Contributions

The main contributions of the thesis are outlined below.

C1. Identify aspects of GDPR compliance involving systems based on decentralization and data immutability, particularly regarding the limitation of Distributed Ledger Technologies.

C2. Design and implement a **decentralized Personal Information Management System** that stores and traces the transfer of personal data in a transparent and non-centralized manner, presented in Part II.

C2.1. **Personal Data Space**. Design and implementation of storage for personal data built on top of a Decentralized File Storage. It will be presented in Chapter 4.

C2.2. **Decentralized Indexing System**. Design and implementation of a system based on a Distributed Ledger Technology and a Distributed Hash Table for respectively indexing and querying data. It will be presented in Chapter 5.

C2.3. **Distributed Authorization System**. Design and implementation of an authorization system based on smart contracts use and a distributed access control mechanism. It will be presented in Chapter 6.

C3. Govern data access through privacy policies expressing GDPR legal bases in the presented decentralized Personal Information Management System. It will be presented in Chapter 7.

C4. Model and implement the verification of authenticity of some claims in digital interactions based on the Self-Sovereign Identity paradigm and on the intelligibility of contextual information. It will be presented in Chapter 8.

3.3 Assumptions

The work presented in this thesis is done under the following set of assumptions:

A1. It is assumed that the systems proposed are developed for general-purpose functioning, meaning there needs to be a discussion on the specific kind of personal data managed. For instance, managing health personal data with the proposed systems might require a detailed specification, both from the technical and legal perspective, that is not fully covered even though the systems can be compatible with it.

A2. We assume that the services operated by the providers implementing the systems presented in this work comply with regulations that are not explicitly cited but are usually needed for their provision. When one or a set of interrelated regulations are not explicitly cited, it means that, with respect to the technologies involved, these are not a matter of research in literature or are not in the scope of our research (e.g., Anti Money Laundering regulations).

A3. We assume that is feasible and desirable for a final user to interact with all the proposed decentralized systems through a device, as in current state of the art technology environment.

3.4 Hypotheses

The general hypothesis of this work is that disintermediation in the storage and exchange of personal data can be used to build personal information management systems compliant with the law in the EU. It can be subdivided into three sub-concepts:

H1.1. Decentralized systems based on Distributed Ledger Technologies can support the replacement of current ICTs-platform-centered personal data management in terms of feasibility, efficiency, and legal compliance.

H1.2. The distributed execution of smart contracts and dedicated cryptographic schemes can be efficiently employed to provide personal data access control without relying on a trusted third party.

H1.3. A decentralized personal information management system can effectively be integrated with, and Semantic Web technologies and standards to (i) enforce data subjects's and/or GDPR-based data access policies, (ii) provide an interoperable way to move personal data and trace related processes, and (iii) identify information related to a data subject's online identity universally.

The main research question that supports this thesis is:

> Are decentralized technologies and semantic web standards able to support individuals' personal data protection and portability optimally?

This can be subdivided into a series of sub-questions:

- **RQ1.1** - Are decentralized technologies able to support the replacement of current platform-centered data protection management?
- **RQ1.2** - How can semantic web vocabularies and disintermediation foster a convergence between the protection of individuals' data and the development of data sharing solutions?
- **RQ1.3** - To what extent can decentralized systems and EU regulations such as the GDPR coexist in order to effectively shift the de facto control of personal data sharing to data subjects?

3.5 Restrictions

The work presented in this thesis is subject to the following restrictions:

R1. The scope of the research will be restricted to personal data that are represented as content stored in an accessible format, such as text files or standard media formats. Throughout this work, we will use the general concept of personal data to refer to an instance of a digital representation of the data.

R2. The legal analysis is also limited to the European Union framework, with no specific national implementation referenced. The analysis is mainly focused on the GDPR.

R3. The software to be run in distributed environments has only been deployed to network where nodes were managed by the authors of this work, and never by independent network node entities.

R4. The models presented in this work have not been proved formally but empirically.

3.6 Research methodology

The research work for this thesis involves several disciplines and areas of information management that may seem completely far apart. Thus the methodology followed in this work is based on two phases:

- The research phase where the state of the art of current problems and related solutions are investigated.
- The design phase where the findings of the previous phase are used as drivers for the creation of an architecture as a solution to the main problems found.

In particular, the research phase is considered as the fundamental step that will precede the design of any software, as it comprehends functional and normative analysis of personal data collection and their use by first-party and third-party entities. The end result of the state of the art analysis will be the input leading to a software solution where functional and non-functional requirements are based on the previous findings. A particular emphasis is given to the GDPR analysis and its compliance.

Based on the above premise, the methodology used in this work consists of several parts.

The first part is related to the design and development of a Personal Data Space. The ground of this work consists of the initial use of decentralized systems. The first contribution consists of developing a Personal Data Space and assessing which related technologies can be better exploited in the final architecture by conducting several performance evaluations and solution comparisons. Some work has been done towards this goal by publishing relevant papers. The turning point of this part of the work is marked by the definition of functional and non-functional requirements, which has helped finalize the system architecture. The non-functional requirements consist of a process of analysis of GDPR, and therefore it is the primary driver of the solution. After the complete design of the architecture, Minimal Viable Prototype has been presented, with the ability to provide the essential functions of storing and giving access to data.

The second part of the methodology consists of the research on the use of the smart contract technology for the management and portability of personal data through the use of rights expressions. The work regarding the study of the current solution has been carried out through: (i) the participation in an ISO standard working group for the development of methods for the conversion of contractual rights into smart contracts; (ii) the creation of a smart contract based framework for decentralized identity management that supports legal reasoning. Both lines of work enabled to focus the research on smart contracts toward the exploitation of languages and ontologies to represent access policies. The output of this part of research is the ISO/IEC 21000-23 Smart Contracts for Media international standard, of which I am one of the editors, and an on-going work of publishing relevant papers.

The final part of this research methodology involves the integration of all findings and software in a decentralized Personal Information Management System and evaluating it. This is mainly driven by the non functional requirement of supporting the system's data subjects as placed at the center of an Internet of Person built entirely for them. Then the employed methods in this case include the evaluation of systems' performances from the user's point of view, simulation of a real-world scenario of such a network of individuals and a measure of the expressing of policies in real-world use cases.

3.7 Evaluation methodology

Regarding the hypotheses defined in this thesis, the following evaluations have been performed:

E1. (for H1.1). The goal of this evaluation is to test whether the implementation of the decentralized Personal Information Management System improves the results by the state of the art solutions in terms of efficiency and legal compliance. To this end, we built the proposed systems, proved their compliance, and evaluated their efficiency from the final users' perspective.

E2. (for H1.2). The goal of this evaluation is to assess that the event distributed execution of smart contracts can support cryptographic schemes to provide access to some encrypted personal data. To this end, the results are compared with different schemes presented and/or implemented for the evaluation.

E3. (for H1.3). The goal of this evaluation is to determine the completeness of the models built to represent privacy policies and online identities. To this end, the proposed models are used to represent the privacy policies and Self-Sovereign Identities.

Part II

DECENTRALIZED PERSONAL INFORMATION MANAGEMENT SYSTEM

Chapter 4

Personal Data Space

The content of this chapter is based on the contributions published here:

- M. Zichichi, S. Ferretti, and G. D'Angelo, *"A Framework Based on Distributed Ledger Technologies for Data Management and Services in Intelligent Transportation Systems,"* IEEE Access, pp. 100384–100402. IEEE, 2020.
- M. Zichichi, S. Ferretti, G. D'Angelo, and V. Rodríguez-Doncel, *"Data Governance through a Multi-DLT Architecture in View of the GDPR,"* Cluster Computing, pp. 1–28. Springer Nature, 2022.
- M. Zichichi, S. Ferretti, and G. D'Angelo, Handbook on Blockchain, *ch. Blockchain-based Data Management for Smart Transportation*, pp. 1–29. Springer Nature, 2022.

The software produced during the development of this chapter is stored here:

- M. Zichichi (2019). Data and scripts for IOTA vehicular scenario. DOI: `10.5281/zenodo.4572578`
- M. Zichichi (2020). Data and scripts for IPFS and Sia vehicular scenario. DOI: `10.5281/zenodo.3552198`

This Chapter acts as the first step toward describing a solution that can help move in the direction of "*empowering individuals with respect to their data.*" Of course, this is one of the main aims of this work, but in this particular instance, we refer to the objective set in the European Strategy for Data (European Commission, 2020). The requirement that "[i]ndividuals should be further supported in enforcing their rights with regard to the use of the data they generate" can be possibly addressed by the use of Personal Data Spaces (PDS), defined as tools and means to let individuals decide **at granular level** about the processing of their data. Indeed, dedicated technologies can help companies comply with GDPR (e.g., the portability right of Article 20) and individuals exercise their rights. The result would address two main issues: the lack of transparency in managing personal information and the inability to access and make personal data interoperable. The PDS can be intended as the first step toward this aim, relying on a new user-centered model for managing personal data, where storage is decoupled from the application logic (EDPB, 2016; ENISA, 2021). This vision is beneficial for the privacy needs of the individual and for building a single data market (European Commission, 2020) that capitalizes on the data interoperability in data spaces for the social good (Furini et al., 2020). Furthermore, providers of personal data apps and the so-called data intermediaries (or neutral brokers) can exploit PDS to prove their compliance with regulations[1]. To this end, the European Union Council has recently approved the Data Governance Act (DGA) (European Parliament, 2022) that aims to enable personal data to be used with the help of a "*personal data-sharing intermediary*". This entity type is designed to help individuals exercise their rights under the GDPR and enable data sharing on altruistic grounds (i.e., Article 2(10) data altruism). A data intermediary can be defined as a mediator between those who wish to make their data available and those who seek to use them while providing some degree of confidence about how the data will be used (Janssen and Singh, 2022a). Recital 23 of the DGA, in particular, specifies that data intermediaries are a category of data sharing service providers who offer their services to data subjects under the GDPR to strengthen individuals' agency and control over their data. Moreover, "*[t]hey would assist individuals in exercising their rights under Regulation (EU) 2016/679, in particular managing their consent to data processing, the right of access to their data, the right to the rectification of inaccurate personal data, the right of erasure or right 'to be forgotten, the right to restrict processing and the data portability right, which allows data subjects to move their personal data from one controller to the other.*"

[1] This issue is also dealt with in the context of the Data Act and Chapter 7 of this work.

In this Chapter (and throughout this whole work), we stem from this vision, and we use such a description of data sharing and intermediary services as a design driver for our system. In particular, we propose a decentralized approach for the design of a PDS based on distributed ledger technology (DLT) and decentralized file storage (DFS). In line with the DGA (and GDPR and Data Act), such decentralized environments can be composed of **known**, but yet distributed, node operators: "*Data intermediation services could include bilateral or multilateral sharing of data or the creation of platforms or databases enabling the sharing or joint use of data, as well as the establishment of specific infrastructure for the interconnection of data subjects and data holders with data users*" (European Parliament, 2022). The resulting PDS system we propose is compliant with the GDPR, thus protecting users' data, and it promotes data sharing as intended by the Data Governance Act (and also Data Act). The main benefit of such a decentralized architecture is that, by bringing together regulations and technologies, it provides individuals with the ability to record their data in some interoperable PDS, guarantees data sovereignty, and enables users to control what personal data they want to share (European Parliament, 2017; Giannopoulou, 2020). The use of DLTs and DFS is of paramount importance in our system architecture. DLTs provide the technological guarantees for trusted data management and sharing, as they can offer a fully auditable decentralized access control policy management and evaluation (Maesa et al., 2019). In the view of the GDPR, this makes it possible to check whether the involved actors comply with the regulation or not, e.g., the trace personal information sharing can help data processors, and controllers easily demonstrate their compliance transparently. As concerns DFS, its combined use with DLT allows overcoming the typical scalability and privacy issues of the latter while preserving the benefits of decentralization (Politou et al., 2020). In practice, DFS is leveraged for storing the actual data outside the DLT, i.e., through "off-chain" storage, and tracing all the data references in the DLT (i.e., "on-chain").

In the following, the original contributions and novelties of this Chapter are described:

- First, we provide an interdisciplinary analysis of technical and non-technical drivers for the design of a PDS. In particular, in the background, related work, and architecture description, we refer to the GDPR and work/analyses related to this.

- Second, we describe the decentralized PDS system based on the use of DFS for the off-chain storage of personal data and a DLT for data integrity and traceability.

- Third, we provide a prototype implementation of the described system, and we evaluate its performance using an experimental evaluation. More specifically, the implementation is based on a client application for communicating with two DFS, i.e., InterPlanetary File System (IPFS) (Benet, 2014) and Sia (Vorick and Champine, 2014), and a public DLT, i.e., IOTA (Popov, 2016).

The remainder of this Chapter is organized as follows. Section 4.1 presents the background concepts behind the proposed architecture and related works. Section 4.2 has the purpose of providing an overview of the PDS architecture we propose. In Section 4.3, we specify the architecture's components, then discuss its GDPR compliance. In Section 4.4, the implementation of the PDS is described and evaluated in terms of performance. Finally conclusions are presented in Section 4.5.

4.1 Background and Related Work

This section describes the technologies used to build the proposed software architecture.

4.1.1 Distributed Ledger Technology (DLT)

Distributed ledger technologies (DLTs) consist of protocols and components that guarantee untampered data availability thanks to the immutable persistence of data in the distributed ledger. DLTs, born with the advent of the Bitcoin blockchain (Nakamoto, 2009), replicate the ledger among nodes of a peer-to-peer (P2P) network. This append-only ledger is expanded through transactions disseminated throughout the network, independently verified by each node to ensure consistency. This protocol allows the exchange of data, currency, or assets without relying on a human intermediary. In particular, DLTs enable: (i) transparency, i.e., the guarantee for the auditability of transactions and data accesses (Maesa et al., 2019; Nakamoto, 2009); (ii) security, i.e., the shifting of the trust that is normally placed to intermediaries towards a distributed consensus mechanism (Androulaki et al., 2018; Buterin et al., 2013; Singh et al., 2020); (iii) immutability, i.e., the verifiability of the data stored in the ledger (Nakamoto, 2009; Politou et al., 2019); (iv) decentralization, i.e., the ability of direct user-to-user interactions and agreements, without intermediaries (Buterin et al., 2013).

4.1.1.1 Types of DLT

Several DLT implementations exist with their pros and cons; however, all of them are built on a network of peer nodes that maintain the distributed ledger. Firstly, implementations can be subdivided into "public" and "private" DLTs. The former type consists of a DLT where anyone can have full access and read the data stored in the ledger, while in the latter, the ledger data is private. A hybrid, probably less standard, model is the "semi-private" DLT, which can be used in scenarios where a private part of the ledger remains internal and shared among known participants, while a public part can still be used by anyone (Mougayar, 2016). Secondly, we can distinguish between two main DLT categories: "permissionless", i.e., where anyone can participate in the consensus mechanism, and "permissioned", i.e., where one or more authorities act as a gate for the participation of new nodes to the consensus mechanism. Bitcoin (Nakamoto, 2009) and Ethereum (Buterin et al., 2013) are examples of public permissionless DLTs, while Hyperledger Fabric is an example of a (semi-)private permissioned DLT (Androulaki et al., 2018). A permissioned solution is often a very convenient approach since it makes it easier to compose a software architecture. Nonetheless, a consortium of trusted entities is usually required to employ a permissioned DLT. These entities can also act as certificate authorities that release public and private keys to access the ledger (Li et al., 2018). Such a solution requires trusting such a consortium (Shahid et al., 2019), while, in contrast, a permissionless approach is more suitable to enable trustless services. Furthermore, DLTs can also be distinguished from their ability to support smart contracts, as seen in Chapter 7. Conversely, some implementations are thought to provide better scalability at the expense of lacking some features, e.g., based on Directed Acyclic Graphs (DAGs) such as IOTA.

4.1.1.2 IOTA

The IOTA ledger is not structured as a blockchain but as a DAG where vertices represent transactions and edges represent validations to previous transactions, i.e., the Tangle (Popov, 2016). Such public data ledger is mainly targeted towards the use in the IoT industry. The IOTA transactions validation approach is thought to address two major pain points associated with traditional blockchain-based DLTs, i.e., latency and fees. IOTA has been designed to offer fast validation, and no fees are required to add a transaction to the Tangle (Brogan et al., 2018). When a new transaction is to be issued, two previous transactions must be selected (i.e., tips selection) and approved by referencing those in the transaction. The result is represented through directed edges in the Tangle. Proof-of-Work (PoW) is performed to validate a transaction and deter denial of service attacks and other service abuses.

4.1.2 Decentralized File Storage (DFS)

In order to overcome the typical DLTs' scalability and cloud services' privacy issues, the use of a DFS is a potential solution for storing files while maintaining the benefits of decentralization. Such storage offers higher data availability and resilience thanks to data replication. A DFS is crucial for DLTs, as it can be leveraged to store data outside the DLT, i.e., off-chain, when the consensus mechanism discourages on-chain storage.

4.1.2.1 IPFS

The InterPlanetary File System (IPFS) (Benet, 2014) is a DFS and a protocol that provides a distributed file system over a P2P network. The purpose of IPFS is to provide a resilient and single-point-of-failure-resistant storage system for sharing data, which does not depend on mutual trust between network peers. In the network, files are represented by IPFS objects and are identified by a CID (content identifier), i.e., the digest produced when a hash function is applied to a file. This hash digest, or CID, is also used to retrieve the referenced IPFS object.

A critical remark is that peers in the IPFS network have no incentive to maintain objects when asked to replicate them. A peer maintains a replica of an object until it needs to free up space in its local storage (a process called "unpinning").

4.1.2.2 Incentivized File Storage and Sia

In order to maintain excellent reliability and ensure that the file can be correctly retrieved, an incentive mechanism can be placed on top of a DFS. Filecoin (Benet and Greco, 2018) is an incentive layer on top of IPFS where participants are rewarded (with Filecoin tokens) for serving and hosting content on their storage. The protocol matches client requests with storage node offers through a blockchain and dedicated smart contracts. Besides Filecoin, other solutions exist that provide incentives to store data persistently. Such an example is Sia (Vorick and Champine, 2014). It consists of a DFS that also leverages smart contracts, i.e., file contracts, to arrange an agreement between storage providers and clients.

While IPFS has already been considered and evaluated in several studies (Hawig et al., 2019; Naz et al., 2019), Sia does not match this level of maturity despite looking very promising.

4.1.3 Related Work

DLTs are mainly known for their use in finance and support of cryptocurrencies. However, in recent years many efforts have been made in other areas. Features of integrity, validity, and authenticity make DLTs attractive for many use cases. In related work, many DLT-based frameworks run decentralized and provide an autonomous and traceable way to manage data.

Ramachandran et al. (2018) provide a decentralized system for data-driven smart city services, in which data are exchanged in a decentralized data marketplace. Their solution involve the combined use of DLTs and a DFS to store, validate and share data in a decentralized way. Droplet (Shafagh et al., 2018) takes advantage of a DLT to provide data holders the ability to share their data through secure encryption key derivation and management mechanisms, with a focus on Internet-of-Things (IoT), generated data. In this study, the DLT is used to hold the keys used for data encryption, and their distribution is the responsibility of the data holder. This system (and the key derivation mechanism in particular) is pluggable to alternative solutions like the one we present in this work. Related to this kind of solution is the work of Jiang et al. (Jiang et al., 2019), where the encryption operations are outsourced. Both works share the use of stealth addresses (Courtois and Mercer, 2017) to provide privacy in authorizations while maintaining traceability. In Blockstack (Ali et al., 2017), users' transactional metadata is stored in a DLT, while the data itself is stored off-chain through a cloud service provider (e.g., Google Drive, etc.). Although this technology shares similarities with our data storage proposal, it is not mainly targeted at authorizing access to third parties. (Lopez and Farooq, 2020) present a framework for Smart Mobility Data Market in which participants share their data and can transact these information with another participant, as long as both parties reach an agreement. They use a DLT managed by a consortium of trusted nodes, such as Companies, Universities and Governments, that allows the execution of smart contracts to self-enforce fair trades between participants and automatically solve disputes. Their research comprise the protection of individuals' personal information, while maintaining data transparency and users' ruled access control. They also provide the use of PETs completely targeted to LBS, such as geomasking and geo-indistinguishability. (Aiello et al., 2020) have designed IPPO, an architecture that allows users to generate and share anonymized datasets on a distributed marketplace to service providers, while monitoring the behavior of web services to discourage the most intrusive forms of tracking.

In the broader scope of DLT-based data sharing, we may find a subset of personal data management solutions that: (i) require an additional effort to comply with regulations such as GDPR; (ii) include architectural components designed following the logic of "Privacy by Design" (Cavoukian, 2009). Again, the primary purpose of a DLT-based system is to provide transparency in accessing personal data but simultaneously enable users to control their data (Aiello et al., 2020; European Commission, 2020). Nevertheless, few studies address GDPR compliance, and even these studies do not compare regulators' opinions regarding specific DLT architectural parts. Zyskind et al. (Zyskind et al., 2015) provide a system in which DLTs are leveraged to hold discretionary access control policies and track user permissions to give or deny access to data. Yan et al. (Yan et al., 2017) present a PDS that allows users to collect, store and give third parties access to their data. Their solution is innovative but expensive and not recommended for GDPR because the system stores personal data on-chain when it is possible to do it off-chain.

Fewer studies provide a system architecture that faces the conflicting characteristics of DLTs. Both (Onik et al., 2019) and (Ahmed et al., 2020) works provide significant contributions. The former propose a model that traces the life cycle of personal data through the data controllers and processors. Personal data are stored off-chain in order to respect the right to be forgotten of GDPR and the data controller informs data subjects about the sharing of data with data processors. A smart contract contains the terms and consent of the data subject for the use of personal data, which must be accepted by the processors. The second focuses on OSNs and their lack of GDPR compliant consent management mechanisms. They present some opportunities for using DLT to address this issue and to provide fine-grained control over personal data, while also discussing the challenges of DLT-based OSNs under GDPR.

4.2 Personal Data Space Architecture

This section discusses the architecture of the PDS solution we designed and developed. The general idea is straightforward: a data subject is the user of a personal device that generates different kinds of personal data; a data holder (which can be the data subject itself) stores and maintains such data in a PDS and a cryptographically immutable reference is stored in a DLT. In the following, we will illustrate the choices behind using such components and their interactions.

4.2.1 Architectural drivers

Several major considerations influenced the architecture of the system we present in this Chapter. We describe them in the following.

4.2.1.1 GDPR Compliance

We examined compliance issues in terms of GDPR in order to **effectively** provide a tool for individuals to exercise their rights, adhering to the most compatible solution possible. Although difficult (see Section 2.4 in Chapter 2), DLT-based solutions can be designed to be GDPR-compliant, provided they are developed with appropriate design features to protect privacy. The state of the art includes studies by regulators such as CNIL, AEPD, STOA (Agencia Española De Protección De Datos, 2019; CNIL, 2018a; Lyons et al., 2018), and companies, organizations and scholars (Erbguth, 2019b; Finck, 2019; Finck and Pallas, 2020; Giannopoulou, 2020; Kondova and Erbguth, 2020; Molina et al., 2020; Rieger et al., 2019a,a,b; Sovrin Foundation, 2020; Zemler and Westner, 2019).

The best practices for a DLT-based design compatible with GDPR include the principles described in Sub-Section 2.4.1 in Chapter 2. In the continuation of this Chapter (and in Chapter 6), we will refer to such practices as these are the pillars of the system we built.

4.2.1.2 Relation between Data Protection and Cryptography

In this work, the implementation of the guarantees related to applying the principles established by the GPDR is based both on the application of specific recommended cryptographic techniques and, most importantly, on the organizational and architectural measures taken following the Privacy by Design approach. Regarding Article 32, controllers and processors must comply with the implementation of "appropriate technical and organizational measures, to ensure a level of security appropriate to the risk". Pseudonymisation is an accepted data protection measure in the adoption of the GDPR (Article 4(5)) and many times referenced as a safeguard (ENISA, 2021), while anonymization techniques can generally provide a strong privacy guarantee (Article 29 Working Party, 2014a). In general, however, providing only pseudonymization and encryption technical solutions that are formally verified secure is not a sufficient and necessary condition to state that data processing security is appropriate to the risk. One has to tackle the data protection problem from a higher point of view, "going beyond the 'traditional' understanding of security" (ENISA, 2017). In our work, we focus on data protection by design and by default by following the guidelines of different cybersecurity agencies and supervisory authorities, which also indicate appropriate pseudonymization and encryption solutions (Agencia Española De Protección De Datos, 2019; Article 29 Working Party, 2014a; ENISA, 2021; Lyons et al., 2018).

In particular, the use of a mixed symmetric-asymmetric cryptosystem, advanced cryptographic hash functions, and DLT validation (all of them explained in detail in the following Sections and also later in Chapter 6) has been covered in these recommendations as a form of pseudonymization and/or anonymization.

4.2.2 Actors and architectural components

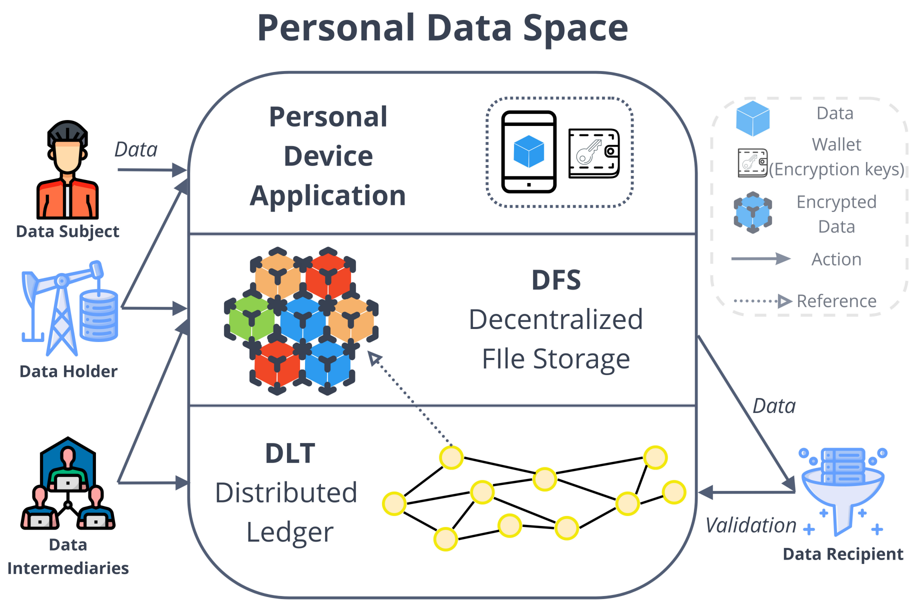
Our first aim is to identify the actors involved in the architecture of the PDS. Then we will give an overview of the architectural components.

4.2.2.1 Actors

We define different actors that have one or more roles in the system. The focus is obviously on the definition of accountability obligations among the system actors and participants since it represents a crucial point for the GDPR. In detail, we identify the following actors:

- ***Data subject*** (DS) - The natural person that uses a personal device that in turn generates personal data.
- ***Data holder*** (DH) - The legal or natural person who has the right or obligation or the ability to make available specific data (both personal and non). Under the GDPR terminology, a data holder can be either a subjectacting as his/her own data controller (i.e., following the Self-Sovereign Identity paradigm (CNIL, 2018a; Giannopoulou, 2020)), or a third-party data controller. In the second case, it can be the entity that produced a product or offers a service that created personal data on the subject's behalf.
- ***Data intermediary*** (DI) - The legal or natural person who mediates between those hoders who wish to make their data available and data recipients. We have two specializations of data intermediary:
  - ***DFS provider*** (SP) - The one that provides the access to the PDS. This actor provides functionalities attributed of storing and serving (encrypted) personal data. Concerning the GDPR, the DFS provider acts as a data controller; however, there are some exceptions, which will be addressed later in this Section.
  - ***DLT provider*** (LP) - The one that provides the access to the DLT. As we will see later (audit node in Chapter 6), some kind of DLT providers may be completely unaware of the exchange of personal data and perform the DLT consent mechanism only by handling non-personal data. In terms of GDPR, the DLT provider acts as a data controller and/or processor. This matter will be later discussed in this Section.

  In both cases, depending on the type of implemented PDS (permissioned or not), the data controller could have agreed on contractual terms that define precisely the roles and duties and the privacy policy towards end users.
- ***Data recipient*** (DR) - The legal or natural person to whom the data holder makes data available. Again, following the GDPR terminology, since the latter is a data



Data subjects interact with their personal device application providing data and keeping secure encryption keys in a wallet. The DFS is used to store encrypted data that recipients can later access. The latter can validate the data against the information found in a DLT.

Figure 4.1: Diagram showing a layered vision of the whole PDS architecture.

controller, the recipient is a data processor by definition. It is often referred to as “Data User” or “Data Consumer”, e.g., in (European Parliament, 2022).

4.2.2.2 Components

Figure 4.1 shows the components of our architecture graphically. These are:

- **Personal device application** - A dedicated client application allows data subjects to decide how/where to store the data and, if the subjects also acts as a data holder, to handle data encryption and secret key generation.

- **(Decentralized) file storage** - It consists of the first and last component with which the users interact since it embodies the storage of the PDS. It contains the data to encrypt or decrypt and can be implemented in different ways, e.g., as centralized cloud-based storage or as a DFS.

- **Distributed Ledger Technology** - DLTs allow avoiding all the typical drawbacks of server-based approaches, such as censorship and single-point-of-failure, and offer features such as data immutability, verifiability, and, most importantly, traceability. These can be used to obtain immutable references to personal data and provide a tamper-proof log, which can be consulted in case of a dispute.

4.3 Components Design

In this Section, a description of the components of the architecture will be provided. Indeed, each component will be examined vertically, with respect to the technology stack, as well as in terms of GDPR compliance. The following will provide a notation representing the informational elements used in the implemented systems. This notation will also be needed to build a model in Chapter 6.

4.3.1 Personal device application

A subject acting as data holder interact with the PDS through a client application. Such a system component contains the software to communicate with the DFS and DLT. Most importantly, it implements cryptographic operations that are used to protect personal data right at the user's device level. If subject and holder are two separate entities, then the holder's device would implement the same cryptosystem described below, while the personal device application would allow the subject to set access policies (see Chapter 6). The data recipient's device implements too the same cryptosystem described in the following.

A specification is needed, at this point, for what regards personal data created. Personal data can be created:

- directly by the subject on its personal device;
- indirectly (i.e., on behalf of the subject) by a data holder through a proprietary product or a service.

In the first case, the subject would act directly as a data holder and decide in the first place for the storing of personal data. In the second case, the subject would indicate some policies (as described in Chapters 6 and 7) to the actual data holder, e.g., employing GDPR's consent. In both cases, we can have two types of data:

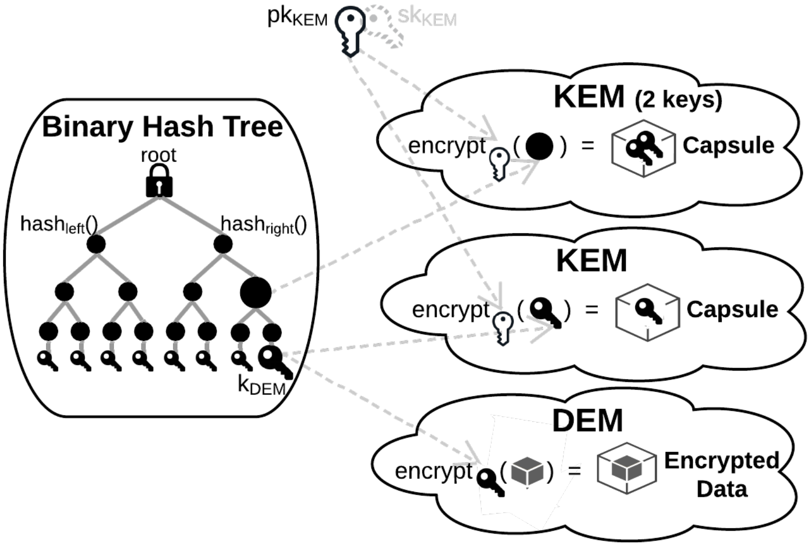


Figure 4.2: Data and key encapsulation mechanisms.

- a "static" datum - describe a subject's static property (because it never or rarely changes), e.g., the subject's date of birth.
- "dynamic" data - describe time series data or a subject's property that changes in time, e.g., the location of a subject. In this case, each element of the sequence represents a particular value associated with that property at a given time (thus, the data structure logs all the value versions).

4.3.1.1 Cryptosystem and Wallet

The personal device application, thus, implements part of the cryptosystem that represents a critical part of the whole PDS, crossing vertically all the other components described in our architecture. We assume that each actor has its unique pair of asymmetric keys, i.e., public key pk_{KEM} and private key sk_{KEM}. And an example of notation is as follows: (pk_{DH}, sk_{DH}) is the keypair of a data holder DH; sk_{SP_i} is the private key of the i-th DFS provider SP_i. In the development of DLT-based applications, the term "wallet" is generally used to represent that piece of software that maintains the cryptographic keys needed to interact with the system. In this work, we will use the same term to indicate that part of the device application that creates and stores keys and executes the cryptographical operations.

The core of our cryptosystem consists of using a hybrid encryption scheme (Herranz et al., 2006). It consists of a scheme where an asymmetric public key pk_{KEM} is used to encrypt a symmetric secret key k_{DEM} through a key encapsulation mechanism (KEM), and then k_{DEM} is used to encrypt the actual data through a data encapsulation mechanism (DEM). This procedure is graphically described in Figure 4.2 and throughout this Sub-Section. The KEM/DEM technique combines the efficiency of symmetric cryptography with the benefits of asymmetric cryptography (Herranz et al., 2006).

Personal data are encrypted using a symmetric content key k_{DEM}. Then, the key is placed in a "capsule" through a KEM, i.e., the content key is encrypted with a public key in a keypair (pk_{KEM}, sk_{KEM}). This capsule is, then, the representation of the hybrid encryption scheme and consists of $c_{k_{DEM}} = Enc_{pk_{KEM}}(k_{DEM})$.

The content key k_{DEM} associated with a static datum is randomly generated. As concerns dynamic data, content keys associated with elements of the sequence are organized as a hash tree, in order to facilitate their management and retrieval. In this case, the *root* of the tree is a secret seed, while child nodes are constructed top-down through the use of multiple hash functions. For instance (see Figure 4.2), a binary hash tree can be constructed starting from a *root* and then recursively creating a left and a right child using, respectively, a left hash function and a right hash function (Shafagh et al., 2018). Finally, leaf nodes are used to derive the content key k_{DEM} and are assigned to each sequence element in an ordered manner, e.g.,time ordered for time series data. In this case, in order to share a sequence interval, the internal tree nodes are encapsulated ($c_{node} = Enc_{pk_{KEM}}(node)$) instead of the content key k_{DEM} of each sequence element. This facilitates data recipients to generate the corresponding set of content keys from a single source, i.e., the internal *node* of the tree.

As an example, we take two cases, one for static data and one for dynamic ones:

- A static datum can be for instance the name of the model of the smartphone a user (i.e., the data subject) is using to interact with an online service; in this case, the $data =$ "iPhone 13" is encrypted using a random content key k_{DEM} stored in the wallet, i.e., $encrypted\_data = Enc_{k_{KEM}}(data)$.

- A dynamic dataset can be, for instance, a location trace of the smartphone user. In this case, the dataset is composed of an ordered (by time) set of data points such as $data_i =$ (*Fri, 16 Sep 2022 09:15:35*, *Latitude: 63.1702*, *Longitude: 29.9086*). A *root* can be created from a random key $root = k_{DEM}$ for creating a binary hash tree associated with a trace of a road trip by car. If data points are generated each minute and $data_i$ represents the position at the i-th minute, whenever a data recipient is interested in obtaining data for 4 consecutive minutes, only an internal *node* of the tree is needed instead of the 4 content keys of each datum k^i_{DEM}. This $node_{l,m}$, with ($l < i < m$), is the one from which the 4 content keys are generated, e.g., $k^i_{DEM} = hash_{left}(hash_{right}(node_{l,m}))$.

4.3.2 (Decentralized) File Storage component

DLTs got momentum, mainly for their ability to write data that remain permanent, i.e., immutability. In most proposals, DLTs are specifically designed to make it difficult to change or delete data, i.e., immutability, and this is leveraged as one of its strengths. In many cases, however, in combination with DLT, off-chain storage is used to store files, keeping only some pointers to those files in the ledger, i.e., hash pointers (see principles 1 and 2 in Sub-Section 2.4.1 in Chapter 2).

In our design, personal data is kept in a PDS associated with a data subject. This PDS consists of the set of encrypted personal data referring to the subject that is stored in a DFS. The use of a P2P network and data replication mechanisms inherent in a DFS, make it so that the storage of personal data is decoupled from both the DLT and the personal device to provide wider data availability. This allows having different DLTs and/or services to refer to the same data storage system and facilitates the creation of a PSD in the perspective of data portability (Article 20, GDPR). In the continuation of this Section, we are going to always use the reference to a DFS to build up a PDS, however, also a more centralized version of a PDS can be set up. Indeed, a PDS can consist of either any commercial cloud file storage service (e.g.,Azure, Google Drive, etc.) or a decentralized one, as no operational logic is required at this level other than storing and obtaining encrypted data. A data holder can use a centralized (proprietary) storage solution, instead of a DFS, for maintaining a PDS. This is completely compatible with our PDS design, as long as the data content stored in the centralized storage can be uniquely referenced and data can be accessed by recipients.

Uniquely referenced means that the resource containing the piece of personal data, e.g., Sectione contained in the (de)centralized file storage, should be identified by making use of a specific protocol to keep the content unmodified for verifiability.

Specifically, instead of referring to a resource “normally”, i.e., *“resource-name-1”*, we make use of the resource content hash digest, e.g., by using the IPFS content id or CID *“QmdmQXB2m...KxDu7Rgm”* (Sub-Section 4.1.2). This is in line with the fact that if the content was a specific one at the time of storing the piece of data in the PDS, an audit must verify that, subsequently, the file may have been altered.

4.3.3 Distributed Ledger Technology component

To guarantee data integrity and verifiability, encrypted personal data could be stored directly on-chain, on a DLT. However, a PDS making use of a DLT needs to be designed to prohibit or prevent storing personal data on-chain (see principle 1 in Sub-Section 2.4.1 in Chapter 2).

As we will see in Chapter 6, the architecture that includes all the systems presented in this work revolves around a private permissioned DLT in a multi-layered design (as proposed by the EU Blockchain Observatory and Forum in (Lyons et al., 2018), Principle 3). Since we have not still introduced the necessity for an access control system and we are describing the PDS as a standalone component, in this Section we are going to focus on the most generic implementation of it, in which the DLT can be not "tightly controlled". Indeed the use of the ledger up to now, with regards to the PDS, is limited to the logging of hash pointers. We follow the approach to reference data and their content on-chain, e.g.,through a hash pointer, and to store them off-chain in a DFS. Once a personal datum is hashed the resulting digest can be used as a pointer, i.e., a hash pointer. Thus, the reference, in the form of a hash pointer stored on-chain, allows for retrieval of data and to verify their integrity.

4.3.4 Design for the GDPR Compliance

This Sub-Section describes the architectural and implementation choices that have driven our design. We refer to Table 4.1 that summarizes the uses of data in the architecture.

4.3.4.1 Encrypted personal data

Table 4.1 presents an asterisk (*) in the pseudonymous type for encrypted data (and for hash digests that are derived from them) because of their risks to re-identification.

Table 4.1: Data classification and use in the architecture

Data	Type	Source	Storage Location
Static and Dynamic Personal Data	*Personal*	Subject Personal Device or Data Holder Device	Private: • Subject Personal Device • Data Holder Private Storage
Encrypted Data	*Pseudonymous**	Encrypting personal data	Private/Public: • Decentralized File Storage • Data Holder Private Storage
On-chain Hash Pointers	*Pseudonymous**	Hashing Encrypted Data using Single-Use Salt	Private/Public: • DLT
Address	*Pseudonymous*	Created from the Subject Personal Device Wallet	Private/Public: • DLT

Even if a conscious use of PET (see principle 2 in Sub-Section 2.4.1 in Chapter 2) can, currently, render encrypted data basically impossible to de-anonymize, it is not clear at all whether the still uncertain prospect of quantum computing should be taken into account (Finck, 2019; Lyons et al., 2018; Sovrin Foundation, 2020). Thus, even if the data stored in the DFS are all and only encrypted, the DFS provider should be accountable as the data controller because such data can be considered pseudonymous. However, in practice, appropriate techniques can be used so that these providers handle meaningless data without additional information (e.g., a decryption key). In such a case, DFS providers have no obligations (Agencia Española De Protección De Datos, 2019). The same discussion holds for DLT providers and hash digests.

4.3.4.2 On-chain hash pointers

As stated earlier, (encrypted) personal data is referenced on-chain but stored off-chain in a DFS. From the point of view of the GDPR, the result of a hash function applied to personal data (without conscious security measures) is considered by the Art. 29 Working Party (now European Data Protection Board) as pseudonymized data because of the likelihood of deriving the input value (Article 29 Working Party, 2014a), thus still in the scope of the GDPR (Recital 26). Following principle 2 in Sub-Section 2.4.1 in Chapter 2, a GDPR-compliant solution consists of the use of Key Reuse Encryption and

Single-Use Salt (Agencia Española De Protección De Datos, 2019), minimizing the risks of de-anonymization (ENISA, 2021).

4.3.4.3 DFS personal data erasure

Finally, we have to deal with the right to be forgotten and, above all, with the principle of data minimization (CNIL, 2018a; Politou et al., 2019). If encrypted data are considered pseudonymous, then their deletion from a DFS is a right to guarantee to the data subject (see Section 2.4 in Chapter 2). It is a simple task for a data intermediator in a centralized, cloud-like architecture. Dealing with DFS is more complicated yet. For instance, in IPFS, there is no way to force and verify that data has been removed from the entire network. However, the GDPR itself provides us with a helping hand for compliance here[2]. Thus, we follow the approach presented in (Politou et al., 2020) for the anonymous delegated deletion protocol, in which the deletion is not entirely granted, but reasonable steps to inform IPFS nodes, i.e., DFS providers, for data deletion can be taken.

4.3.4.4 DLT personal data erasure and addresses

In the case of DLTs, we consider on-chain hash pointers (i.e., salted hash digests) and addresses as pseudonymous data, and they need to be deleted (see principle 3 in Sub-Section 2.4.1 in Chapter 2). When the operating environment is more easily controlled and regulated as in permissioned DLTs (as our proposal in Chapter 6, our solution includes the use of pruning (Finck, 2019; Politou et al., 2019). In permissionless DLTs, we can only refer to the conscious use of PET for protecting hash pointers, thus using Key Reuse Encryption and considering the key's deletion as the pseudonymous datum's deletion. In the case of the addresses, the discussion is different. In line with principle 4 in Sub-Section 2.4.1 in Chapter 2, several approaches may be leveraged to prevent the owner of a public key (address) from being identifiable as a natural person (Christensen, 2018). We refer to the Dual-Key Stealth Address Protocol (Courtois and Mercer, 2017) in our KEM for limiting (when possible) the identification of data recipients or subjects on DLT. Each recipient is represented by two public keys, a 'scan' pk_{scan} and a 'spend' pk_{spend}, and both corresponding private keys, sk_{scan} and sk_{spend}, are needed to generate a new address derived from pk_{new} and referable in public, i.e., on

[2] Article 17(2) of the GDPR: *"the controller, [...] shall take reasonable steps, including technical measures, to inform controllers which are processing the personal data that the data subject has requested the erasure"*.

the DLT. However, by sharing only the sk_{scan} and the pk_{spend}, the recipient can prove the ownership of the address derived from pk_{new}.

4.3.4.5 Data intermediators accountability

In permissionless P2P networks such as the ones that can form DLTs and a DFS, it is difficult to individuate liability obligations for the participants in the system, mainly because participants can be unknown or in another legislation. Thus, following several suggestions, in our final design (that will be described entirely in Chapter 6), we decided to embrace the shared responsibility approach (CNIL, 2018a; Rieger et al., 2019b).

4.4 Implementation and Evaluation

The main critical points of the devised architecture, which need to be studied to evaluate its feasibility and scalability, concern the responsiveness and reliability of both DLT and DFS systems. Our aim then, is to implement and test a PDS that integrate technologies that are currently employed in the Web 3 "world". Moreover, for our first performance evaluation we focused on two permissionless network with the aim to refine our design (as we will describe in Chapter 6, on the basis of the obtained results).

4.4.1 Implementation

The implementation of the components of our proposed PDS consists of the following:

- The **personal device application** has been developed as a module that was used for simulating several instances of devices that interact with the PDS. We performed a simulation on the basis of a scenario where personal data is uploaded voluntarely by smartphone users, i.e., data subjects (henceforth referred to only as users), to the PDS. Due to our requirements for scalability and responsiveness, we focused on some typical crowdsourced application data, i.e., the location of a user while using a public service. We employed a dataset containing Rio de Janeiro (Brasil) buses' real mobility traces (Dias and Costa, 2018) to simulate user traces on board of such buses. In our simulation, a personal device of a user on-board a bus runs a software that periodically retrieves data from sensors, e.g.,temperature, air pollution, geolocation data. These data were used to generate real requests transmitted to the DFS and DLT cmponents, with intervals provided by the real mobility traces. We simulated one user per bus. We consider:

  - Small sized data (∼100 Bytes), e.g., latitude and longitude;
  - Large sized data (∼1 Megabyte), e.g., photos.

  The implementation can be found in (Zichichi, 2019).

- The **DFS component** has been developed in three different solutions: IPFS Proprietary, IPFS Service and Sia Skynet (the difference between IPFS and SIA is explained in Section 4.1.2):

  - **IPFS Proprietary**: A dedicated IPFS node, specifically in charge of storing the data generated by our simulation, that is connected to the main IPFS network. Thus, in this configuration, simulated users' personal devices were the only ones sending requests for storing data to this node.
  - **IPFS Service**: A generic IPFS service provider, i.e., Infura (Infura Inc, 2020). This is a general purpose service that provides free access to the main IPFS public network for getting and storing data. During the tests the utilized IPFS node was receiving other concurrent requests, coming from real users all over the world.
  - **Sia Skynet**: A special Sia public node offering free access to the Sia network, but with limited storage space available. In particular, in this case, the Sia node has already formed contracts with every available host, rewarding their file replication.

  The implementation can be found in (Zichichi, 2020).

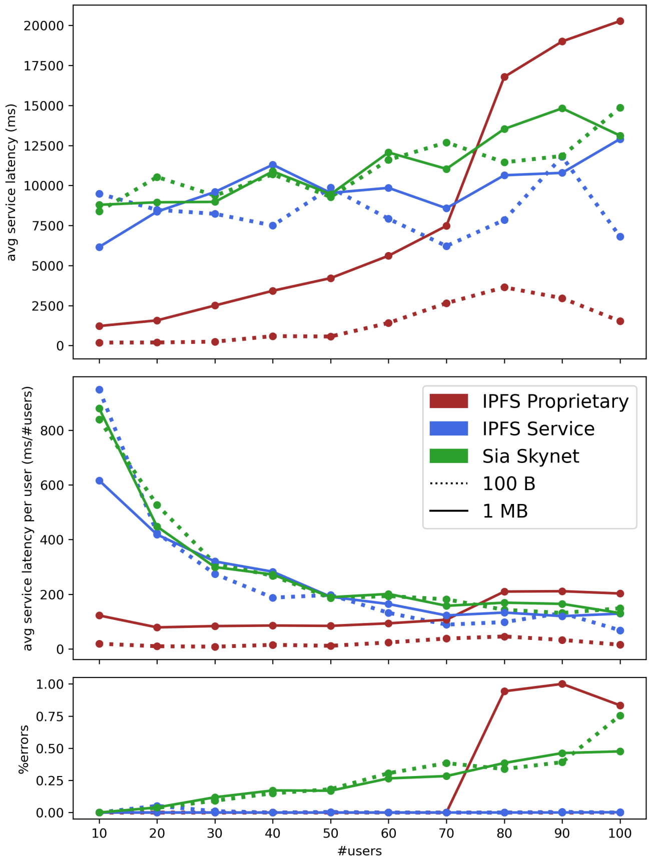
- The **DLT component** consists of the public IOTA DLT network. We exploited an important feature offered by IOTA, i.e., the Masked Authenticated Messaging (MAM). MAM is a second layer data communication protocol that adds functionality to emit and access an encrypted data stream over the Tangle, i.e., channels formed by a linked list of transactions in chronological order. Once a channel is created, only the channel owner can publish encrypted messages and the channel subscribers can read them (Brogan et al., 2018). Each user own one or more MAM channels where the on-chain hash pointers are stored. We considered different heuristics for the selection of a IOTA full node from a (dynamic) pool of public nodes to pair to each user (~ 60 active nodes). The rationale behind this choice was based on the assumption that personal devices may not have computation capabilities to behave as DLT full nodes (Elsts et al., 2018). One of the heuristics, called Fixed Random, requires each user to be assigned to a random IOTA full node from the pool for the whole duration of the test. Another heuristic was the Adaptive RTT: each user keeps a trace of past interactions with full nodes and creates a ranking based on the experienced Round Trip Time (RTT) (Jacobson, 1988); then for each message to be uploaded on IOTA a full node is chosen based on ranking.

4.4.2 Evaluation

Due to the fact that our interest is in the performance concerning data uploading from the user's point of view and that users' devices generally interact to only one peer node in both DLT and DFS networks, we focused on the request response latency in both cases for evaluating the feasibility and scalability of our proposal.

4.4.2.1 DFS Component

For testing out our implemented DFS component we simulated users' devices on a dedicated device, i.e., a Google Cloud VM n1-standard-1 with 1 Intel Skylake vCPU and 3.75 GB RAM. To evaluate the response time of the DFS provider, we conducted our performance evaluation of the three different types of DFS nodes with the objective of carrying out a stress test. The interval between one request and the next is varied by following the mobility trace dataset and varying the number of users (from 10 to 100). Each simulated user sends exactly 15 messages in 15 minutes and consecutive runs of the simulation are separated by an interval of 10 or 20 minutes. Tests were conducted in order of mesage dimension, i.e., small messages first, then larger ones, for DFS requests.



Average latency between the sending of a message to DFS providers node and the response on top. Percentage of response errors at the bottom.

Figure 4.3: Plot showing test results comparing the DFS provider implementations.

Results Figure 4.3 and Table 4.2 report the average latency between sending a message to the considered DFS node and its response, together with the confidence interval and the relative percentage of errors (i.e., HTTP status code 500 or 504). In the case of errors, the messages are not considered in the average. In Figure 4.3 results are represented as a dotted line for small messages (100 B) and solid lines for larger messages (1 MB). The plot in the middle of the Figure reports latency relative to the number of users sending requests in order to let us assess the deviation from the usual response latency. Table 4.2, on the other hand, give us a glimpse of the numbers associated to the results.

Overall, IPFS performs better than Sia in terms of latency and errors. In particular, IPFS Proprietary presents an average latency per user of about 40 ms in the case of

Table 4.2: Latencies and errors when sending messages to IPFS nodes.

Users	IPFS Node	Data Size	Avg Latency	Conf. Int. (95%)	Errors
10	Proprietary	Small	0.19 sec	[0.18, 0.2] sec	0.0%
		Large	1.22 sec	[1.17, 1.28] sec	0.0%
	Service	Small	9.49 sec	[9.09, 9.9] sec	0.0%
		Large	6.16 sec	[5.75, 6.57] sec	0.0%
	SIA	Small	8.39 sec	[7.74, 9.05] sec	0.0%
		Large	8.8 sec	[8.33, 9.26] sec	0.0%
40	Proprietary	Small	0.59 sec	[0.57, 0.62] sec	0.0%
		Large	3.42 sec	[3.31, 3.54] sec	0.0%
	Service	Small	7.5 sec	[7.18, 7.83] sec	0.0%
		Large	11.3 sec	[11.01, 11.58] sec	0.0%
	SIA	Small	10.7 sec	[10.35, 11.04] sec	14.93%
		Large	10.89 sec	[10.56, 11.21] sec	17.17%
70	Proprietary	Small	2.65 sec	[2.56, 2.74] sec	0.0%
		Large	7.48 sec	[7.3, 7.66] sec	0.0%
	Service	Small	6.22 sec	[6.09, 6.34] sec	0.0%
		Large	8.58 sec	[8.42, 8.74] sec	0.0%
	SIA	Small	12.69 sec	[12.42, 12.97] sec	38.4%
		Large	11.04 sec	[10.82, 11.25] sec	28.32%
100	Proprietary	Small	1.53 sec	[1.48, 1.58] sec	0.0%
		Large	20.27 sec	[19.71, 20.83] sec	83.33%
	Service	Small	6.81 sec	[6.69, 6.92] sec	0.0%
		Large	12.91 sec	[12.68, 13.14] sec	0.21%
	SIA	Small	14.85 sec	[14.33, 15.38] sec	75.4%
		Large	13.12 sec	[12.93, 13.3] sec	47.53%

small messages, while, conversely, both IPFS and Sia services averaged a latency of approximately one second. With regard to the error percentage, however, we note a high rate with Sia and a low level in IPFS configurations.

Focusing on IPFS Service and Sia, we can see that the behaviour between handling small and large files does not vary much. We can associate this fact with the large amount of computational resources that the two services have, as opposed to the Proprietary node which suffers greatly from the increase in file size. More in detail, the IPFS Service always has better or similar latencies to those of the Sia Service, and furthermore the error rate is very low. The Sia service, on the other hand, needs to reject more and more requests, i.e., generate errors, to maintain a stable latency as the number of requests increases. In fact, the number of errors seems to increase linearly with the number of users sending requests.

An interesting result is that, according to the use of IPFS Service and Sia, the latency experienced for each user decreases when the number of users increases. This is probably due to the fact that the data uploads are handled periodically by these providers. Hence, the more the requests in the buffer the more the data they process per interval. However, in Sia the amount of errors increases with the number of active users, meaning that the provider is not completely able to properly process all these requests.

Conversely, the IPFS Proprietary shows a linear performance per user, since according to our IPFS node implementation, data are processed as soon as they are received. For large messages, this is true only up to a certain threshold of users, after which we experienced a sudden error increase, with a corresponding latency increase. This is due to the fact that after a certain limit, the node is not able to properly handle all the requests, thus causing a cascading effect on the overall performance. In general, IPFS Proprietary always works better except for over 80 users in the case of large files. This means that a dedicated node is always preferable, but must be limited to a rate of 60-70 users requests per minute.

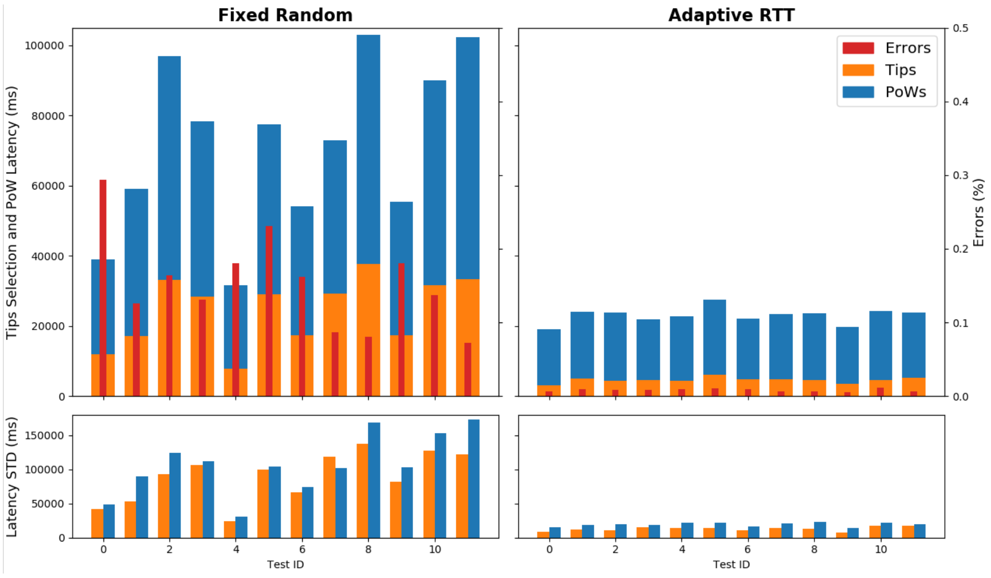
The results of this evaluation suggest that, in presence of a properly tuned DFS nodes, devoted to handle the communications with a controlled set of users, a PDS can be adequately supported in a scenario such as the one taken into consideration.

Table 4.3: Results on IOTA, with 60, 120, 240 users.

Users	Heuristic	Avg Latency	Conf. Int. (95%)	Errors
60	Fixed Random	72.68 sec	[70.43, 74.94] sec	15.37%
	Adaptive RTT	22.99 sec	[22.69, 23.29] sec	0.81%
120	Fixed Random	87.75 sec	[85.38, 90.12] sec	29.49%
	Adaptive RTT	27.35 sec	[27.11, 27.58] sec	1.1%
240	Fixed Random	177.62 sec	[174.25, 181.0] sec	42.81%
	Adaptive RTT	73.26 sec	[72.68, 73.85] sec	7.55%

4.4.2.2 DLT Component

For testing out our DLT component, we mainly focused on the IOTA node selection and the service scalability, and we simulated users' devices on a dedicated device, i.e.,



60 bus tests: average latencies, standard deviation and errors for the three different schemes (lower is better).

Figure 4.4: Histogram showing test results comparing two heuristics for issuing data to the IOTA DLT

an Intel Core i7-6700HQ CPU, NVIDIA GTX 950 GPU, and 8 GB RAM. We varied the number of user devices in the range 60, 120, 240, and for each test configuration, we replicated the experiment 12 times. For each user, we used one hour of trace data. Each user's device was set to generate approximately 45 messages/hour based on the bus paths. The result was an hour-long test, where each user generated a message to be issued to its own MAM channel every 80 sec on average (we argue it is a reasonable time interval to sense data in an urban public service scenario). Each message to be published in the MAM channel requires three Tangle transactions to be issued, i.e., one containing the data and two other messages for the signature. For each MAM message, we recorded the upload request outcome, i.e., successful or unsuccessful, as well as the latency between message transmission to a node and confirmation of its insertion in the Tangle. This time interval is characterized mainly by two operations needed for storing the transaction in the IOTA ledger: (i) the tips selection and (ii) PoW.

Results Table 4.3 shows a summary of the results obtained for different repetitions of a specific test, while Figure 4.4 shows average latency, standard deviation, and errors for a single test involving 60 users. Two main measures experienced during a series of tests are reported: (i) the average latency, including both the tips selection and PoW phases, and (ii) the percentage of errors, i.e., the number of messages that failed to be added to the Tangle, due to full nodes' errors (i.e., HTTP status code 500 or 504). Results show that, on one side, the measured latencies are relevant, and the PoW phase is the one that weighs the most, as shown in Figure 4.4. Indeed, the random selection of a full node for issuing a transaction does not yield good results since the number of errors and the measured latencies are relatively high. On the other hand, the good news is that the performance will improve if we carefully select the full node to issue a transaction. In fact, the use of the "Adaptive RTT" heuristic has a low amount of errors, on average around 0.8% and average latency amounts to 23 seconds. However, this is still far from a real-time update of the DLT, and the level of acceptability of latency values truly depends on the application scenario. In terms of scalability, results in Table 4.3 show that average latencies increase significantly with the number of users in all cases. There is an essential difference between the 60 and 240 users scenario. For 240 users, we have a message generation rate of about $\sim$ 3 msg/sec to be issued to the IOTA DLT. Assuming that the workload is evenly distributed among all the nodes in the pool, each node receives, on average, a new message request every $\sim$ 20 sec. Bearing in mind that, at best, it takes 23 sec for a full node to process a message completely, then we see that an initial overhead of a few seconds leads to a considerable increase at the end of the test, i.e., $\sim$ 73 sec in the "Adaptive RTT" heuristic. It means that further improvements are needed to solve scalability issues in this scenario.

4.4.2.3 Discussion

Several novel challenges must be faced to fully exploit the potential and promote the development of PDS based on DLTs and DFS. We argue that integrating DLTs and DFS can help create a framework providing data integrity, confidentiality, and persistence. An essential and critical outcome of this work is concerned with the implementation and experimental assessment we performed, showing the related results of the current technologies available. It is well known that decentralized and secure DLTs still have scalability issues (Bez et al., 2019). Thus, here we focused on testing DLTs and DFS where data is uploaded.

Latencies measured to store data into the considered DFS, i.e., IPFS or SIA can be considered acceptable for the urban public service scenario described. In this case, as a measure of scalability, the best performances were obtained when the number of dedicated DFS providers followed the equation $\#nodes = \frac{\#requests}{sec}$, where $\#requests$ is the number of data upload requests generated by the users in our scenario.

On the other hand, for what concerns the employed DLT, i.e., IOTA, we conclude that at the moment of the test execution, the results were not viable for real-time scenarios but acceptable for less demanding services. Tests show a latency between 23 and 27 seconds for 0.75 to 1.5 MAM messages insert requests per second, with, at best, an experienced tps, i.e., transactions per second, of 0.13 (considering 1 MAM message roughly equal to 3 IOTA transactions). It means that, for latency on average of $\sim$ 25 seconds, with a configuration similar to ours during tests, available IOTA nodes in such a scenario should scale following $\#nodes \geq \frac{k \times \#requests}{sec}$, with $k = 53$. The $\#requests$ is the number of MAM messages insert requests generated by the users in our scenario. Hypothetically, having a DLT protocol that allows $tps = 1$ would require having available IOTA nodes in such a scenario that scale following the same formula but with $k = 2$.

Clearly enough, an adequate infrastructure that supports general-purpose PDS must be well set in all cases to build a scalable architecture that can handle a possibly high data generation rate properly. In other words, we think the issue is more concerned with the system deployment than the DLT/DFS protocol. For instance, an edge-computing architecture can be merged with the framework and used to geographically place node gateways, which receive data from users and insert them into DLT/DFS.

4.5 Conclusions

In this Chapter, we presented the architecture for a personal data space based on the use of distributed ledger technologies (DLTs) and decentralized file storage. These two kinds of technologies provide a resourceful environment for interactions between data subjects, holders, and recipients. The rationale was to provide individuals with a tool to effectively exercise (at least part of) their rights, as in the General Data Protection Regulation (GDPR), and to enable data sharing as the newly proposed European Data Governance Act intended. We analyzed the tensions between the GDPR and DLTs and, in the light of Self-Sovereign Identity, we introduced three architectural components: (i) a personal device application, (ii) a (D)FS, and (iii) a DLT. A PDS based on centralized or decentralized file storage enables data persistence and a place where data can be shared after being protected through a hybrid encryption scheme that includes a key encapsulation mechanism (KEM) and a data encapsulation mechanism (DEM). The DLT brings the immutability and transparency features to the whole process. Furthermore, we provide a prototype implementation developing the DLT component as an IOTA public DLT and leveraging IPFS and Sia as DFS components. At first, we discussed their qualitative differences; then, we compared them experimentally in terms of execution time, taking an urban public service scenario as a use case. Results from our performance evaluation show that: (i) up to a certain overload, a proprietary service, where a dedicated node is in charge of running an IPFS node, appears to provide stronger assurances for responsiveness and reliability; (ii) as concerns the employed DLT, i.e. IOTA, we conclude that at the moment, the obtained results are not viable for real-time applications but acceptable for less demanding services.

4.5.1 DLT-agnostic considerations

Our implementation of the PDS is based on the use of IOTA as DLT and IPFS and Sia as DFS. However, the architecture presented in this chapter can be considered agnostic to the DLT (or DFS) chosen. In fact, any DLT that supports the issuing of transactions that include arbitrary data (i.e., the hash pointer) can be used. In our implementation we refer to the IOTA DLT for its features, but it is possible to implement the same architecture in Bitcoin or Ethereum, for example. The disadvantage for both would be a loss of performance, as issuing a transaction for a block takes longer. The same is true for DFS, and we have demonstrated this using two different configurations. It is possible to use a traditional FS, such as a Cloud storage service, as long as some API can be used to upload and download the data. Then the unique and immutable URI, such as the CID of IPFS, must be created on purpose if not already present.

Chapter 5

Decentralized Indexing

The content of this chapter is based on the contributions published here:

- M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, *"Complex Queries Over Decentralised Systems for Geodata Retrieval,"* IET Networks, pp. 1–16. Wiley, 2022.
- M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, *"Towards Decentralized Complex Queries over Distributed Ledgers : a Data Marketplace Use-case,"* in Proc. of the 30th IEEE International Conference on Computer Communications and Networks (ICCCN), pp. 1–6, IEEE, 2021.

The software produced during the development of this chapter is stored here:

- C. Giansante, and M. Zichichi (2021). Hypercube DHT Simulation. DOI: `10.5281/zenodo.6548266`
- F. La Piana, A. Leurini, D. Tropea, and M. Zichichi (2021). Hypercube DHT implementation ans vehicular scenario. DOI: `10.5281/zenodo.5810396`

This Chapter acts as a digression concerning the description of a system for the protection and portability of personal data. However, it is fundamental to tackle a critical issue of decentralized systems that, generally, is not fully considered.

As we have seen in Chapter 4, benefits often cited of the inclusion of DLTs and DFS in a system include the enabling of high data availability, integrity, authenticity, and auditability; all aspects needed to build novel applications for a "more" decentralized Internet (Belotti et al., 2019). One of the concerns that are still open for these novel technologies, is related to the data discovery and lookup operations. Data inserted in DLTs and DFS are usually unstructured, and no efficient mechanisms are available to query certain kinds of information, such as data generated by sensors in a given geographical area. Thus, data lookup can be prolonged and expensive, even if anyone can run public DLTs and DFS nodes, such as IOTA and IPFS. Data are rarely stored in a format that can be consumed directly, and they need to be filtered and indexed before executing any complex query. Data are referenced through addresses or indexes that, most of the time, are not related to the content of the data and thus are not helpful for their categorization.

Consider again the example of personal data made in Sub-Section 4.3.1 of the previous Chapter, i.e., *data* = (*Fri, 16 Sep 2022 09:15:35*, *Latitude: 63.1702*, *Longitude: 29.9086*). This piece of data, once (encrypted and then) uploaded on a PDS based on IPFS, can be retrieved using the following CID: `Qme1vfyukJJjVA7gpXExPkY49f3jghq7wYe42CXJmaSeiL`. Using the approach described in the previous Chapter, this CID, which represents the hash of the (encrypted) piece of data, is stored in an IOTA transaction that, in turn, will result in a message ID similar to this: `6c93b210f625c06987b429a2a56dc7a816a91ee7da24ca226e6a0acdf4f7806c`. Now, both IDs can be seen as what we intend for website URLs, e.g., `http://www.websiteXYZ.com`, since all of them are unique strings of text used to retrieve content: a webpage in the case of the URL, a transaction in the case of the IOTA message ID, and a piece of data in the case of the IPFS CID. However, in the case of the URL, we often have a meaningful way of **indexing a content**, i.e., if I can interpret `websiteXYZ` as the name of the company or product that is related to the content (webpage) associated to the URL. While on the other hand, I cannot interpret in any way the IOTA message ID, and I can use the IPFS CID only to verify the content integrity[3]. Specifically, data can only be accessed by knowing their respective identifier or location and cannot be searched based on the

[3] The IPFS CID supports too InterPlanetary Linked Data formats, enabling decentralized data structures to be universally addressable and linkable and thus enhancing the expressiveness of the CID itself (IPLD Team, 2016). However, we reserve this discussion for Chapter 8.

specific content. In other words, these systems lack a viable and decentralized data management scheme that enables the efficient execution of “complex” and meaningful queries.

Currently, the most spread but ill-posed solution is to resort to the use of centralized ”explorers” that re-arrange DLTs and DFS data to create databases dedicated only to data indexing and queries (Blockchain.com, 2020). It nullifies all the benefits of decentralization seen in Chapter 2, as far as information retrieval is concerned. One example is that DFS and, in general, other P2P-based technologies also have a prominent role against censorship since shutting down a peer node will not prevent contents from being available on the Internet due to the P2P mechanism of content replication. A relevant example is the shutdown of Wikipedia that happened in 2017 in certain countries, while the IPFS could still guarantee access through the mirroring of the website (Santos et al., 2019). Thus, a decentralized indexing and query mechanism is also needed to overcome the weaknesses of the single point of failure and arbitrary control.

In this Chapter, we propose a decentralized system for key-value metadata-based lookup, which allows retrieving contents stored in DLTs and/or DFS. Of course, in the context of a PDS, this decentralized mechanism is needed by data recipients to lookup for specific kinds of personal data. Our approach relies on a Distributed Hash Table (DHT) as a layer placed on top of the DLTs/DFS, which offers the possibility to perform multiple keyword searches. The DHT has a P2P hypercube overlay structure (Joung et al., 2007). Each domain of the hypercube structure corresponds to a keyword, and specific DHT nodes are responsible for a particular portion of the keywords space. Navigating inside the hypercube makes it possible to query the node that maintains information about contents stored in the DLT/DFS with specific keywords as metadata. The hypercube DHT maintains an association between a particular keyword set and, possibly, the ID of a message in IOTA or a CID in IPFS. Our approach is independent of the underlying DLT or DFS and can be easily extended to other decentralized technologies.

In the following, the original contributions and novelties of this Chapter are described:

- First, we describe the design of such a hypercube-based DHT architecture on top of DLTs and/or DFS. In particular, the hypercube is a logical layout in which there are 2^r nodes, each labeled with a r-bit string identifier (rID) and connected to the r nodes whose rID differs by only one bit (Joung et al., 2007). Each node is responsible for a specific keyword set derived from their rID. Even if our approach is agnostic for the underlying technology, we provide a particular design of a tailor-made system for IOTA.

- Second, we report an experimental validation of the implementation of the proposed hypercube DHT architecture. In particular, we provide results showing how the size of the hypercube and the number of objects stored in the DHT affect the search procedures. Results confirm that our system allows for multiple keyword searches in a reasonable time, i.e., in the order of the logarithm of the number of hypercube DHT nodes.

The remainder of this paper is structured as follows. Section 5.1 provides the background and related works. In Section 5.2, we present the system architecture for the decentralized indexing system based on the hypercube DHT. In Section 5.4, the implementation of the system together with its validation and an experimental evaluation is provided. In Section 5.5, we then provide the final remarks.

5.1 Background and Related Work

In this Section, we introduce background notions needed for the rest of the paper, and then we discuss the related works.

5.1.1 Distributed Hash Table (DHT)

A Distributed Hash Table (DHT) is a decentralized system for the distributed storage of contents that provides the functionalities of a hash table, i.e., a data structure that efficiently maps "keys" into "values". The rationale of this approach is to store the information in the various nodes of the system, providing a routing mechanism to efficiently get which node owns a certain resource (Joung et al., 2007). Each local view of the DHT nodes will look like a traditional hash table, mapping from a key (i.e., the univocal representation of an item) to values (i.e., addresses of the peers owning such a resource). In addition, each node stores a partial view of the entire network, with which it communicates routing information. A routing procedure typically traverses several nodes, approaching the destination at each hop to reach nodes from one part of the network to another.

The association of objects to DHT nodes is obtained through a one-way function (e.g., hash function) that maps any item into a binary sequence of n bits. The idea is to distribute the storage workload among the DHT nodes according to the objects' key, i.e., the n bit string obtained after applying the function. Each DHT node is identified through an n bit ID, which lies in the same ID space used to identify contents. Then, based on its ID, each node is in charge of maintaining information on those contents in a specific ID space interval. The lookup of a content x thus becomes looking for the node in the DHT that manages a subset of the ID space that contains x (D'Angelo and Ferretti, 2017).

This type of infrastructure has been used as a key element to implement complex and decentralized services, such as Content-Addressable Networks (CANs) (Ratnasamy et al., 2001), DFS (Benet, 2014), cooperative web caching, multicast, and domain name services.

5.1.2 Related Works

The decentralized data search on DLT and DFS is a field that scholars and developers have recently addressed. The Graph is one of the first protocols to provide a "Decentralized Query Protocol" (The Graph, 2020). The Graph network consists of a system built upon Ethereum and IPFS, allowing users to query data stored using these two technologies. The Graph users can query several indexers nodes by paying for their metered usage within a query market. The organization of the network is similar to what is referred to as DAO. However, their method for storing indexes is different from our proposal. Indeed, instead of using a DHT network to store data, the Graph P2P network is based on a Service Addressable Network used to locate nodes capable of providing a particular service, which can be any arbitrary computational work.

Specifically for IPFS, a generic search engine has been developed to overcome the file search limitation, namely "ipfs-search" (IPFS Community, 2021). This solution is centralized and does not escape the problem of concentration, similar to the conventional web. In response to this, a decentralized solution called Siva (Khudhur and Fujita, 2019) has been proposed. An inverted index of keywords is built for the published content on IPFS, and users can search through it. However, Siva is offered as an

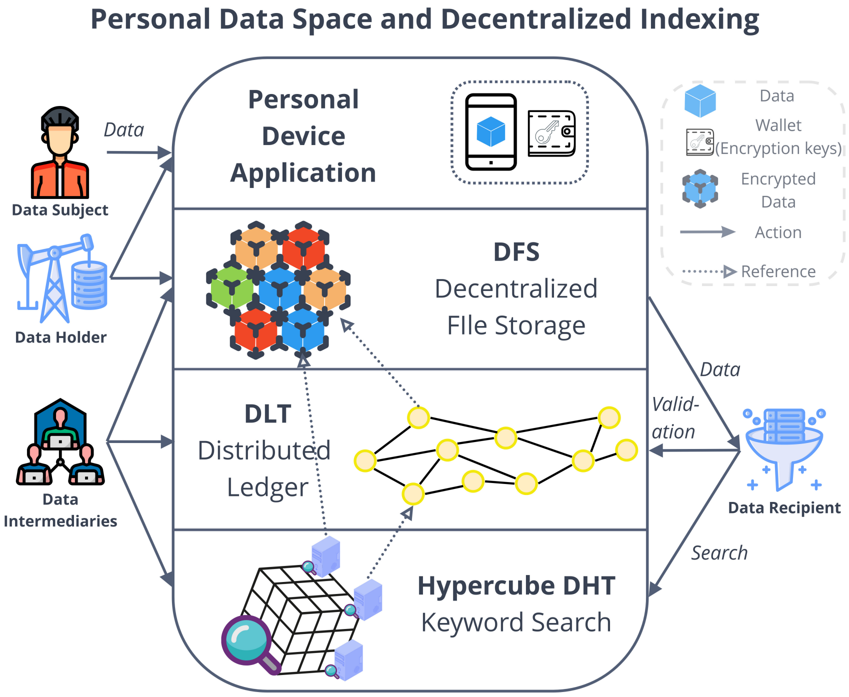


Figure 5.1: Diagram showing a layered view of the Decentralized Indexing architecture

enhancement of the IPFS public network DHT and does not feature any optimization for a keyword storage structure apart from caching. In (Jiang et al., 2020), the authors propose a layer-one keyword search scheme that implements oblivious keyword search in DFS. Their protocol is based on keywords search with authorization for maintaining privacy with retrieval requests stored as a transaction in a blockchain (i.e., layer-one). Finally, a layer-two solution for keyword search in DFS has been proposed in (Zhu et al., 2020), where a combination of a decentralized B+Tree and HashMaps is used to index IPFS Objects (Benet, 2014).

5.2 Decentralized Indexing Architecture

In this Section, we will focus on the description of the architecture for the decentralized indexing of DLT and DFS data. Such a decentralized system design allows for queries based on multiple keyword searches. We refer to it as the hypercube DHT. This solution can be leveraged in several cases and with different technologies, especially when decentralization is required. It is essential to emphasize that the hypercube DHT solution is agnostic to any underlying system. The only requirement the underlying system must have is to use a unique ID to refer to specific content. The hypercube DHT represents "just" the means for binding specific keywords to a specific ID in a decentralized fashion.

5.2.1 Decentralized Indexing in the Personal Data Space

Before going into the detail of the hypercube DHT specification, we also explain its interoperability with other decentralized systems, such as DLTs and DFS, that are the ones that build up the PDS (see Chapter 4).

Figure 5.1 shows a layered view of the decentralized indexing architecture on top of our proposed PDS. It is based on the following components:

- the **Personal Device Application** continues to be the data subject, holder and recipient interface to the various components.
- the **DFS component** continues to be used to store data in an encrypted form and can be seen as a layer-one technology because it does not depend on any other technology;
- the **DLT component** continues to provide a ledger for the first part of the data indexing and validation in the form of hash pointers, but where these pointers have no meaning; it too can be seen as a layer-one technology;
- the **Hypercube DHT** is a layer-two technology that depends on the previous ones, i.e., a mapping of DLT and DFS IDs to keywords is the link between the layer-one and layer-two; the hypercube DHT stores these keywords, and provides a distributed mechanism for the search of data.

The link between the hypercube DHT layer-two and the DLT and DFS layer-one can happen in three instances:

Only DFS: Taking IPFS as a DFS example, an IPFS Object, e.g., a file, is uniquely identified through a CID; when a piece of (encrypted) personal data is uploaded to IPFS, then the hypercube DHT stores the mapping between the resulting CID and some meaningful keywords related to the piece of personal data.

Only DLT: Taking IOTA as a DLT example, a transaction or IOTA message is uniquely identified through an ID; when some data is stored in the Tangle, then the hypercube DHT stores the mapping between the resulting message ID and some meaningful keywords related to the data.

DLT + DFS: When a piece of (encrypted) personal data is uploaded to IPFS, and the resulting CID is stored in the Tangle (as a hash pointer), then the hypercube DHT stores the mapping between the resulting message ID and some meaningful keywords related to the piece of personal data.

From now on, we will only refer to the $dixID$ as a generic ID related to one of the three above cases (such as the message ID and CID shown in this Chapter's introduction). The $dixID$, when used in the specific DFS or DLT, will resolve to particular content, e.g., an IPFS Object, an IOTA message, an IOTA message containing a hash pointers (CID), etc. Thus, the hypercube DHT maps a $dixID$ to a keyword set, where the keyword refers to the content resolved by the $dixID$.

5.3 Hypercube DHT Component Design

In this Section, a description of the main component of the architecture will be provided. The structure of the Hypercube DHT will be described, followed by the specification of the query protocol.

5.3.1 Hypercube Structure

Considering D as the set of all $dixID$ possible, the idea is to map each $dixID \in D$ to a keyword set $K_{dixID} \subseteq W$, where W is the keywords space, i.e., the set of all keywords considered. Thus, in general, a keyword set $K \subseteq W$ can be associated with content (i.e., the data related to it) or a query (i.e., we are looking for some content associated with specific data). By using a uniform hash function $h : W \rightarrow \{0, 1, \ldots, r-1\}$, a keyword set K can be represented by the result of such function, i.e., a string of bits u where the 1s are set in the positions given by $one(u) = \{mod_r(h(k)) \mid k \in K\}$. In other words, each $k \in W$ has a fixed position in the r-bit string given by $h(k)$, and that position can be associated with more than one k (i.e., hash collision). Then, every keyword set K is represented by a r-bit string where the positions are "activated", i.e., are set to 1, by all the $k \in K$.

We use these r-bit strings, rID, to identify logical nodes in a DHT network, e.g., for $r = 4$ a node rID can take values such as 0100 or 1110. Inspired by (Joung et al., 2007), we refer to the geometry of the hypercube to organize the topological structure of such a DHT network. $H_r(V, E)$ is a r-dimensional hypercube, with a set of vertices V and a set of edges E connecting them. Each 2^r vertices represents a logical node, while edges are formed when two vertices rID differ by only one bit, e.g., 1011 and 1010 share an edge. In the DHT, the network node represented by a vertex $rID = u$ is directly connected, i.e., neighbor, to a node represented by a vertex $rID = v$ that shares an edge with u. The Hamming distance is be used to find out how far apart the two vertices u and v are within the hypercube, i.e., $Hamming(u, v) = \sum_{i=0}^{r-1}(u_i \oplus v_i)$, where $\oplus$ is the XOR operation and u_i is the bit at the i-th position of the u string, e.g., for $u = 1011$ and $v = 1010$, we have $Hamming(u, v) = 1$.

5.3.2 Keyword-based Complex Queries

In the hypercube DHT, contents can be discovered through queries based on the lookup of multiple keywords associated with the $dixID$. Such queries are processed by exploiting the indexing scheme described in the previous Section. Assuming that we have a keyword space $W = \{$*"Bologna", "San Donato", "Temperature", "Celsius"*$\}$, we can then elaborate an rID from a keyword set containing one or more of these keywords. For instance, the keyword set $K_{dixID} = \{$*"Bologna, Temperature"*$\}$ could represent the content indexed in IPFS through the CID equal to $dixID =$ `Qme1vfyukJJjVA7gpXExPkY49f3jghq7wYe42CXJmaSeiL`. Assuming that $r = 4$, $mod_r(h(\text{"Bologna"})) = 0$ and $mod_r(h(\text{"Temperature"})) = 2$, then, the resulting rID will be equal to $rID = 1010$. Let say that $u \in V$ is the node with $rID = 1010$; then u is responsible for K_{dixID} and it maintains a list associating K_{dixID} to the actual CID $dixID$, but also to all the other CIDs that are represented by the same keywords *"Bologna"* and *"Temperature"*, i.e., data representing temperatures measured in the city of Bologna. When a user decides to perform a query to the hypercube DHT based on the keyword set containing the two keywords, the system will reach node u, responsible for that keyword set, in a mechanism described in the following two Sub-Sections.

5.3.2.1 Multiple Keywords Search

The system that we propose provides two functions for performing queries based on multiple keywords:

- **Pin Search** - this procedure aims at obtaining all and only the $dixID$s associated exactly with a keyword set K, i.e., $\{dixID \in D \mid K_{dixID} = K\}$. Upon request, the responsible node returns all the $dixID$s that it keeps in its table (i.e., internal storage) and associated with K to the requester.

- **Superset Search** - this procedure is similar to the previous one, but in addition, it also searches for $dixID$s that can be described by keyword sets that include K, i.e., $\{dixID \in D \mid K_{dixID} \supseteq K\}$. Since the possible outcomes of this search can be quite large, a limit l is set to the number of returned results. Upon request, the responsible node returns to the requester all the $dixID$s that it keeps in its table and the $dixID$s that its neighbors (or the neighbors of its neighbors, and so on) keep in their table, associated with K.

In the Pin Search, the system needs to retrieve $dixID$s only from one node. While, for Superset Search, the system needs to retrieve $dixID$s from all the nodes responsible for a superset of K. Such nodes are contained in the sub-hypercube $SH(S, F)$ induced by the node u responsible for K, where S includes all the nodes $s \in V$ that "contain" u, i.e., $u_i = 1 \Rightarrow w_i = 1$, while F includes all the edges $e \in E$ between such nodes. Thus, during a Superset Search, the induced sub-hypercube is computed, and then only nodes in such sub-hypercube are queried using a spanning binomial tree as described in (Joung et al., 2007) (definition 4.2). More specifically, the l limit is a query parameter that indicates the maximum number of $dixID$s to return when traversing the spanning binomial tree.

5.3.2.2 The Query Routing Mechanism

Users can inject queries into the system external to the DHT to any $v \in V$ DHT node. Through a routing mechanism, a query q with a defined keyword set k will reach a node $u \in V$ that is responsible for that keyword set. If q is of type Superset Search, then the query will reach all the nodes accountable for keyword sets that include K until the limit of $dixID$s l is reached. This process is described in detail in Algorithm 1. This

Algorithm 1: QueryRoutingMechanism

```
Input: $q$ query, $K$ keyword set, $l$ limit
Global Data: $rID(v), one(v), neighbors(v)$
Result: $\{dixID \in D \mid K_o \supseteq K\}$
$one(u) \leftarrow \{h(k) \mid k \in K\}$
$rID(u) \leftarrow GetRIDFromOnes(one(u))$
if $rID(u) \neq rID(v) \wedge From(q) = $ “User“ then
    $w \leftarrow \{n \mid n \in neighbors(v) \wedge \text{Min}(\text{Hamming}(n, u))\}$
    return QueryRoutingMechanism($w, q, K, l$)
else
    if $Type(q) = $ “PinSearch“ then
        return GetDixIDsFromIndexTable($K$, $-1$)
    else if $one(u) \subseteq one(v)$ then  // i.e., SupersetSearch
        dixIDsList ← GetDixIDsFromIndexTable($K$, $l$)
        $l \leftarrow l-$ Length(dixIDsList)
        From($q$) ← “Node“
        while $l > 0$ do
            $c \leftarrow$ GetNextSBTreeChild($u$)
            cList ← QueryRoutingMechanism($c, q, K, l$)
            dixIDsList ← dixIDsList + cList
            $l \leftarrow l-$ Length(cList)
        end
        return dixIDsList
    end
end
```

algorithm is executed by each node every time it gets a new query from a neighbor or a user. The routing mechanism from a node $v \in V$ receiving a query q from a user, with a keyword, set K and a limit l is as follows:

1. if node v is not responsible for K, i.e., $\{h(k) \mid k \in K\} = one(u) \neq one(v)$, then it computes, for all its neighbor nodes, the Hamming distance to node u;

2. node v broadcasts the query q to the neighbor w with the lowest distance to u;

3. these two steps are repeated by w and by the subsequent nodes until the query q reaches u;

4. if the query q is of type Pin Search u returns the $dixID$s associated with K, i.e., $\{o \in O \mid K_o = K\}$;

5. else (in the case of a Superset Search) u computes its children nodes in the spanning binomial tree of the induced sub-hypercube;
6. then node u broadcasts q to the children nodes until the limit of $dixID$s l is reached;
7. the children nodes will repeat the process from step 5 with their children and then return the $dixID$s;
8. finally node u returns the aggregated $dixID$s associated with different supersets of K, i.e., $\{o \in O \mid K_o \supseteq K\}$.

5.4 Implementation and Evaluation

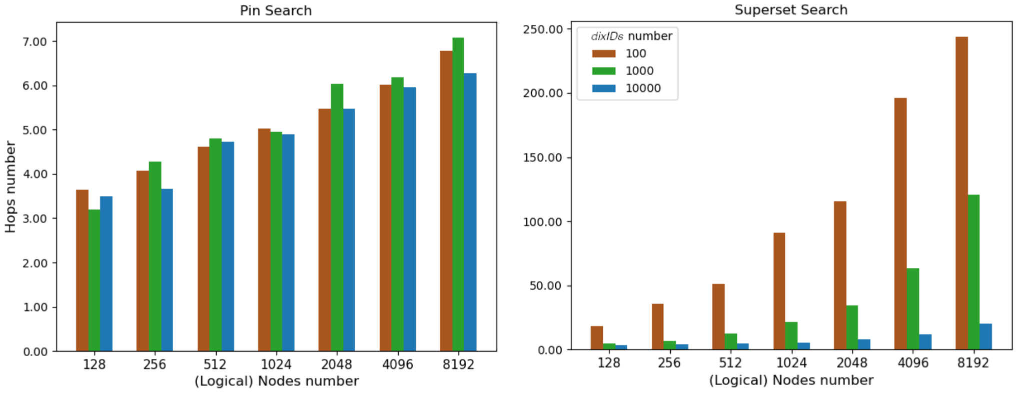
In this Section, we are interested in describing how the implementation of the decentralized indexing performs, particularly regarding the hypercube DHT. Because the network underlying the hypercube DHT can arbitrarily grow in size, we will first validate the query mechanisms through a simulation of a large P2P network implementing our proposal. After that, we will focus on the performance evaluation related to the responsiveness and reliability of our actual software implementation of a hypercube DHT node based on simulated user interactions generated from a vehicular scenario.

5.4.1 Hypercube DHT Validation

To validate the efficiency of the routing mechanism, we conducted a simulation assessment using PeerSim, a simulation environment developed to build P2P networks using extensible and pluggable components (D'Angelo and Ferretti, 2011; Montresor and Jelasity, 2009). The simulation implementation and the results data can be found as open source code in (Giansante and Zichichi, 2021).

To validate the Pin Search and Superset Search mechanisms, we tested different sizes of the hypercube DHT. Specifically, the number of nodes varied from 128 ($r = 7$) up to 8192 ($r = 13$). Then, for each dimension r, a different number of randomly created keywords-$dixID$ couple was inserted in the DHT. The number of $dixID$ taken into consideration varies in the range 100, 1 000 and 10 000.

Given that it consists of a simulated test, we considered the number of hops required for each new query as a parameter to be evaluated. A hop occurs when a query message is passed from one DHT node to the next. The query keyword sets were randomly generated, and the starting node was randomly chosen. For each type of test, 50 repetitions were performed, and then the average results were calculated. The limit value was set for the Superset search to $l = 10$.



Number of hops on average when variating the number of nodes and keywords-dixID couples for the Pin Search (left) and Superset Search (right).

Figure 5.2: Histogram showing the average number of hops for a query.

Table 5.1: Pin Search Number of Hops.

# of nodes	Average			Standard Deviation			Confidence Interval (95%)		
	100	1000	10000	100	1000	10000	100	1000	10000
128	3.64	3.2	3.5	1.33	1.32	1.12	(3.2, 4.0)	(2.8, 3.5)	(3.1, 3.8)
256	4.08	4.28	3.66	1.45	1.48	1.31	(3.6, 4.4)	(3.8, 4.6)	(3.2, 4.0)
512	4.62	4.8	4.72	1.57	1.70	1.24	(4.1, 5.0)	(4.3, 5.2)	(4.3, 5.0)
1024	5.02	4.96	4.9	1.68	1.67	1.69	(4.5, 5.4)	(4.4, 5.4)	(4.4, 5.3)
2048	5.48	6.04	5.48	1.76	1.85	1.69	(4.9, 5.9)	(5.5, 6.5)	(5.0, 5.9)
4096	6.02	6.18	5.96	1.55	1.61	1.62	(5.5, 6.4)	(5.7, 6.6)	(5.5, 6.4)
8192	6.78	7.08	6.28	1.63	1.60	1.64	(6.3, 7.2)	(6.6, 7.5)	(5.8, 6.7)

Table 5.2: Superset Search Number of Hops.

# of nodes	**Average**			**Standard Deviation**			**Confidence Interval (95%)**		
	100	1000	10000	100	1000	10000	100	1000	10000
128	18.28	4.54	3.52	8.44	1.54	1.19	(15.9,20.6)	(4.1, 4.9)	(3.1, 3.8)
256	35.90	6.80	4.16	17.89	2.25	1.43	(30.9,40.8)	(6.1, 7.4)	(3.7, 4.5)
512	51.18	12.16	4.46	37.85	3.29	1.31	(40.6,61.6)	(11.2,13.0)	(4.1, 4.8)
1024	91.06	21.70	5.08	72.44	6.23	1.68	(70, 111)	(19.9,23.4)	(4.6,5.5)
2048	115.70	34.56	7.84	98.39	13.00	1.98	(88, 142)	(30.9,38.1)	(7.2,8.3)
4096	196.00	63.38	11.92	186.88	25.37	2.64	(144, 247)	(56.3,70.4)	(11.1,12.6)
8192	243.90	120.38	20.38	253.59	68.65	6.28	(173, 314)	(101, 139)	(18.6,22.1)

5.4.1.1 Results

The Pin Search tests results are shown in Table 5.1 and Figure 5.2 (left), while Superset Search ones are shown in Table 5.2 and Figure 5.2 (right). Overall, the number of hops required to transmit a message from the source node to the destination node increases as the hypercube DHT dimension r increases.

In the case of the Pin Search, the average number of hops increases from about 3.5 for 128 nodes ($r = 7$) to about 6.72 for 8192 nodes ($r = 13$). The number of $dixID$s in the testbed does not affect the final outcome, since the path to reach the target node only follows the rationale of the hypercube DHT and does not depend on the number of keywords-$dixID$ associations stored in the DHT.

In the case of the Superset Search, anomalous values stand out corresponding to a high number of hops between nodes with lower $dixID$s number. It can be explained by the fact that the Superset Search traverses the hypercube DHT nodes until it finds the number of $dixID$s indicated by the limit, i.e., $l = 10$. In a network with many nodes and few $dixID$s, the query might take longer to reach the l limit, because many nodes are "empty", i.e., have no $dixID$ to return. For configurations where $dixID$s are uniformly distributed, e.g., 4096 nodes and 10000 $dixID$s, then the Superset Search has similar results to the Pin Search, and the difference depends on the l value. Indeed, the first 5.96 hops, on average are required to reach the responsible node (i.e., same as the Pin Search), then other $11.92 - 5.96 = 5.96$ hops are needed to reach other nodes to reach the l limit.

5.4.1.2 Discussion

The results obtained validate our hypercube DHT because they reach the theoretical efficiency of a hypercube structure. Indeed, the Pin Search number of hops is of the order of the logarithm of the hypercube nodes number, i.e., $\log(n) = r$. In particular, on average they are equal to $\frac{\log(n)}{2} = \frac{r}{2}$. For what concerns the Superset Search number of hops, on average, it is in the order of the $\frac{\log(n)}{2} + \pi(l,u)$, where $\pi(l,u)$ is the average number of hops required to obtain a number l of $dixID$s from u's neighbors.

These results show the goodness of the solution, in theory, reaches a balanced trade-off between memory space and response time. Especially in the DLTs case, where searching for a piece of data in a transaction means traversing all the ledger "transaction sea", this is a substantial improvement.

5.4.2 Implementation and Evaluation

After having validated our hypercube DHT proposal, we implemented the whole decentralized indexing architecture and evaluated its performance.

5.4.2.1 Implementation

The implementation of the components of our proposed decentralized indexing consists of the following:

- The **personal device application** has been developed again as a module that was used for simulating several instances of devices that interact with the PDS and the hypercube DHT. We performed another simulation similar to the one present in Chapter 4 based on a scenario where personal data is uploaded voluntarily by smartphone users to the PDS and hypercube DHT. The scenario consists of a road hazard detection application. While driving, users upload the data gathered from their smartphones' gyroscope, accelerometer, and GPS to detect and measure the vibrations in potholes and bumps. The geodata related to the position where the hazard is recorded and other data points are uploaded to the PDS, and the IOTA message id, i.e., the $dixID$, retrieved after entering the data in the Tangle, is inserted into the hypercube DHT. These keywords are determined by the encoding process based on the geoposition related to the data:

  - the first encoding consists in converting the latitude and longitude of the geoposition data into *Open Location Code (OLC)* (Rinckes and Bunge, 2015). This code represents areas in the world with a precision of 3.5 squared meters (when the code has a length of 11 characters). The fewer digits, the larger the squared area, and vice versa. For instance, a 4-digit code such as $6P23$ identifies a particular squared area with a side of 110 km.
  - the second encoding is a transformation of an OLC to an rID. For example, given the OLC "6P0000" and $r = 6$, each or a contingent group of characters, depending on the granularity chosen, is associated with a specific position in the rID: "6" with position 4 and "P" with position 5. Then "6P0000+" would be converted to "000011". Since the code "6PH57VP3" contains the code "6P0000+" (while for the OLC logic the latter represents an area that contains the area represented by the former), then the rID representing the former code must include the "000011" representing the latter, i.e., a rID such as "101011" because $one(101011) \supset one(000011)$.

  The implementation can be found in (La Piana et al., 2021).

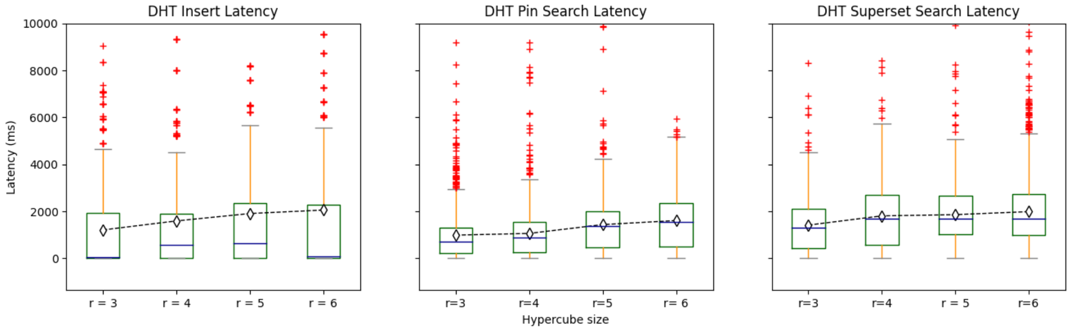
- The **DFS component** is the same as the one implemented in Chapter 4.
- The **DLT component** is the same as the one implemented in Chapter 4.
- The **Hypercube DHT component** is implemented as software that each node runs for maintaining the index table and answering the queries it receives. The source code is written in Python, and it implements a network peer node exposing the four main actions employing the Flask server framework (Grinberg, 2018), i.e., Insert object, Remove object, Pin Search, Superset Search. The implementation can be found in (La Piana et al., 2021).

5.4.2.2 Evaluation

To test the inserting and retrieving of data to and from the decentralized indexing system, we simulated a vehicular scenario in which road hazard detection was performed through the implemented system. The vehicular environment was simulated to generate data and queries to the rest of the system deployed and executed in a real environment, i.e., a dedicated host with a quad-core Intel Core i7 CPU and 16GB RAM.

In all tests configuration, the unique physical host ran the software for 8, 16, 32, and 64 different logical nodes and populated the network each time with $dixID$ generated following the simulation.

The simulation contained a geographical area with different routes, 10 vehicles for each route with a starting point, a destination and a path to follow, and a set of road hazards (e.g., potholes or obstacles) randomly placed in the routes. In the insert phase, each vehicle that encounters a road hazard uploads the data to the PDS and the obtained $dixID$ to the hypercube DHT associated with the current geoposition keyword. The retrieve phase is concurrent to the previous one and consists of periodically querying the system to query for road hazards (every 3 minute). Given a specific location, the system must report all the registered road faults within that area. This is possible thanks to the use of the Superset Search.



*Results are reported as box plots and with a line plot where the diamond represents the mean value of the overall latency. The rectangle identifies the Inter-Quartile Range (IQR), i.e., values from the 25th to the 75th percentile, representing the middle 50% of values. Hence, the lower part of the box (let denote it Q1) is the first quartile (25th percentile), and the highest (denote it Q3) is the third quartile (75th percentile). The blue line inside the box is the median value. The lower and upper values identified by the vertical line are the whiskers. In box plots, the whiskers are defined as 1.5 times the IQR. Thus, the lower whisker is Q1 - 1.5*IQR, while the upper whisker is Q3 + 1.5*IQR; they represent a common way to describe the dispersion of the data. Finally, the red "×" symbols outside the whiskers are the outliers.*

Figure 5.3: Box plot representing the DHT's Insert, Pin, and Superset Search operations latency.

Results The process of signaling hazards to the system involves two operations performed in sequence. The first consists of uploading the data to the DLT and DFS components of the PDS. We omit the evaluation of this operation as the previous

Chapter (see Section 4.4 in Chapter 4) already gave us an idea of the performance of different DFS implementations and IOTA. The second operation consists in inserting the geoposition keywords-$dixID$ association in the hypercube DHT. The process of retrieving the hazards from the system involves the two specular operations performed in sequence, i.e., searching the data through the hypercube DHT by using geoposition keywords and then fetching the data using the obtained $dixID$s.

- *Hypercube DHT Insert*: Figure 5.3 (left) shows how the average time for inserting the object in the DHT grows as a function of the parameter r that determines the size of the hypercube DHT. the average insertion time can be considered low as it varies from a few more than 1 second, with $r = 3$ (8 nodes), to a few more than 2 seconds with $r = 6$ (64 nodes). A second aspect concerns the presence of outliers in all the simulations carried out; in some cases, the insertion in the hypercube DHT requires longer than average. The presence of outliers, whose frequency increases as the size of the network increases, is partly due to the randomness factor in the choice of the node that receives the request. If the requested node might be very distant in terms of the Hamming distance from the node responsible for the geoposition keywords, it may take many nodes forwards before reaching it. It translates into an increase in latency.
- *Hypercube DHT Retrieve*: In Figure 5.3 (center and right), it can be seen that, when retrieving objects from the hypercube DHT, the increase in the network size has a relatively low impact on the overall latency. In fact, despite the number of nodes in both types of search, the average search time in a rather complex situation, such as the one assumed in the creation of the scenarios, is low. The Pin Search has a minimum latency value of 1 second for $r = 3$ with an increase to only $1,6$ seconds when r is doubled (Figure 5.3, center). At the same time, however, the results present much more outliers when compared to the Superset Search (Figure 5.3, right). The Superset Search latencies increase by $\sim$ 400 milliseconds concerning the Pin Search ones.

Discussion Results show that, as expected, the hypercube DHT on top of a DLT or DFS approach allows for fast identification of data that satisfies a given keywords-based query. In our implementation, the latency for executing the insert and retrieve operations in the hypercube DHT runs in sublinear time concerning r. It should be noted that the two types of search, i.e., Pin Search and Superset Search, have a slight difference in terms of average latency that is due to the Superset Search limit parameter l, which in this use case satisfies the geographical area related to the hazards and the geoposition of the user executing the query. In general, the hypercube DHT results show the goodness of the solution in the trade-off between memory space and response time, as expected after the validation.

5.5 Conclusions

In this Chapter, we proposed a system based on a hypercube DHT that provides an efficient decentralized indexing and content lookup through multiple keyword-based queries. This layer-two solution can be applied to different kinds (or combinations) of DLTs and DFS. In our proposal, we address the problems of PDS regarding efficient personal data lookup and the possibility of implementing complex queries without reinstating an element of centrality. We first show the design of such a decentralized system and its implementation; then, we offer a case study in which the proposed architecture is used for geodata storing and retrieval based on a road hazard detection scenario. Being r the hypercube dimension (a value in the order of the logarithm on the number of DHT nodes), on average $\frac{r}{2}$ number of hops (i.e., when a query message is passed from one DHT node to the next) are required for a Pin Search, i.e., a punctual search based on a specific keyword set. Regarding the Superset Search, i.e., a broader search based on a particular keyword set and its supersets, the number of hops depends on the ratio between the limit l assigned to the query and the distribution of objects between nodes.

5.5.1 DLT-agnostic considerations

Our implementation of the decentralized indexing is based once again on the use of IOTA as DLT. In this case too, the architecture presented in this chapter and the hypercube DHT can be considered agnostic to the DLT chosen. In fact, any DLT solution proposed up to the writing of this work, uses unique $dixID$ to represent a meaningful piece of the ledger, e.g., a transaction or a block. In Ethereum, for instance, a transaction $dixID = 0x875564682295f303bf...0399805ea24c3c525806b$ can include the relevant information and the *dixID* can be used in the hypercube DHT as we have shown in our experiments. The disadvantage again would be a loss of performance due to the Ethereum consensus mechanism.

Chapter 6

GDPR-compliant Multi DLT Architecture for Access Control

The content of this chapter is based on the contributions published here:

- M. Zichichi, S. Ferretti, G. D'Angelo, and V. Rodríguez-Doncel, *"Data Governance through a Multi-DLT Architecture in View of the GDPR,"* Cluster Computing, pp. 1–28. Springer Nature, 2022.
- M. Zichichi, S. Ferretti, and V. Rodríguez-Doncel, *"Decentralized Personal Data Marketplaces : How Participation in a DAO can Support the Production of Citizen-Generated Data,"* Sensors, pp. 1–31. MDPI, 2022.

The software produced during the development of this chapter is stored here:

- M. Zichichi, and J. Sparber (2021). Personal data decentralized access control tests. DOI: `10.5281/zenodo.4572552`
- M. Zichichi (2021). Rust implementation of the umbral threshold proxy re-encryption scheme. DOI: `10.5281/zenodo.6548260`
- M. Zichichi (2022). Implementation of a dao that governs the exchange of data with an aggregator. DOI: `10.5281/zenodo.6548262`

This Chapter complements the previous two by outlining a system that closes some of the shortcomings of the one described so far and adds an essential tool for reaching agreements in sharing personal data, namely smart contracts. We enter into the description of the architecture of a complete decentralized Personal Information Management Systems (PIMS) (see Section 2.3.2 in Chapter 2). As already anticipated, the PIMS is an instrument that enriches the PDS capability of storing and making available data by providing individuals with decisional power over all aspects of their data. The decentralized aspect paves the way toward data intermediation where data holders and processors are **constrained** to act precisely as the subject instructed, e.g., through consent. Decentralization inevitably necessitates some forms of cryptography to render decentralized systems secure. In (some) centralized systems, one can afford to lose some confidentiality because of the trust in the system maintainer or in the laws that it needs to comply with, e.g., "my latest blood test result document is not encrypted in my Google Drive storage". On the other hand, in decentralized systems, one is forced to not tolerate any lack of confidentiality because of the lacking trust in the system maintainer**s**. A decentralized system that places the user at the center (see Section 2.3 in Chapter 2) seems counterintuitive, but at the same time, if one is put at the center of an environment where no one can be trusted, comprehensive security (i.e., cryptography in this case) is vital. If, on the other hand, one (thinks he/she) trusts another entity, comprehensive security is scratched by the fact that this entity might "make one's life easier". It is our interpretation of the Web 3.0 idea, and its implementation is conveyed through cryptographical methods and tools described in this Chapter.

We refer again to the European strategy for data (European Commission, 2020), with regards to the urgent need to place individuals at the center of personal data management, and we focus on *relieving* the absence of technical instruments that make the exercise of one's rights simple and not excessively burdensome. In our architecture, on top of the PDS and decentralized indexing layers, the data intermediaries implement an authorization service to provide data access to potential data recipients. Access to the data stored on a PDS can be allowed by the data holder through smart contracts. These contracts are encoded instructions and data structures that maintain a record of eligible data recipients, i.e., those to whom to issue the keys needed to access the encrypted data. Data subjects are *relieved* in exercising their rights by the decentralized computation of their policies, which is inevitably secured by the cryptographical access control mechanisms. As seen in these few lines, smart contracts are central to this Chapter's development and the continuation of this work. Moreover, its use for these purposes is already being addressed in the EU Data Act Regulation proposal (European Commission, 2022) (Article 11): "*The data holder may apply appropriate technical protection measures, including smart contracts, to prevent unauthorized access to the data and to ensure compliance with Articles 5, 6, 9, and 10, as well as with the agreed contractual terms for making data available*". It gives us a (future) practical use of the system we will describe in the following. The Data Act lays down rules: (i) on making data generated by the use of a product or related service available to the user of that product or service; (ii) on the making data available by data holders to data recipients; (iii) on the making of data available by data holders to public sector bodies or Union institutions. Data holders and recipients are roles with the same meaning we have used up to now (described in Chapter 4). The regulation applies to a wide range of actors, but mainly to (i) manufacturers of products and suppliers of related services placed on the market in the EU, (ii) their users, (iii) other data processors or data holders making data available in the EU, (iv) their recipients, and (v) public sector bodies and Union institutions. A PIMS results as a suitable solution for what has been provisioned in Article 4, i.e., making their generated data to users continuously and in real-time through electronic means. Moreover, Article 11 states that the data holder may use smart contracts to prevent unauthorized access to the data and to ensure compliance with the agreed contractual terms for making data available. It is our aim in this Chapter and the next one (Chapter 7). Our proposed PIMS adds to the PDS and decentralized indexing the use of smart contracts to enable its users the features of data custody and traceability of personal data and a system for access authorization.

Finally, in this Chapter, we will focus on some aspects of the architecture of PDS and decentralized indexing that raised some concerns about their compliance with the GDPR. In particular, we will describe how to implement a permissioned network of data holders while maintaining comparable security to the permissionless version described so far, that is, based on the public permissionless networks of IPFS and IOTA. This is done by transforming the current DLT component into a multi-DLT system, achieving the security of decentralization, high throughput, and low latency. The Audit

DLT, i.e., mainchain, consists of an already existing public permissionless DLT (e.g., Ethereum or IOTA) that stores batch transaction validations of the authorization DLT, i.e., a (semi)-private permissioned sidechain executing the authorization mechanism.

The original contributions and novelties of this Chapter are described in the following:

- First, we describe a novel PIMS based on a multi-DLT GDPR-compliant design. We propose an extension of our PDS and decentralized indexing system with a component for the secure control of access to personal data. These components are aggregated through a novel multi-DLT system where a permissioned DLT provides the authorization mechanism, and a permissionless DLT provides security and auditability.
- Second, we provide an interdisciplinary analysis of technical and non-technical drivers for designing a GDPR-compliant decentralized PIMS that can be generalized to different systems handling personal data. Furthermore, we discuss our proposal's security and privacy properties based on a privacy impact assessment methodology.
- Third, we provide a prototype implementation of the described system, and we evaluate its performance by employing an experimental evaluation. More specifically, the implementation is based on an Ethereum private blockchain where we (i) test the distributed data access control execution and (ii) evaluate the implementation of the smart contracts in terms of gas usage.

The remainder of this Chapter is organized as follows. Section 6.1 presents the background concepts behind the proposed architecture and related works. Section 6.2 has the purpose of providing an overview of the PDS architecture we propose. In Section 6.3, we specify the architecture's components, then discuss its GDPR compliance. In Section 6.5, the implementation of the PDS is described and evaluated in terms of performance. Finally conclusions are presented in Section 6.6.

6.1 Background and Related Work

In this Section, we describe the technologies that will be used for building the proposed software architecture, and we introduce the decentralized PIMS-related literature.

6.1.0.1 Smart Contracts

An immutable set of instructions whose execution is calculated deterministically by all (or several, depending on the protocol) peers in the DLT network is embraced by the definition of the smart contract. Each node executing the instructions receives the same inputs and produces the same outputs, thanks to a shared protocol. Hence, these properties allow the issuer of a smart contract not to require the presence of a trusted human third-party validator to check the terms of an agreement (which is why the term contract is used). However, since it consists of executable code, the issuer must also be sure that the behavior implemented is correct (e.g., through code verification). In Ethereum (Buterin et al., 2013), the smart contract is a set of instructions and a state, where the latter is modified utilizing transactions that enclose data and references to the former. The state evolution is ultimately traced in the ledger. These contracts can be considered trustless based on the assumption that most participant nodes are honest and follow the Ethereum protocol. In particular, this protocol allows computing (*quasi-*)Turing-complete programs, i.e., smart contracts, capable of processing any type of calculation where steps are bounded. A price, measured in a unit called "gas", is associated with each smart contract execution, and a gas limit is imposed to avoid infinite computation (Buterin et al., 2013).

6.1.1 Cryptographic Access Control and Keys Distribution

Access control is the ability to regulate access to some resources by enforcing permissions established by a set of policies, e.g., discretionary, mandatory, role-based, attribute-based (Maesa et al., 2019). In cryptographic access control (Kayem et al., 2010), policy enforcement depends on the security of the underlying cryptographic primitives and appropriate key distribution. Centralized control of data accesses, conveyed through a central access control server and keys distributor, entails the risks of a single point of failure and, above all, privacy leakages (Jemel and Serhrouchni, 2017). On the other hand, cryptographic access control paradigms built around secret sharing or proxy re-encryption can offer a better guarantee of privacy and security in the key distribution. It is obtained through proper decentralized key exchange mechanisms, as described in the following Sub-Sections and in Section 6.3.2.

6.1.1.1 Secret Sharing

Secret sharing (SS) was first proposed in (Shamir, 1979) and (Blakley, 1979) by Shamir and Blakley. It consists of a threshold scheme (t, n) in which each participant in a set of n participants owns 1 of the n shares of a secret, and any subset of $t \leq n$ participants can reconstruct it. Consider the key used to decrypt data as a secret. In a network of n nodes, consensus can be reached by issuing t shares to an eligible data recipient to allow the latter to decrypt the data. Any node cannot access the data independently, as it would need the help of other $t - 1$ nodes.

6.1.1.2 Proxy Re-Encryption

In a dynamic distributed communication between an arbitrary number of data holders and recipients, proxy re-encryption (PRE) represents a scalable cryptographic protocol that allows ciphering a datum without the need to know the recipient of that datum in advance. The general definition of PRE (Ateniese et al., 2006) consists of a public key encryption protocol that contains a re-encryption phase in which the plaintext is not revealed. Specifically, in this phase a sender (i.e., a data holder) with a key pair (pk_1, sk_1) generates a re-encryption key rk_{1-2} to be sent to a semi-trusted proxy server together with a ciphertext c_{pk_1} encrypted with the public key pk_1. Then the proxy server can run the proxy re-encryption algorithm to generate a new ciphertext c_{pk_2}, containing the plaintext, which is decryptable by a receiver (i.e., a data recipient) with the key pair (pk_2, sk_2). The proxy only uses rk_{1-2} and c_{pk_1}; therefore, it has no access to the plaintext. However, it must be semi-trusted because thanks to rk_{1-2}, it can re-encrypt any ciphertext with pk_1 in favor of the recipient. A specific instance of PRE is one-way proxy re-encryption, where the re-encryption function is one-way.

6.1.2 Related Work

The main intent of decentralized execution is to make the processes involved in managing access control fully auditable while at the same time ensuring strong cryptographic properties (Maesa et al., 2019). Systems based on DLTs and smart contracts can be leveraged in access control mechanisms to solve problems related to centralization and privacy leakage (Jemel and Serhrouchni, 2017; Rouhani and Deters, 2019) and to store, share and transmit data securely (Aiello et al., 2020; Hassanzadeh-Nazarabadi et al., 2021). In the following, we will describe works related to decentralized access control and some projects that involve using DLTs for GDPR-compliant data sharing. Finally, we will compare the works we cited in this Chapter and the Chapter related to the PDS (see Chapter 4) using Table 6.1.

6.1.2.1 DLT-based PIMS

Many researchers have attempted to envision data management systems where the user retains control over data, and in some cases, GDPR compliance is taken into account. (Hawig et al., 2019) propose an architecture based on distributed technologies to exchange health data. They aim to propose a system that provides immutability, interoperability, and GDPR compliance. DLTs have been involved in many cases of health information exchange, and there is a general trend in this field for integrating DLT-based systems in healthcare processes (Jiang et al., 2018). The work of (Koscina et al., 2019) enables healthcare data exchange through a distributed architecture, focusing on exploiting smart contracts for consent. Their specification is instantiated in the Horizon 2020 MyHealthMyData (MHMD) project, conceived to be the first open network of biomedical information to connect organizations and individuals. MHMD users keep a digital copy of their medical data in a personal data account that can be hosted on any cloud-based data management service. Users customize the dynamic consent preferences through smart contracts according to the type of data requested, by whom, and for what purpose. Data transactions are always traceable and compliant with GDPR. Another Horizon 2020 project related to DLT-based architectures for personal data is the DECODE project, which provides tools for individuals to take control of their data and share it. Thanks to these tools, it is possible to build a data-centric digital economy where data is generated and gathered by individuals and IoT devices, with appropriate privacy protections. Other authors that focus on personal data sharing are (Yan et al., 2017). They present a PDS that allows users to collect, store and give third parties fine-grained access to their data using a SS scheme. DeepLinQ (Chang et al., 2018) is a multi-blockchain architecture similar to our proposal. It aims to support privacy-preserving data sharing in the healthcare sector through granular access control and smart contracts.

Table 6.1: Summary of the features and comparison of the related works concerning our decentralized PIMS.

Work	Schemas	Off-chain	On-chain Anonymization	GDPR	Keys Distribution
(Shafagh et al., 2018)	Dual-Key Regression + ACL	FS	Yes, Stealth addresses	Not explicit, Possibly compliant	Efficient
(Jiang et al., 2019)	ECC	DFS	Yes, Stealth addresses	Not explicit, Possibly compliant	Data holder burden
(Ali et al., 2017)	ECDSA + Symmetric Encryption	DFS	No, Pseudonymous	Not compliant	Data holder burden
(Zhang et al., 2018)	ABE	FS	Yes, Attribute-Based Signature	Not explicit, Possibly compliant	Single point of failure
(Wang et al., 2018)	ABE	DFS	No, Pseudonymous	Not compliant	Data holder burden + Keys On-chain
(Xu et al., 2020)	ABE	DFS	Yes, Identity managed by central auth center	Not explicit, Possibly compliant	Single point of failure
(Chang et al., 2018)	RBAC	FS	Yes, Multi-DLT architecture	Not explicit, Possibly compliant	Efficient
(Zyskind et al., 2015)	ECDSA + Symmetric Encryption (+ Multi-party computation)	DFS	No, Pseudonymous	Not compliant	Data holder burden
(Yan et al., 2017)	Hierarchical SS	No	No, Pseudonymous	Not compliant	Efficient
(Truong et al., 2020)	ECDSA + ACL	FS	Yes, Private DLT	Compliant	Single point of failure
(Onik et al., 2019)	ECDSA + Symmetric Encryption	DFS	Yes, Private DLT	Compliant	Single point of failure
(Egorov et al., 2017)	KEM/DEM technique + TPRE	DFS	No, Pseudonymous	Not explicit, Possibly compliant	Data holder burden
Ours	KEM/DEM technique SS + TPRE + ACL	DFS	Yes, Multi-DLT architecture	Compliant	Efficient

ECC stands for Elliptic Curve Cryptography, ECDSA for Elliptic Curve Digital Signature Algorithm, ABE for Attribute-Based Encryption, RBAC for Role-Based Access Control, SS for Secret Sharing and TPRE for Threshold Proxy Re-Encryption. Possibly GDPR compliant means the provided architectures could be made compliant with low effort, e.g., appoint data controllers for permissioned DLTs nodes. The efficiency in the keys distribution operation is intended for the system user.

6.1.2.2 Decentralized and Attribute-Based Access Control

Existing literature provides many DLT-based access control system implementations based on attribute-based encryption (ABE) (Waters, 2011). This solution provides policy expressiveness without introducing many elements into the system infrastructure. ABE encrypts the data using a set of attributes that form a policy, and only those with a secret key that meets the policy can decrypt the data. This DLT-based access control through ABE is a specific case of general attribute-based access control (ABAC), where access is given after a policy evaluation based on subjects' attributes. An instance is using the eXtensible Access Control Markup Language (XACML) to enable access to a subject having attributes such as "email domain equals to abc.com" (Rouhani and Deters, 2019).

In (Zhang et al., 2018), the authors designed a system using ABE-based access control and smart contracts to grant data access, with a similar policies mechanism to our solution. In contrast, authors of (Wang et al., 2018) and (Xu et al., 2020) propose similar frameworks that combine DFS and blockchains to achieve fine-grained ABE-based access control. However, in all three works, the secret attribute keys are issued directly by the data holder in the DLT or central authority. This limits data sharing from the security (key immutably stored in DLT) and GDPR (right to data deletion) perspective.

Among these schemes, ABE presents issues such as privacy leakage from the private key generator (Hur and Noh, 2010) and a single point of failure (Jemel and Serhrouchni, 2017; Rouhani and Deters, 2019). Moreover, it also presents issues on feasible (in terms of efficiency and security) decentralized key generation and revocation (Meessen et al., 2018). In this Chapter, we focus on access key distribution rather than policy evaluation, e.g., ABAC. In fact, to reduce the complexity of the smart contract, we leverage an access control list (ACL) instead of attributes.

6.2 Personal Information Management System Architecture

In this Section, we provide an overview of the architecture of a multi-DLT PIMS based on the use of a smart contract for personal data access control. An audit DLT and an authorization DLT interact in a multi-DLT architecture to provide scalability and security to the whole PIMS. In the following, we will expand the architecture proposed up to this Chapter and then define a formal model for the cryptographic part of our proposal, which will not be rigorously proved in its entirety. We believe that a formal demonstration of it is outside the scope of our work and would take away space from another critical issue for the security of our system, namely the discussion of how we have implemented Privacy by Design in a decentralized context.

6.2.1 Actors and architectural components

Our first aim is to identify the actors involved in the architecture of the PIMS. Then we will give an overview of the architectural components.

6.2.1.1 Actors

We identify the following actors:

- ***Data subject*** (DS) - The same one as in Section 4.2 of Chapter 4. In the following, we will refer to this actor using the following notation DS.
- ***Data holder*** (DH) - The same one as in Section 4.2 of Chapter 4. In the following, we will refer to this actor using the following notation DH.
- ***Data intermediary*** (DI) - The same one as in Section 4.2 of Chapter 4. In this case, we have three specializations of data intermediary because the DLT providers are now of two kinds:
  - ***DFS provider*** (SP) - The same one as in Section 4.2 of Chapter 4. In the following, we will consider a SP_j being part of a set of DFS providers $SP = SP_1, ..., SP_m$, with $1 \leq j \leq m$.
  - ***DHT provider*** (TP) - The hypercube DHT (physical) node described in Chapter 5. In the following, we will consider a TP_c being part of a set of DHT providers $TP = TP_1, ..., TP_d$, with $1 \leq c \leq d$.
  - ***Authorization server*** (AS) - An actor within the (semi-)private permissioned authorization DLT that enacts the distributed authorization. In the following, we will consider an AS_i being part of a set of authorization servers $AS = AS_1, ..., AS_n$, with $1 \leq i \leq n$.

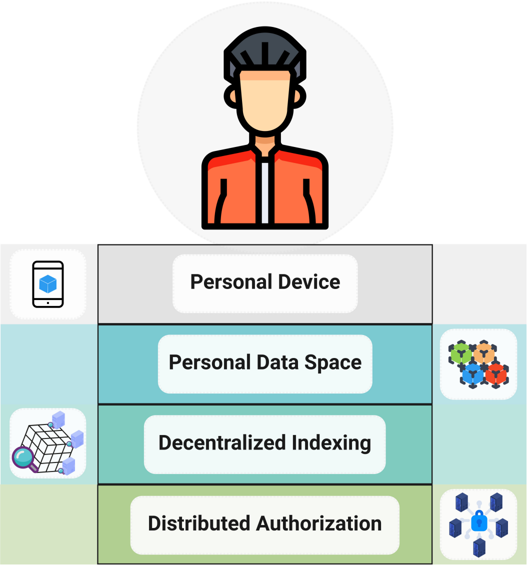


Figure 6.1: Layered view of the decentralized PIMS components.

   - *Audit DLT node* (AN) - The one that takes part in the audit DLT consensus mechanism, i.e., a mainchain full-node (Buterin et al., 2013; Nakamoto, 2009). This actor is in charge of registering the state updates of the authorization DLT into the audit DLT. In the following, we will consider an AN_f being part of a set of audit DLT nodes $AN = AN_1, ..., AN_g$, with $1 \leq f \leq g$.

- *Data recipient* (DR) - The same one as in Section 4.2 of Chapter 4. In the following, we will refer to this actor using the following notation DR.

6.2.1.2 Components

Regarding Figure 6.1, in the following, we describe the components of our architecture:

- **Personal device application** - Described in Chapter 4. It is a dedicated application that allows a data subject: (i) to decide how/where to store the data and to handle their encryption and secret keys generation when the subject is also a data holder; (ii) to manage the personal data access control thanks to the authorization DLT.

- **Personal Data Space** - Described in Chapter 4. It embodies the PIMS storage thanks to a DFS and contains the data that the holder encrypts or the recipient decrypts.

- **Decentralized Indexing** - Described in Chapter 5. A hypercube-structured Distributed Hash Table (DHT) is used to provide decentralized indexing for the search for data. This system is responsible for associating keywords to addresses or references stored in the DFS or the authorization DLT.
- **Distributed Authorization**
  - **Authorization DLT** - We leverage a (semi-)private permissioned ledger to orchestrate the access control mechanism and implement the on-chain hash pointers (see Chapter 4). This ledger is managed by a set of predetermined authorization servers appointed to check the access credentials and distribute the capsules that contain the secret keys used for the data encryption.
  - **Audit DLT** - This component is used to provide proof of a correct audit for the authorization DLT when one or more authorization servers (if not all) act maliciously. It consists mainly of a public permissionless ledger where the states of the authorization DLT are logged.

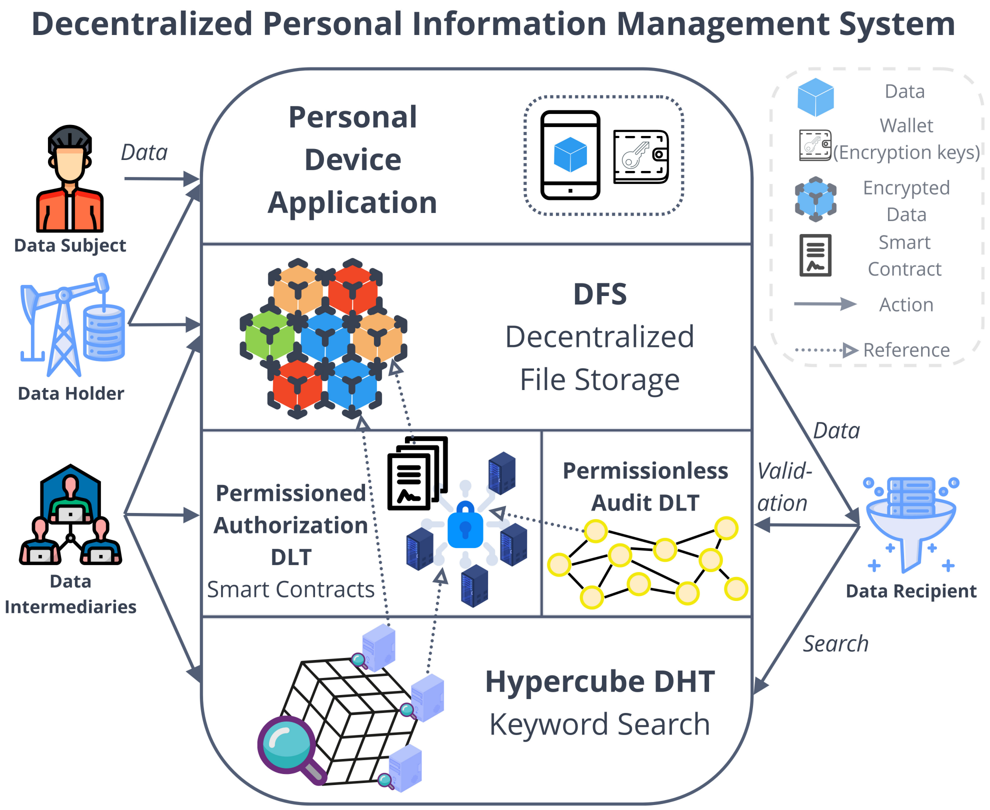
6.3 Components Design

This Section describes the components that form the decentralized PIMS architecture. These are graphically described in Figure 6.2, showing their relations and interaction with actors.

6.3.1 Personal data space and Decentralized Indexing

The PDS and decentralized indexing components used in the PIMS are described in Chapter 4 and Chapter 5. However, there are some differences with the previous descriptions of such components. In particular, the DLT component has been extended. It continues to be used to store personal data hash pointers, but it is now structured as a multi-DLT architecture and supports smart contracts (more on this in the following Sub-Sections). Moreover, the use of the personal device application and the distinction between data holder and subject can be further explained.

Figure 6.3 shows the interaction of data holders and subjects with the PIMS components. In particular, a data subject acting as a holder is represented on top. This



The top-right legend identifies the elements in the diagram. Solid arrows represent interactions between an actor or a set of actors and a system or network. Dashed arrows represent hash pointers to elements.

Figure 6.2: Diagram showing the architecture of the PIMS.

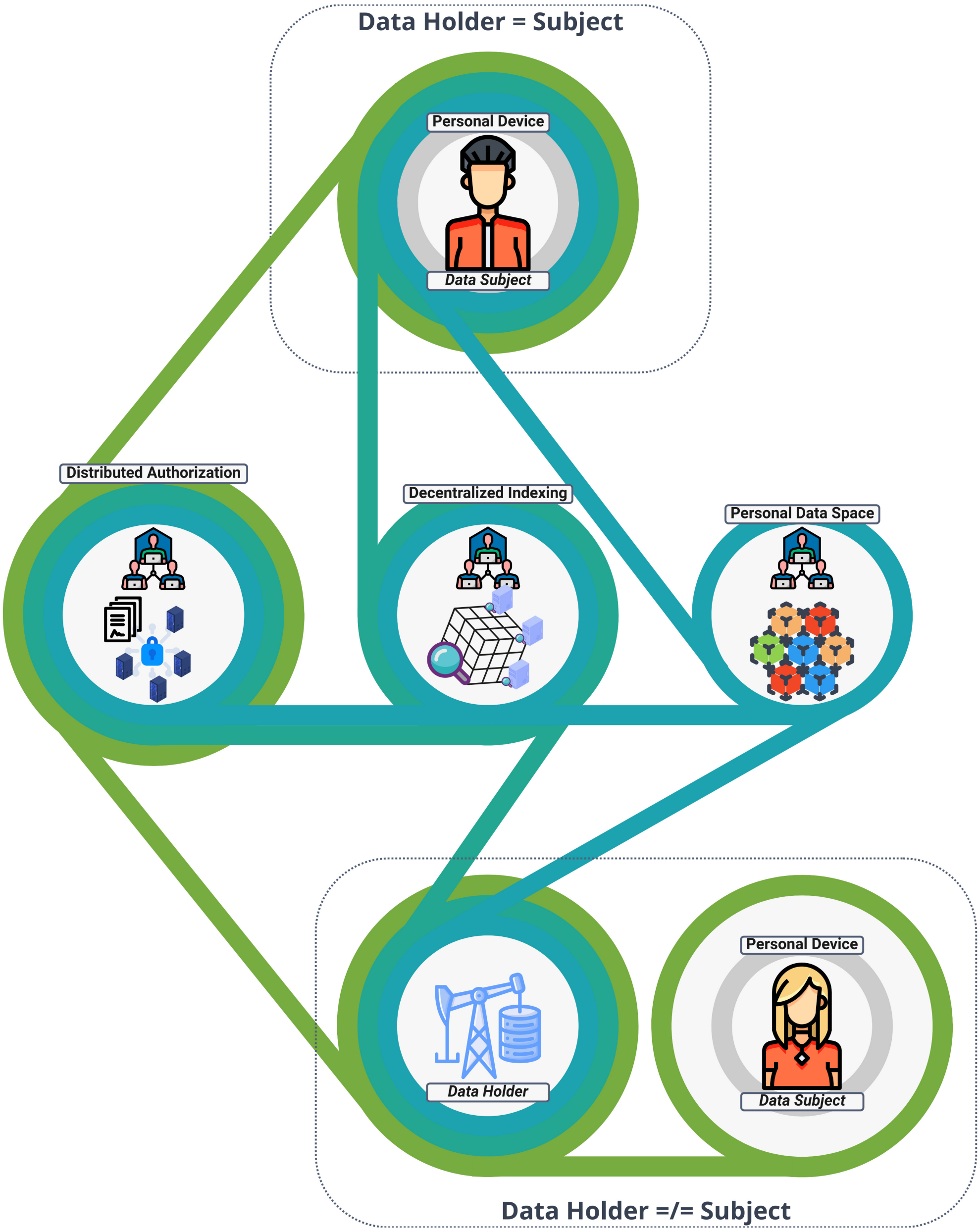


Figure 6.3: Diagram showing the relations between actors and the difference between data subject and holder.

case fully embraces the SSI paradigm, in which subjects are the only controllers of their data. In this case, the personal device application holds the wallet for managing encryption keys and communicates with the PDS, decentralized indexing, and distributed authorization components. At the bottom of Figure 6.3, a data subject interacts with a data holder to manage his/her personal data. In this case, the personal device application allows to use of the distributed authorization component and "indirectly" manage personal data sharing. In particular, the subject sets a set of policies (that will be enriched in Chapter 7) that the holder follows by system design through the authorization DLT. When referring to the data holder, we always consider an actor that covers both cases.

Model In the following, we use a model to refer to the elements managed in the system.

- The data holder actor controls a set of personal data that have not been encrypted, i.e., $PD = \{pd_l \mid 1 \leq l \leq o\}$. Notice that we generalized static and dynamic data in only one case to make the model simpler to be understood.
- Furthermore, $E = \{k_{pd_l} \mid Enc_{k_{pd_l}}(pd_l), 1 \leq l \leq o\}$ is the data holder's set of keys used to encrypt personal data and $EPD = \{epd_l \mid epd_l = Enc_{k_{pd_l}}(pd_l), 1 \leq l \leq o\}$ is the set of encrypted personal data.
- Data holders and authorization servers control a set of capsules $C = \{c_{k_{pd_l}} \mid c_{k_{pd_l}} = Enc_{pk_{DH}}(k_{pd_l}), 1 \leq l \leq o\}$ that contain a key used to encrypt a piece of personal data, as described in Section 4.3.1 in Chapter 4.
- We consider that all DFS providers SP store the data holders' set of encrypted personal data EPD and the associated set of decentralized identifiers used to $dixID$ to identify these. In this case the $dixID$ is equal to an hash pointer obtained by hashing the EPD, i.e., $HP = \{hp_{epd_l} \mid hp_{epd_l} = hash(epd_l), 1 \leq l \leq o\}$ (e.g., in the IPFS DFS these hash pointers are CIDs). Thus hp_{epd_x} is both the identifier of the epd_x datum in the DFS and an on-chain hash pointer, i.e., that will be stored in the authorization DLT.
- We also consider that all DHT providers TP store a subset of hash pointers that are indexed by keywords in the hypercube DHT, i.e., $HP^{DHT} \subseteq HP$.

6.3.2 Distributed Authorization

In our architecture, we leverage a network of authorization servers to provide access to eligible data recipients. The necessity of a distributed authorization system built on top of such a network has two reasons: i) to release the data holder from the burden of completely handling capsules distribution, which can be very expensive in terms of communication in case of fine-grained access; ii) to exploit smart contract distributed computation while complementing it with off-chain capsule distribution mechanisms since it is impossible to store secret keys or decrypt messages on-chain.

Authorization servers use a (semi-)private permissioned authorization DLT as a sidechain and the public permissionless audit DLT as the mainchain (Singh et al., 2020). These servers perform on-chain tasks such as smart contracts execution and off-chain tasks such as capsule distribution. In the following, the on-chain term will refer to the ledger of the sidechain (i.e., authorization DLT).

Authorization servers are the only ones in charge of enforcing the authorizations for data recipients made explicit in the access control smart contracts. We take advantage of the high degree of confidence that a DLT offers for the data written in the ledger. Therefore we concentrate on the trust given to the entities that have to read these data and follow the correct policy[4] Our proposal to decentralize this power in various authorization servers allows us to shift the trust from a centralized server to the protocol. Indeed, authorization servers may be considered semi-trusted or completely untrusted, but through the threshold mechanism presented in the following, the subject benefits from a cryptographic proof of security (Blakley, 1979; Egorov et al., 2017; ENISA, 2021; Shamir, 1979).

6.3.2.1 (Semi-)Private Authorization DLT

In the description of our sidechain, we reference the Ethereum protocol and its smart contracts (Buterin et al., 2013). However, multiple authorization DLTs can be implemented with different configurations or consensus algorithms, e.g., Hyperledger Fabric

[4] Consider the scenario where a single centralized server stores keys. Not only is this server susceptible to a single point of failure, but worse, it can act as an honest-but-curious provider. For instance, if curious, an online social network that correctly follows the protocol to share a user's geolocation data with the user's friends can also access this information.

(Androulaki et al., 2018; Bigini and Lattanzi, 2022). For instance, the consensus algorithm adopted by the network does not necessarily have to be the Proof-of-Work as in Bitcoin (Nakamoto, 2009) but can be chosen to provide a faster service. We refer to the Proof-of-Authority (PoA) consensus algorithm (Toyoda et al., 2020), which does not depend on solving a "cryptographic puzzle"; to issue a new block, this must be signed by the majority of the authorities, i.e., the nodes that are explicitly authorized to create new blocks and secure the blockchain.

In the authorization DLT, we have adopted extensive measures to minimize personal data transmission and/or storage through the ledger, following the principles of Privacy by Design. We define the authorization DLT as a "(semi-)private" one because, even if the DLT is a private permissioned one, there must be a way to access some parts of the ledger by users, for instance, the smart contract containing the validation data. Authorization servers, thus, maintain the full copy of the ledger, but they give read access rights in three cases:

- In the case of an audit, the whole ledger can be released to the entity allowed to perform it. It can be defined in the agreements between servers and holders/recipients;
- Data subjects and holders can access their ledger metadata (e.g., Ether balance (Buterin et al., 2013)) and the data related to their smart contracts in a complete mode, e.g., the hash pointers;
- Data recipients can access their ledger metadata and all the smart contracts in the DLT in a restricted mode; these are the on-chain hash pointers, the data schema, the holder's address, and the hash digest of the access control list only (see next Sub-Section for the detail of these elements).

6.3.2.2 Access Control Smart Contracts

The primary use of the authorization DLT is the execution of smart contracts implementing personal data access control. Access to the personal data stored in PDS can be allowed by the data holder through smart contracts. These access control smart contracts encode the eligibility of data access through a data structure, namely an access control list (ACL).

In practice, each piece of encrypted personal data $epd_l \in EPD$ is referenced in a specific smart contract through its on-chain hash pointer, i.e., hp_{epd_l}. Thus, the smart contract stores a subset of the HP set held in the DFS, i.e., $HP^{on\text{-}chain} \subseteq HP$. Figure 6.4 shows the data holder storing a new hp_{epd_l} in the access control smart contract.

In addition, concerning personal data, the contract also maintains a data schema. It is similar to the Sovrin scheme (Sovrin Foundation, 2020) and uPort claim spec (Lundkvist et al., 2017), i.e., a machine-readable format for specifying what to expect from the shared data. The recipient needs this in order to handle the data computation better, and it can be defined directly by the data holder or can be a specific standard of the authorization DLT.

The access control smart contract mainly consists of code to manage the ACL representing the access rights to a piece of encrypted personal data. The ACL contains a list of DLT addresses representing the data recipient. Addresses are associated with identities through the Intelligible Identity described in Chapter 8. At the same time, the data subject gives the policies for adding or removing addresses, e.g., by consent. The ACL can be (i) directly modified by the subject or (ii) indirectly modified by the data holder based on a policy set by the subject (in this Chapter, we will not focus with details on the distinction of their roles, see Chapter 7 for further details). Once a recipient is listed in the ACL, i.e., authorized to access some content, the authorization servers can verify this information through the ACL stored in the ledger. Eventually, when the recipient demonstrates to own the address listed in the ACL, servers release the $c_{k_{pd_l}}$ capsule that includes the k_{pd_l} content key needed to decrypt the encrypted data epd_l. See Algorithm 2 for the details of this operation and the next Sub-Section.

6.3.2.3 Capsule Distribution Mechanism

Consider the diagrams in Figure 6.4 and Figure 6.5. Previously, a data holder created a keypair (pk_{DH}, sk_{DH}) and a data recipient a keypair (pk_{DR},sk_{DR}). The first operation consists of storing the encrypted data epd_l in the PDS and obtaining the reference to the data, i.e., the hash pointer hp_{epd_l}. Then the access control smart contract is updated with the hash pointer hp_{epd_l}. While this can be considered a setup, the first real capsule distribution phase occurs when the holder shares the capsule $c_{k_{pd_l}}$ associated to epd_l with n authorization servers AS (function 3.n in Figure 6.4). The holder creates n fragments of the capsule such that $c_{k_{pd_l}} = \sum_i^n c^i_{k_{pd_l}}$ and each server receives its own

Algorithm 2: Authorization server providing access to data after a request

```
Global Data:
$i$ Server id
$C$ set of capsule stored by the server
$cha$ challenge message sent to recipient
$res$ response message for the challenge
Input:
$pk_{DR}$ Data recipient's public key
$contractAddr$ smart contract address
$hp_{epd_l}$ piece of data hash pointer
$sign$ signature of a challenge response
Result:
$c^i_{k_{pd_l}}$ capsule fragment for the $hp_{epd_l}$ piece of data

// validate signature to authenticate recipient
$obtainedPubKey \leftarrow$ verify($cha$, $res$, $sign$)
if $obtainedPubKey == pk_{DR}$ then
    // identity confirmed
    $acl \leftarrow$ getACLFromSmartContractWithAddr($contractAddr$)
    $eligiblesSet \leftarrow$ getValuesFromHashPointer($acl$,$hp_{epd_l}$)
    if $contains($eligiblesSet, $pk_{DR})$ then
        // eligible recipient
        $c^i_{k_{pd_l}} \leftarrow$ getCapsuleFromInternalStorage($C$,$hp_{epd_l}$)
        return $c^i_{k_{pd_l}}$
    end
end
return "Error: Not Authorized"
```

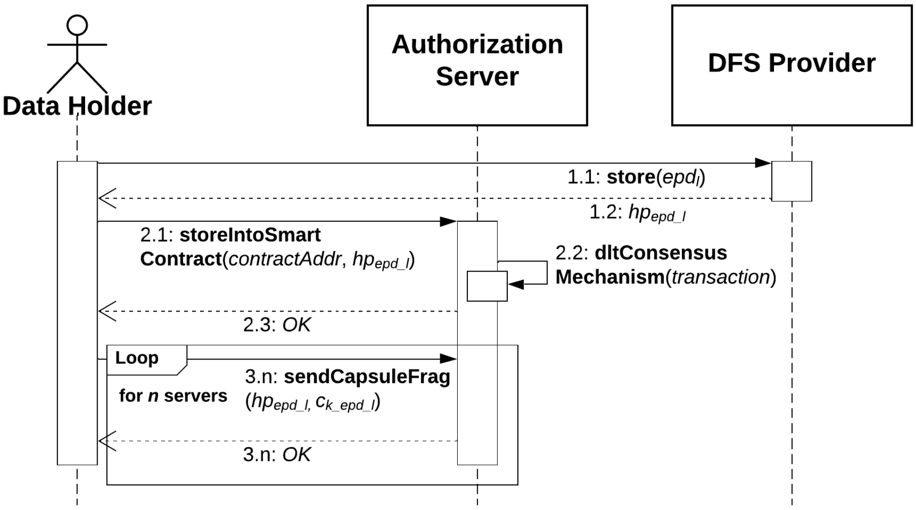

Data Holder
Authorization Server
DFS Provider
1.1: store(epd_l)
1.2: hp_epd_l
2.1: storeIntoSmart Contract(contractAddr, hp_epd_l)
2.2: dltConsensus Mechanism(transaction)
2.3: OK
Loop
for n servers
3.n: sendCapsuleFrag (hp_epd_l, c_k_epd_l)
3.n: OK

Figure 6.4: Sequence diagram describing the process of personal data storage by a holder.

capsule fragment. Here the sum operation represents the fragment aggregated function associated with the SS or TPRE methods, which is discussed in the next paragraph.

A data recipient may be allowed to access the stored data simply because the holder adds pk_{DR} to the ACL in the smart contract (function 1.1 in Figure 6.5). The second capsule distribution operation comes after the data access request made by the recipient to the authorization servers. The data access request is composed of these elements $\{pk_{DR}, contractAddr, hp_{epd_l}, sign\}$, where *contractAddr* is the address of the smart contract containing the ACL. The *sign* element is the signature of a challenge-response message to be signed with pk_{DR}, to allow each server to identify the data recipient, i.e., the recipient proves that owns the pk_{DR} public key. The function 2.2 in Figure 6.5 is the one each authorization server executes to provide access, i.e., release a capsule fragment eventually, and the pseudocode for this operation is shown in Algorithm 2. Upon receiving the data access request, t authorization servers (after having identified the recipient) check the ACL, found in the smart contract with address *contractAddr*, for the presence of pk_{DR}. Only if pk_{DR} is in the ACL, then each of the t servers releases its owned capsule fragment to the recipient (see return message 2.3 in Figure 6.5). Finally, the recipient can "open" the capsule and obtain the k_{pd_l} content key needed to decrypt the desired data after this was downloaded from the DFS.

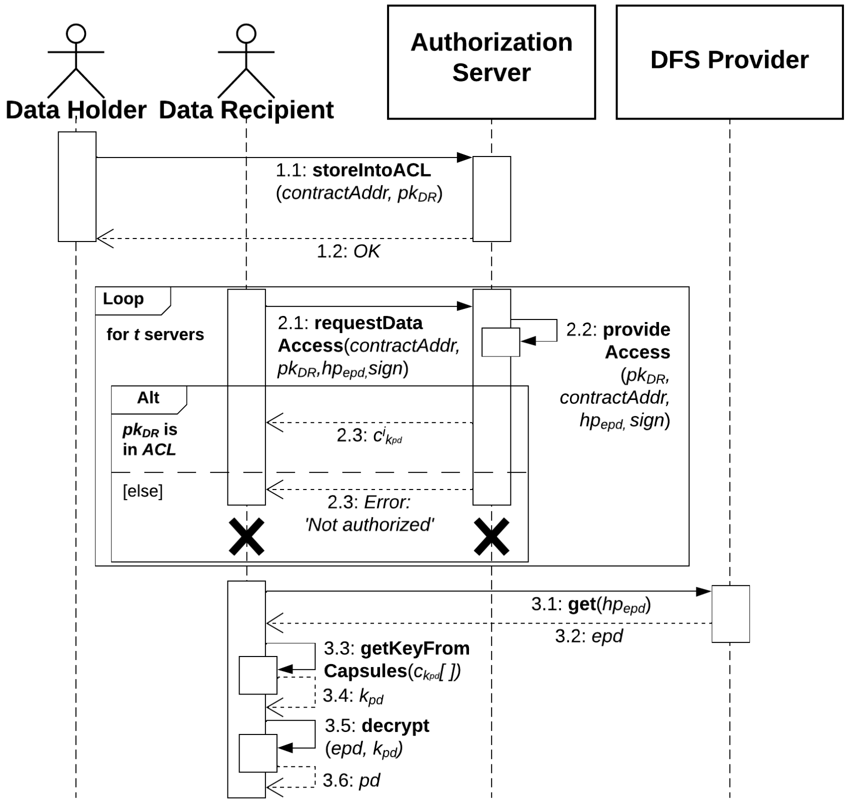

Data Holder
Data Recipient
Authorization Server
DFS Provider
1.1: **storeIntoACL** (*contractAddr*, pk_{DR})
1.2: *OK*
Loop
for *t* servers
2.1: **requestData Access**(*contractAddr*, pk_{DR},hp_{epd},*sign*)
2.2: **provide Access** (pk_{DR}, *contractAddr*, hp_{epd}, *sign*)
Alt
pk_{DR} is in *ACL*
2.3: $c^i_{k_{pd}}$
[else]
2.3: *Error: 'Not authorized'*
3.1: **get**(hp_{epd})
3.2: *epd*
3.3: **getKeyFrom Capsules**($c_{k_{pd}}$[ ])
3.4: k_{pd}
3.5: **decrypt** (*epd*, k_{pd})
3.6: *pd*

Figure 6.5: Sequence diagram describing the process of personal data access by a recipient.

Capsule Fragmentation In our architecture, two possible interchangeable mechanisms for capsule fragmentation can be employed. We recall what shown in Section 4.3.1 of Chapter 4 with regards to the generation of a capsule, i.e., in the specific case of a data holder holding a newly generated keypair (pk_{DH}, sk_{DH}), then a capsule containing the content key k_{pd_l} used to encrypt a piece of data pd_l is generated as $c_{k_{pd_l}} = Enc_{pk_{DH}}(k_{pd_l})$. In the following, we describe how these two cryptographic schemes work:

- **Secret sharing (SS)** By using this scheme, a new keypair (pk_{DH}, sk_{DH}) must be generated for each piece of data and then the sk_{DH} is "fragmented" into n fragments. For the data recipient, only $t < n$ fragments are sufficient to reconstruct sk_{DH}, decrypt the capsule and obtain k_{pd_l}.
  - A capsule fragment $c^i_{k_{pd_l}}$ consists of the original (encrypted) capsule $c_{k_{pd_l}}$ plus a fragment of sk_{DH}.

- The secret sk_{DH} can be represented as an element a_0 of a finite field, then $t-1$ elements are chosen randomly from this field, $a_1, ..., a_{t_1}$. Using these elements this polynomial curve can be constructed $f(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{t_1} x^{t_1}$. Every AS_i is given a point found in the curve $(x_i, f(x_i))$, with $1 \leq i \leq n$, called $cFrag_i$.
- Therefore, here, an untrusted authorization server alone cannot decrypt the capsule because it needs other $t-1$ *cFrag*s.
- Indeed, in order to obtain a_0, and thus sk_{DH}, a subset of cardinality t of the n points $(x_i, f(x_i))$ are needed to perform the following interpolation: $a_0 = f(0) = \sum_{j=0}^{t-1} f(x_j) \prod_{m=0, m \neq j}^{t-1} \frac{x_m}{x_m - x_j}$.

- **Threshold proxy re-encryption(TPRE)** By using this scheme, each authorization server acts as a proxy and re-encrypts the capsule fragment it receives through a re-encryption key $pk_{DH \rightarrow DR}$. The holder generates this key each time a recipient with public key pk_{DR} becomes eligible to access data.
  - Unlike the SS schema, the sk_{DH} is not the fragmented element that the server receives, but the re-encryption key is. Thus, a capsule fragment is subdivided into two kinds[5], i.e., (i) the one that the holder sends to each server that consists of the original encrypted capsule $c_{k_{pd_l}}$ plus a fragment of $pk_{DH \rightarrow DR}$, and (ii) the $c^i_{k_{pd_l}}$ obtained after a re-encapsulate process, that the server sends to eligible recipients.
  - The first kind of fragment is obtained through the same process shown for the SS, i.e., as a point $(x_i, f(x_i))$ of the constructed polynomial curve. We can refer to x_i as $kFrag_i$.
  - Then, $kFrag_i$ but it is used by the server to obtain $c^i_{k_{pd_l}}$, i.e., the $c^i_{k_{pd_l}} = cFrag_i = ReEncapsulate(c_{k_{pd_l}}, kFrag_i)$ (Nunez, 2018).
  - The recipient will receive $cFrag_i$, which can be used with other $t-1$ fragments to reconstruct a capsule encrypted with the recipient public key pk_{DR}. This because the original encrypted capsule $c_{k_{pd_l}}$ has been re-encrypted using the TPRE scheme.

[5] we are simplifying the description of the operations, but further details can be found in (Nunez, 2018)

- In this case, the process for decrypting the newly obtained capsule is similar to the interpolation shown previously for the secret sharing. The newly obtained capsule can be "opened", i.e., decrypted, using the recipient's private key sk_{DR}.
- TPRE solves the possible PRE collusion between proxy (i.e., server) and receiver (i.e., recipient) by assuming that at most $t-1$ proxies can be colluded and not follow the policies.

The system architecture we present supports both techniques, and each can be chosen based on different benefits and drawbacks. While SS relieves the holder of any interaction after the capsule has been fragmented the first time, there is still the possibility that t nodes are malicious. The holder cannot intervene to prevent keys from being disclosed. TPRE allows greater control over possible receivers; however, it has the disadvantage of requiring that a new re-encryption key $pk_{DH \rightarrow DR}$ is generated for each new recipient.

6.3.3 Audit DLT

As already mentioned, the above specification of architectural components is strongly influenced by two main issues. First, the tension between DLTs and the GDPR (Lyons et al., 2018). It led to using an authorization DLT with a set of designated authorization servers. Second, the scalability issues of current public permissionless DLTs (Bez et al., 2019) led to adopting a more scalable consensus mechanism in the authorization DLT. It means that using a permissioned DLT instead of a public permissionless one was due to legal and practical reasons. However, we opted for a multi-DLT architecture, including a public permissionless DLT, to reach strong confidence in data immutability. Indeed, permissioned DLTs allow a reasonable degree of confidence, but the ledger can be altered when most nodes are malicious (Singh et al., 2020).

A multi-DLT architecture achieves decentralization, high throughput, and low latency (Chang et al., 2018). The layer-one, i.e., the mainchain, consists of a public permissionless DLT used to perform batch transaction validation for layer-two, i.e., the sidechain, which is the (semi-)private permissioned authorization DLT. The basic idea is that the states of each authorization DLT, e.g., the ledger's transaction blocks in the case of a blockchain, are rendered immutable by registering them in a public permissionless DLT in the form of a hash digest. This is a common mechanism of sidechain technologies (Singh et al., 2020). The development of the audit DLT can be carried out with a smart-contract enabled blockchain, e.g., Ethereum, since it eases the information management, or also non-smart-contract enabled DLTs, e.g., IOTA DLT, for better scalability. Moreover, the audit DLT is also leveraged to store the information regarding the appointed authorization servers so that all of them can be identified and known by data holders and recipients, and optionally to let them stake an amount of cryptocurrency as an incentive for the correct behavior.

On this audit DLT, only non-personal data are stored, i.e.:

- a growing list of digests (i.e., roots of Merkle trees) that represent the advancement of the authorization DLT state, $\mathit{authStates} = \{\mathit{authState}_\tau \mid 0 \leq \tau \leq T\}$;
- a (modifiable) list of addresses representing the entities that provide the authorization service, i.e., authorization servers.

Authorization DLT consensus algorithm Several approaches might be implemented for updating the authorization DLT state in the audit DLT. In the following, we present one of them. According to our solution, to trace the evolution of the authorization DLT, we consider it as a blockchain that evolves its states in "rounds", i.e., the generation of r blocks is a round. At the end of each round, we store the last block's hash in the audit DLT. The pseudocode for this operation is shown in Algorithm 3. The r value must be chosen appropriately:

- it must be $r \neq 1$, to allow the blockchain to grow faster than the audit DLT, due to the different consensus algorithm (Bez et al., 2019); with $r = 1$, the blockchain would have to keep pace with the audit DLT;
- it must assume values $r \geq 2$, in order to ease the reaching of a secure level of data entropy to keep hashed data with a low risk of de-anonymization and thus to consider it non-personal (Agencia Española De Protección De Datos, 2019);
- it cannot assume a large value because the delay between one publication in the audit DLT and the next one may make it possible to alter the blockchain when the majority of servers are malicious.

Algorithm 3: Consensus algorithm for publishing the authorization DLT states to the audit DLT

```
Global Data: (constants)
$n$ Authorization servers number
$r$ round number
$id$ Server id
Global Data: (variables)
$step$ round step
$id_{latest}$ latest id of the round robin
$blocks$ list of blocks' hash digest
Input:
$block_{hash}$ hash digest of the latest block
Result:
$digest$ published in the audit DLT

$blocks \leftarrow \text{append}(blocks, block_{hash})$ // append latest block hash in the blocks hash list for this round
if step $== r$ then
    // round finishes
    if $id == id_{latest}$ then
        $digest \leftarrow \text{merkleTree}(blocks)$ // returns the root of a Merkle tree having blocks hash digests as leaves
        deployToAuditDLT($digest$)
    end
    $id_{latest} \leftarrow (id_{latest} + 1) \mod n$
    $step \leftarrow 1$
    $blocks \leftarrow [\,]$
end
$step \leftarrow step + 1$
```

Finally, the transaction to be issued in the audit DLT is signed with a multi-party signature of all authorization servers.

Table 6.2: PIMS architecture components description resumed.

Component	Description	Technologies To Use	Schemas	GDPR Compliance
(Personal) Device Application	Handles data subjects, holders and recipients keys and data	Data sensing, Cryptosystem and Wallet	KEM/DEM technique	SSI: holder = subject, or Data holder is controller, Data recipient is processor
Personal Data Space	Stores and provides encrypted personal data	DFS *e.g., IPFS*, FS *e.g.Dropbox*	On-chain hash pointers, Anonymous delegated deletion	DFS provider is controller/processor, Right to be forgotten
Authorization DLT	Validates data access requests and provides smart contracts for data access	(Semi-)private permissioned ledger	SS + TPRE + Smart Contract ACL	Authorization servers are joint controllers, Right to be forgotten, Privacy by Design
Audit DLT	Provides the proof of a correct audit for the authorization DLT	Public permissionless ledger	Multi-DLT sidechain protocol	Audit DLT node handles only non personal data
Decentralized Indexing	Provides keyword-based search for personal data	Hypercube DHT	Pin and Superset Search	DHT provider is controller/processor

6.4 GDPR compliance and analysis

In this Section, we discuss the GDPR compliance with regards to the PIMS architecture described, and then we provide a security and privacy analysis based on a Privacy Impact Assessment. Table 6.2 helps summarizing the description of the PIMS components.

6.4.1 Design for GDPR Compliance

- **Data intermediaries accountability** - The configurations available for implementing the PIMS architecture are many, e.g., having actors employing different roles such as authorization DLT and DFS provider together or using a Proof-of-Stake based authorization DLT. In any case, we consider data intermediaries, i.e., authorization servers, DFS providers, and DHT providers, as known entities that operate in a permissioned environment and have reached an agreement among themselves and with PIMS users. Most of the time, they act as data controllers and thus must be fully compliant with GDPR. The audit DLT node is possibly external to the PIMS if all the other actors behave correctly. Then, it controls only non-personal data (from now on, we assume this case and never the case that some actor stores personal data in the audit DLT). The data intermediaries agree on contractual terms that define the roles and duties and the privacy policy towards end users, i.e., data subjects, holders, and recipients. All the intermediaries within their systems (i.e., authorization DLT, FS, or DHT) act as joint controllers in a shared responsibility approach (CNIL, 2018a; Rieger et al., 2019b). The legal basis for personal data processing is guaranteed by mutual agreements. Moreover, from a data governance perspective, the whole set of intermediaries can be considered a data altruism organization operating on a non-profit basis (European Parliament, 2022).

- **Data intermediaries role** - From the GDPR point of view, the intermediaries might assume different roles depending on the operations performed on data. DFS providers handle only encrypted personal data that, with appropriate techniques, can be rendered meaningless without additional information. We attribute the role of the data controller to remain in the general case, but it could be the case that they have no obligations. The same goes for DHT providers that only manage hash pointers, i.e., non-personal data, but where the associated keywords might represent personal data. Concerning authorization servers, as stated earlier, they act as joint controllers for the transactional data that they verify, store, and put on/off-chain. However, authorization servers process a data holder request for distributing capsule fragments. Thus they are likely to be processors as they act on behalf of data holders. Servers can be controllers for some activities and processors for others (Lyons et al., 2018; Sovrin Foundation, 2020)
- **Authorization DLT personal data erasure** - When the operating environment is more easily controlled and regulated as in permissioned DLTs, it is more feasible to include the use of pruning (Finck, 2019; Politou et al., 2019). It consists of deleting old transactions and blocks (on demand or after a predefined period) and keeping the old block headers containing the hashed version of the removed

data to ensure the sidechain's security. However, this technique has been judged weakly applicable in permissionless DLTs (Palm, 2017). We apply this mechanism for the specific case where personal data erasure is needed in the authorization DLT.

- **Capsule distribution and data processing legal bases** - We assumed that the most used legal basis for providing access to personal data in the PIMS would be consent. For instance, consent conveyed in an electronic form could be conveyed together with a capsule fragment to each authorization server. It will be further described in Chapter 7. However, we are interested in concluding the discussion regarding using SS and TPRE schemes. Indeed, when using the TPRE scheme, some forms of the legal basis (Article 6(1)(a), GDPR) for rightfully obtaining a piece of personal data are hindered. If the data subject does not perform (or does not allow the holder to perform) the creation of a re-encryption key, in any event, the encrypted personal data may never be decrypted. In this case, a legal basis, such as the vital interest, could not be enacted to access data. To avoid such an issue, a re-encryption key should exist for each subject, enabling the re-encryption of a piece of data in favor of one or more authorization servers (the subject and/or holder can appoint that server).

6.4.2 Security and Privacy Analysis

In this Section, the security and privacy properties of the discussed model protocols are discussed but not proven formally. Different methods can be considered for such a task, such as game-based proof techniques (Unterweger et al., 2018), but we leave this task for future works. Here we analyze our model's security and privacy and conduct a risk analysis to assess the good practice to pursue GDPR compliance, such as in (Campanile et al., 2021).

In our analysis, we will also refer to the CNIL Privacy Impact Assessment methodology, which takes a qualitative approach (CNIL, 2018b). The assessment is carried out by estimating severity and likelihood using qualitative criteria. Severity represents the magnitude of risk and is primarily estimated in terms of the extent of potential impacts on data subjects. Likelihood represents the feasibility of a risk to occur, and it is primarily estimated in terms of the level of vulnerabilities of the supporting assets

Table 6.3: Summary of the PIMS model and related security and privacy threats.

Actor	Controlled/Processed Data	Possible Enacted Threats	S	L
Data holder DH	$PD = \{pd_l\}$, $E = \{k_{pd_l} \mid Enc_{k_{pd_l}}(pd_l)\}$, $C = \{c_{k_{pd_l}} \mid c_{k_{pd_l}} = Enc_{pk_{DH}}(k_{pd_l})\}$, all with $1 \leq l \leq o$.	(iv) Collusion with another actor	3*	1*
		(v) Tampering the ledger	3*	1*
		(vi) Repudiation	2*	1*
Data recipient DR	If authorized processes: the $c_{k_{pd_l}}$ and its related pd_l	(i) Illegitimate access to p.data	2	1
		(iv) Collusion with another actor	2	2
		(vi) Repudiation	1	2
		(vii) Denial of service	2	1
DFS providers $SP =$ $SP_1, ..., SP_m$	$EPD = \{epd_l \mid epd_l = Enc_{k_{pd_l}}(pd_l)\}$, $HP = \{hp_{epd_l} \mid hp_{epd_l} = hash(epd_l)\}$, all with $1 \leq l \leq o$.	(i) Illegitimate access to p.data	3	1
		(ii) Unwanted modific. of p.data	3	1
		(iii) Disappearance of p.data	3	1
		(iv) Collusion with another actor	3	2
		(v) Tampering the ledger	3	1
		(vii) Denial of service	2	2
DHT providers $TP =$ $TP_1, ..., TP_d$	HP^{DHT}, $K_{hp^{DHT}_{epd_l}}$, with $1 \leq l \leq o$, i.e., keywords associated to pointers	(ii) Unwanted modific. of p.data	1	2
		(iii) Disappearance of p.data	1	2
		(vii) Denial of service	1	1
Authorization servers $AS =$ $AS_1, ..., AS_n$	$HP^{on\text{-}chain}$, ACL, for each AS_i with $1 \leq i \leq n$: $C^i = \{c^i_{k_{pd_l}} \mid 1 \leq l \leq o\}$.	(i) Illegitimate access to p.data	3	1
		(ii) Unwanted modific. of p.data	2	1
		(iii) Disappearance of p.data	2	1
		(iv) Collusion with another actor	3	3
		(v) Tampering the ledger	3	1
		(vi) Repudiation	1	2
		(vii) Denial of service	2	1
		(viii) Lack of involvement in audit	2	3
Audit DLT nodes $AN =$ $AN_1, ..., AN_g$	$authStates$	(iv) Collusion with another actor	3	1
		(v) Tampering the ledger	3	1
		(vii) Denial of service	2	1
		(viii) Lack of involvement in audit	2	1

*The rightmost columns show some qualitative measurements in terms of severity (S) and likelihood (L) of the risk related to each threat, with scale: (1) negligible, (2) limited, (3) significant, (4) maximum (CNIL, 2018b). Values refer only to the risk impact for the data subject (values with the * also refer to the impact for other entities).*

concerned and the level of capabilities of the risk sources to exploit them (CNIL, 2018b). The scale used for severity and likelihood is (1) negligible, (2) limited, (3) significant, (4) maximum. While discussing the risks qualitatively, we will also go into the details of our model. Table 6.3 shows a summary of the system model and the severity and likelihood assessments related to possible threats. These threats are described in detail in the following:

(i) **Illegitimate access to personal data** - This threat is indicated as a feared event by CNIL (CNIL, 2018b) and indeed represents one of the most critical issues that our model tries to tackle.

(ii) **Unwanted modification of personal data** - Again, another CNIL's feared event can cause misuse, errors, and malfunctions, especially for the data subject.

(iii) **Disappearance of personal data** - The last CNIL identified a feared event that can result in similar results as an unwanted modification.

(iv) **Collusion with another actor** - It is a threat that deals mainly with the security of the whole model as any actor can collude with others trying to obtain a favorable result, e.g., managing to perform one of the previous three events.

(v) **Tampering the ledger** - This threat involves actors that might want to alter the traced information regarding the system processes, e.g., granting access to data.

(vi) **Repudiation** - This threat involves the fact that repudiation can be exercised by one of the actors in order to avoid accountability for its past actions.

(vii) **Denial of service** - Any actor providing a service can interrupt its service due to faulty or malicious behaviors. At the same time, any actor acting as a client can try to attack the service and interrupt its functioning.

(viii) **Lack of involvement in audit** - Finally, some actors might want to hamper the correct execution of an audit simply by not participating in it.

In the following, we will discuss the threats and risks analysis from the point of view of each model's actor.

Data holder Assuming that the implemented interface for the users is not faulty (and thus, it will not be considered in this analysis), data subjects acting as data holders might still provide *security* threats at the level of the processes of controlling and moving personal data. In this case, intentional and unintentional behaviors or implementation faults can (indirectly) provide harm to themselves as a data subject and to the actors providing their services lawfully.

(iv) The data holder can directly or indirectly (e.g., through a system fault) collude with the data recipient in order to render one or more authorization servers accountable for having provided illegitimate access to personal data to the recipient. The severity would be significant for the servers. In our model, authorization servers (i.e., data controllers) are protected from this threat because they maintain the ACL and its history thanks to the (semi-)private DLT and thus can demonstrate the data holder's malicious behavior.

(v) The same holds if the data holder modifies the smart contract data intending to tamper with the ledger, i.e., the history of the modification of the smart contract is kept by all the servers through the DLT.

(vi) Finally, the data holder can try to repudiate some of the actions performed; however, all the holder interactions with the systems involve the use of a public key or an address in the form of a digital signature, thus not being repudiated.

Data recipient

(i) The data recipient can attempt to illegitimately access personal data and thus threaten the subject's privacy. It seems difficult for this threat to materialize by exploiting the properties of the model. However, the severity would be limited by the few keys k_{pd_l} that the recipient can obtain from honest authorization servers.

(iv) However, collusion with another actor to access personal data, possibly through hacking or stolen credentials, is limitedly likely. Nevertheless, since each $pd_l \in PD$ is encrypted with a unique key, it would be difficult to access large quantities of data, thus limiting the severity of this threat.

(vi) The data recipient can repudiate the data access request made; however, this has almost no impact on the data subject (negligible severity).

(vii) Finally the recipient could try to limit the provision of service from the other actors in the system, i.e., $SP_1, ..., SP_m$, $AS_1, ..., AS_n$, $AN_1, ..., AN_g$. However, this would be very unlikely and with limited severity for large enough m, nand$g$, as the computation would be highly distributed and the data highly replicated.

DFS providers

(i) Each DFS provider SP_j has access to the whole set of encrypted personal data EPS and can try to decrypt it illegitimately. Even if the severity of this threat is significant, it is not very likely for a large enough t, where t is the threshold for the capsule fragment scheme because SP_j would need to collude with t malicious authorization servers.

(ii) Unwanted modification of data is very unlikely for a large enough m. Even in the case not decentralized, i.e., a single DFS provider, each modification to $epd_l^{'} \in EPD^{'}$, would show a discrepancy between the hash $hp^{'}_{epd_l^{'}} \in HP^{'}$ obtained from the new modified data and the hash $hp_{epd_l} \in HP^{on\text{-}chain}$ saved on-chain, e.g., in the case of a Cloud service could be a violation of the Service Level Agreement (D'Angelo et al., 2018).

(iii) Disappearance of personal data is likely to happen only in the case of a single DFS provider; however, it would be another violation of SLA. In the decentralized use case, high data replication would limit this threat.

(iv-v) As seen in the previous points, the DFS provider could collude with t malicious authorization servers to access data illegitimately, but this is very unlikely. An SP_j could also collude with some servers in order to modify an hash $hp^{'}_{epd_l} \in HP^{on\text{-}chain}$ saved on-chain. It is very unlikely because it would require breaking the consensus algorithm of the (semi-)private DLT.

(vii) A denial of service for data holders and recipients could be limitedly likely in the case of a single DFS provider and unlikely in the decentralized context of a DFS.

DHT providers

(ii) Unwanted modification of personal data handled by DHT providers is not unlikely; however, with very low severity in terms of privacy threats. The association between a keyword set $K_{hp^{DHT}_{epd_l}}$ and an hash pointer $hp^{DHT}_{epd_l}$ could represent personal data in terms of pseudonymous data when keywords are non-random and the $hp^{DHT}_{epd_l}$ could trace back to the data subject. For instance, a keyword set $K_{hp^{DHT}_{epd_l}} = \{$*"Photo", "Tour Eiffel", "23/10/2022"*$\}$ could be used to infer that $hp^{DHT}_{epd_l}$ points to a photo of the Tour Eiffel and that the data subject was in Paris the 23rd of October 2022. In this case, the unwanted modification of personal data, e.g., modifying the keyword to *"24/10/2022"*, would not be a severe threat, as keywords are only used to operate and not as an authoritative source.

(iii) Disappearance of personal data is likely to occur if a DHT provider responsible for some subject data ceases to operate without replicating the data in a remaining part of the DHT network. Also, in this case, the severity would not be high, as keywords could be re-inserted into the system.

(vii) A denial of service for data holders and recipients in terms of searches in the hypercube DHT is not very likely due to the P2P properties of the system, e.g., replication.

Authorization servers

(i) Considering the whole set of encrypted personal data EPD, a single authorization server AS_i could try to decrypt this set to access personal data illegitimately. Also, in this case, the severity of this threat is significant, but the event is unlikely because other $t-1$ malicious servers are needed.

(ii-iii) Considering the set of capsule fragments C^i and the ACL as personal data, a single AS_i could modify or delete them without the data subject's permission. In this case, the severity is limited because AS_i would need $n-t$ other malicious servers to modify/delete capsule fragments so that they can be made useless, $t-1$ malicious servers to modify the original capsule, and enough servers to disrupt the consensus mechanism to modify/delete the ACL.

(iv) The collusion of AS_i with other servers is significantly likely to happen if all these do not have the right incentive to deviate from this behavior. This event would be of significant severity because it would allow the colluded servers to illegitimately access personal data and let other recipients do it too. This event depends as well on the t value. Thus a large enough t and choosing the right incentives are the key factors that enable this threat to be avoided.

(v) Tampering the ledger could lead to the modification of the ACL and thus allow illegitimate access to personal data. This threat is unlikely as an AS_i would need enough servers to disrupt the consensus mechanism.

(vi) A AS_i can repudiate the action of sending out capsule fragments on request, but capsules are digitally signed. However, this has almost no impact on the data subject (negligible severity).

(vii) A denial of service for data holders and recipients is unlikely in the decentralized context of the (semi-)private DLT.

(vii) Lack of involvement in an audit is significantly likely to happen for an authorization server, as it would mean simply not to show the ledger of the (semi-)private DLT to the auditors. This could be the only option to hamper the audit process, as an alteration of the ledger would be detected thanks to the audit DLT *authState*. In this case, the server, being a controller, would be liable for not cooperating, on request, with the supervisory authority (GDPR, Article 31).

Audit DLT nodes

(iv-v) Very unlikely that the majority of audit DLT nodes $AN_1, ..., AN_g$ could collude and modify the *authStates* by themselves or on demand by colluding with an authorization server. It is very unlikely for large enough g, as in Bitcoin and Ethereum public blockchains.

(vii) An audit DLT node could deny the service of storing an *authState* to a server, but again with a large enough g, it is very unlikely for the server not to find other nodes providing the service.

(viii) Lack of involvement in an audit is very unlikely to happen as the audit DLT nodes might not even know about the actual meaning of the *authStates* information.

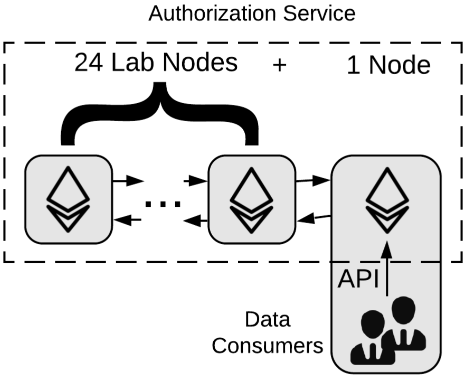


Figure 6.6: Setup of the authorization DLT network for testing SS vs. TPRE.

Being nodes of a public permissionless DLT, they would simply share the ledger as it is public to anyone.

6.5 Implementation and Evaluation

This Section describes the implementation and tests to evaluate the proposed system. Here also, as in Section 4.4 of Chapter 4, we are interested in the main critical points related to scalability, responsiveness, and reliability of the state of the art technologies we employed. In particular, we implemented the multi-DLT architecture based on the Ethereum protocol, considered at the center of the Web 3 "world" for developing smart contracts.

6.5.1 Implementation

The implementation of the components of our proposed PIMS consists of the following:

- The **personal device application** has been developed as a module that was used for simulating several instances of devices that interact with the PIMS. Our primary focus of such an implementation was to build a wallet and a cryptosystem compatible with the SS and TPRE schemes.
- The **PDS component** implementation is the same shown in Chapter 4.
- The **authorization DLT** implementation consists of a set of Solidity smart contracts and three different DLT setups (Zichichi, 2022c). Two setups have been

developed on a private network of Ethereum nodes running the OpenEthereum software in order to test the SS scheme against the TPRE one: i) for the SS scheme, we used the native Secret Store (OpenEthereum, 2020) provided by the OpenEthereum application, while ii) for the TPRE scheme we developed a server application running the Umbral python library in NuCypher (Egorov et al., 2017). The third setup is the fully implemented authorization service, running on a private network of Ethereum nodes using the Consensys GoQuorum software (Mazzoni et al., 2022). ConsenSys Quorum is an open-source protocol layer aiming to build Ethereum-compatible environments for enterprises. The rationale behind this choice is to be able to implement private smart contracts and transactions for protecting personal data stored on-chain by the data holders, a feature that GoQuorum supports. We have also rewritten the source code of the TPRE Umbral protocol in Rust and made it openly available (Zichichi, 2021e). The new Rust implementation is integrated with the GoQuorum software and run by every authorization server.

- The **audit DLT** implementation consists of the use of IOTA for storing messages, such as in Chapter 4.

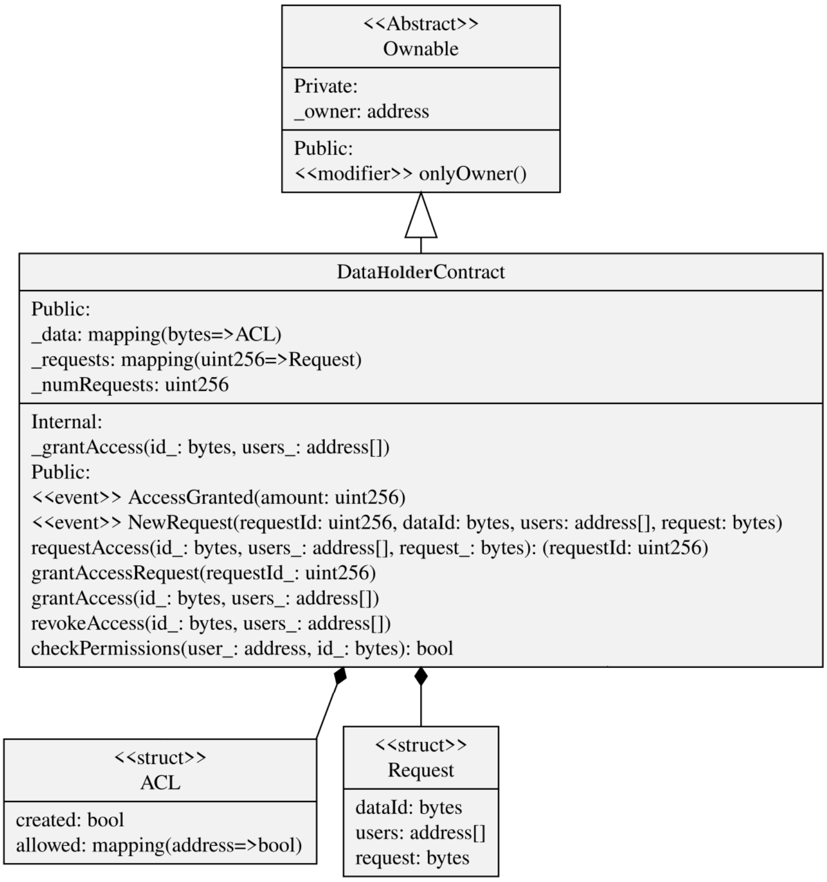
- The **decentralized indexing component** implementation is the same shown in Chapter 5.

6.5.1.1 Smart Contracts implementing the data access control

As seen in Sub-Section 6.3.2, the interesting aspect of smart contracts is that an algorithm executed in a decentralized manner enables two parties, i.e., data holder and recipient, to reach an agreement on the sharing of the data. It increases the disintermediation in such a process, leaves traces to be later audited, and provides incentives to all the actors to behave correctly. Figure 6.7 shows the UML Class Diagram of the smart contract implementations we will discuss in the following.

- Each data holder has previously deployed a *DataHolderContract* in the authorization blockchain.

- In the "step zero" the recipient has obtained a list *DataHolderContract* addresses that point to $epd_l \in EPD$, with $1 < l < o$, through as hash pointers $hp_{epd_l} \in$


<<Abstract>>
Ownable
Private:
_owner: address
Public:
<<modifier>> onlyOwner()
DataHolderContract
Public:
_data: mapping(bytes=>ACL)
_requests: mapping(uint256=>Request)
_numRequests: uint256
Internal:
grantAccess(id: bytes, users_: address[])
Public:
<<event>> AccessGranted(amount: uint256)
<<event>> NewRequest(requestId: uint256, dataId: bytes, users: address[], request: bytes)
requestAccess(id_: bytes, users_: address[], request_: bytes): (requestId: uint256)
grantAccessRequest(requestId_: uint256)
grantAccess(id_: bytes, users_: address[])
revokeAccess(id_: bytes, users_: address[])
checkPermissions(user_: address, id_: bytes): bool
<<struct>>
ACL
created: bool
allowed: mapping(address=>bool)
<<struct>>
Request
dataId: bytes
users: address[]
request: bytes

Some classes, attributes, and methods have been removed to make the diagram clearer.

Figure 6.7: UML Class Diagram of *DataHolderContract*.

Table 6.4: DataHolderContract smart contract methods gas usage.

Smart Contract	Method	Gas usage
DataHolderContract	grantAccess()	96 436
	requestAccess()	142 648
	grantAccessRequest()	77 706
	revokeAccess()	30 126

$HP^{\text{on-chain}}$. Then the recipient produces a data access consent request in a string form (more on this in Chapter 7).

- The recipient gives as input the requested hp_{epd_l}, request string and an array of addresses to the method *requestAccess()* of the *DataHolderContract*. Figure 6.7 shows *id_* as a parameter representing the hash pointer and an array of addresses *users* for representing the Ethereum accounts that will be granted access.
- A *NewRequest* event will reach each data holder. This one decides to provide consent to the data processing based on the data access consent request received through the event. If so, the data holder invokes the *grantAccessRequest()* method in the *DataHolderContract*.
- The recipient uses the *checkPermissions()* method to check if the data holder granted access to the requested data.
- The recipient can now access all content keys for the decryption of all the data holder's data through the authorization servers.

Smart Contracts Gas Usage In Ethereum, the *gas* is a unit that measures the amount of computational effort needed to execute operations. Thus, the higher the gas usage for a method, the more intense the computation of a blockchain node to execute the method's instructions. In Table 6.4, we provide the execution cost for the main methods of the *DataHolderContract* smart contract.

These smart contract methods can be considered relatively cheap compared to state of the art. This result is needed because these methods are executed many times. The method *requestAccess()* is the one with the highest gas usage because it inputs several parameters, i.e., hash pointers, a list of Ethereum accounts, and a string for the request.

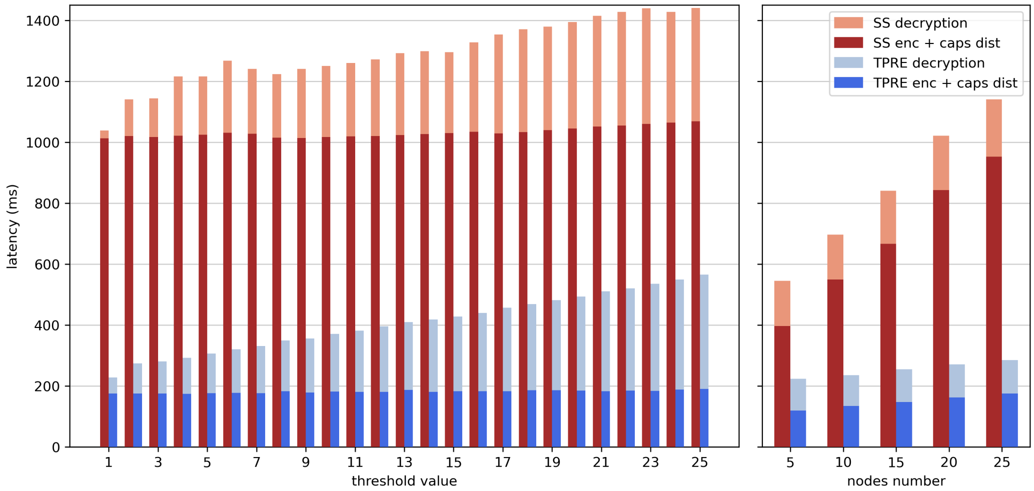


Figure 6.8: Encryption and decryption latencies for SS and TPRE.

Generally speaking, the methods executed the most do not appear to concern their execution in a private permissioned blockchain environment. However, this is one of the main aims of our performance evaluation.

6.5.2 Evaluation

In this performance evaluation, we are mainly interested in the user experience provided by a PIMS, such as that we implemented. The source code for the testing can be found in (Zichichi, 2022c; Zichichi and Sparber, 2021). Thus, the main critical points are related to the authorization service's scalability, responsiveness, and reliability. In particular, we focus on two kinds of tests: i) comparison between the two SS and TPRE cryptographic schemes and ii) the time required for the access control operations.

6.5.2.1 Cryptographic schemes performances

The first experimental test was run using our first two prototype setups, based on the private network of OpenEthereum nodes. We tested how the network responds to the simulated data recipients' requests based on different system configurations. We have measured the latency needed to perform each operation from the point of view of the data holder and data recipient device application. In this test, we configured a network of 25 nodes to form the authorization DLT. 24 of these nodes are hosted in the same laboratory of the University of Bologna (each one is equipped with an Intel Core i5-2400

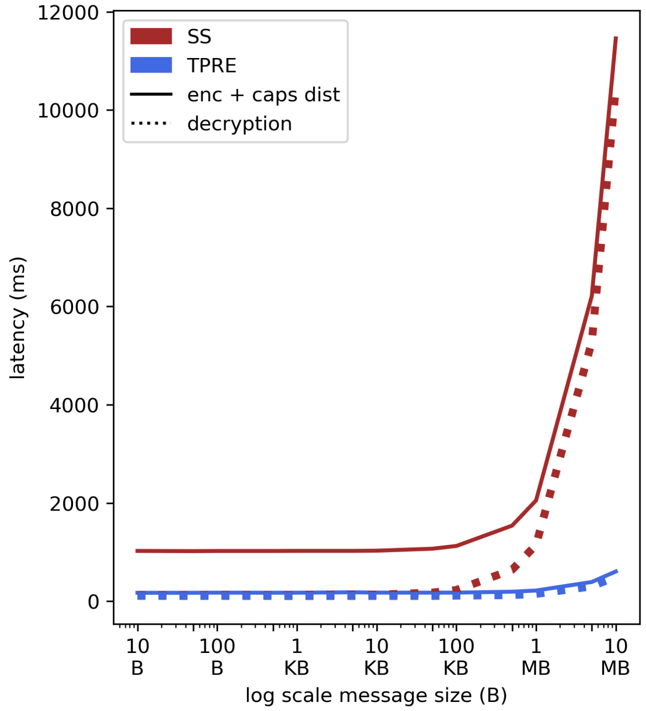


Figure 6.9: Latencies when encrypting and decrypting messages varying message sizes.

CPU and 8 GB RAM). The simulation device is external to the laboratory and acts as the 25th node and a data recipient's device application. A graphical representation of such a network is shown in Figure 6.6. Each of the 25 nodes provides the capsule distribution service by running (not simultaneously) the two SS and TPRE schemes implementations.

The following tests were performed to evaluate the authorization DLT. Here we refer to n for the total number of network nodes and t for the threshold (i.e., expressed as the number of nodes) needed to reconstruct a capsule completely. We tested:

- **Threshold Variation**: this test case involves the variation of t, after having fixed $n = 25$ as the number of nodes and 30 B as the message size. We intend to show the latencies measured when the threshold value increases.

- **Nodes Number Variation**: we set up this test case to assess the different configurations of the authorization DLT by varying the number of authorization server nodes, i.e., n. The threshold value was set to $t = 2$, while 30 KB was the size of a message.

Table 6.5: Threshold latencies (mean) when encrypting (+ distributing capsules) and decrypting messages with SS and TPRE schemes.

Thresh	SS Encryption	TPRE Encryption	SS Decryption	TPRE Decryption
5	1024 ms	176 ms	192 ms	130 ms
10	1017 ms	182 ms	233 ms	189 ms
15	1030 ms	183 ms	265 ms	245 ms
20	1045 ms	185 ms	349 ms	309 ms
25	1069 ms	190 ms	371 ms	376 ms

Table 6.6: Nodes number latencies (mean) when encrypting (+ distributing capsules) and decrypting messages with SS and TPRE schemes.

# nodes	SS Encryption	TPRE Encryption	SS Decryption	TPRE Decryption
5	397 ms	120 ms	148 ms	104 ms
10	549 ms	135 ms	148 ms	101 ms
15	666 ms	147 ms	175 ms	108 ms
20	843 ms	163 ms	178 ms	108 ms
25	952 ms	175 ms	188 ms	110 ms

- **Message size variation**: tests were carried out to assess the encryption and decryption performance with different types of messages, ranging from small text data (10 B) to larger data (10 MB).

For all tests, every single data point was obtained by repeating the operation 10 times (with a time interval of 300ms between each request) and then averaging the resulting latency values.

Results

- **Threshold Variation** - As described before, the threshold is a key parameter for both TPRE and SS. In essence, it defines how many nodes must agree on the status of the smart contract. We fixed the number of nodes on the network to 25 and then tested values of t from 1 to 25. As the left-most bar chart in Figure 6.8 shows, the encryption (+ capsule fragments distribution) time remains mostly constant ($\sim$183 ms for TPRE and $\sim$1045 ms for SS). On the other hand, the decryption time increases linearly with t, and both TPRE and SS present a very similar trend. This is because, with the increase of t, more nodes are involved in the decryption. While SS and TPRE schemes take approximately the same

Table 6.7: Message size latencies (mean) when encrypting (+ distributing capsules) and decrypting messages with SS and TPRE schemes.

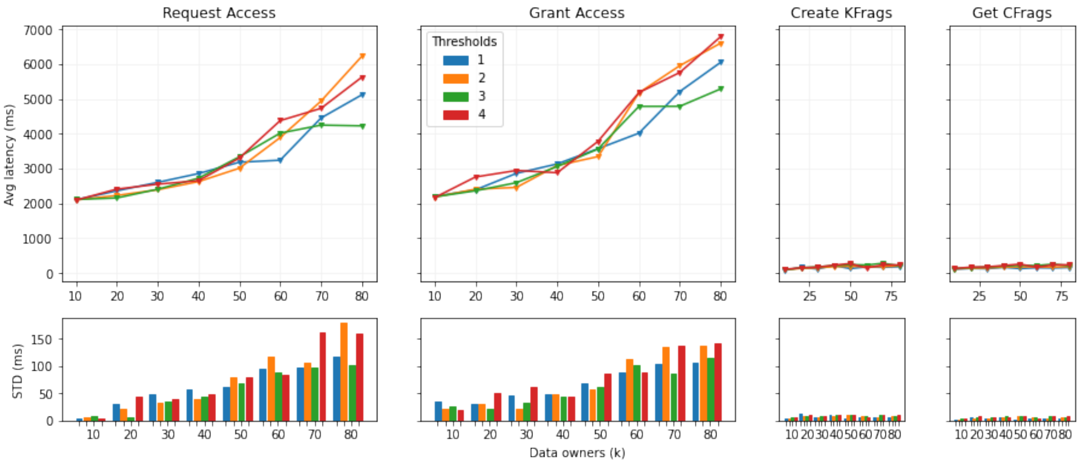
Size	SS Encryption	TPRE Encryption	SS Decryption	TPRE Decryption
10B	1026 ms	174 ms	126 ms	111 ms
50B	1022 ms	174 ms	126 ms	109 ms
100B	1025 ms	177 ms	125 ms	110 ms
500B	1025 ms	175 ms	125 ms	109 ms
1KB	1027 ms	176 ms	126 ms	108 ms
5KB	1027 ms	185 ms	129 ms	109 ms
10KB	1031 ms	178 ms	135 ms	109 ms
50KB	1071 ms	177 ms	178 ms	110 ms
100KB	1127 ms	178 ms	231 ms	116 ms
500KB	1541 ms	196 ms	642 ms	127 ms
1MB	2054 ms	220 ms	1150 ms	151 ms
5MB	6214 ms	394 ms	5278 ms	305 ms
10MB	11456 ms	608 ms	10452 ms	502 ms

time to perform the decryption operation, the same cannot be said for the whole encryption operation. The largest difference in latency can be observed in the generation and distribution of capsule fragments embedded in the encryption operation. SS performs a distributed capsule generation scheme based on Elliptic Curve Discrete Logarithm (Tang, 2005) to set up the keys needed in KEM and DEM. In contrast, TPRE capsule and fragment generation are performed locally and distributed to nodes. This difference is heavily reflected in encryption results, e.g., in Table 6.5 one can see ∼879 ms time difference for $t = 25$.

- **Nodes Number Variation** - Network scalability is crucial in distributed systems. While the data holder can set the threshold value, choosing between higher security or faster operations, the number of nodes and the resources allocated are usually fixed. In the center chart in Figure 6.8, it is possible to see that, as expected, the time costs of operations increase, in general, with the number of nodes n. However, we must note that the values for the SS results grow faster than those for TPRE results. It makes the TPRE method more scalable. Concerning the previous results (threshold value), Table 6.6 confirm that the encryption values here are higher for SS and increase linearly with n in SS and TPRE. The decryption times remain almost constant because t is never altered. Even in these tests, we can see the effect of SS capsule generation and distribution since a larger number of nodes means an increase in the time needed to reach all of them.

- **Message Size Variation** - Figure 6.9 and Table 6.7 show the latency averages with respect to the message sizes for the process of encryption/decryption. The encryption also includes the capsule fragments distribution. The plot suggests that the TPRE scheme implementation can handle the whole process better. From 10 B to 1 MB, TPRE has a constant response latency, $\sim$ 175ms for encryption, while SS results curve exhibits a noticeable inflection point as the message size reaches 500 KB. Then it skyrockets from 1 MB onward. In this case, there is a latency increase of 864%, from 1 MB to 10 MB, for the encryption of the message (without considering capsule generation) and of 809% for the decryption. This requires two explanations. First, it is inevitable to increase after 1 MB for both schemes because, before this threshold, the message size has almost no impact on the operations, i.e., latencies are stable. Second, we have a clear difference between SS and TPRE scheme after the 1 MB, probably due to the SS implementation that during all the tests has presented an overhead concerning the TPRE implementation; moreover, capsules fragments distribution might also have a significant impact at the expense of the SS schema here, since this is common also in the other tests.

Discussion Our performance evaluation shows that, with respect to SS, TPRE is: (i) faster when increasing the size of the messages; (ii) more scalable, as it better manages the increase in the number of nodes executing the protocol; (iii) more efficient when increasing the threshold value due to its shares generation method. On the other hand, TPRE has the drawback of requiring the data holder to generate a re-encryption key for each new data recipient. Results clearly show how the TPRE scheme, implemented as a system based on NuCypher, performs better than the SS scheme, implemented as OpenEthereum Secret Store. An essential comment about a significant difference between the considered TPRE and SS schemes is needed. We have performed the performance evaluation assuming that the data holder device that releases the re-encrypted capsules is always operational and online. However, in a real-world scenario, this device might introduce some delays when using the TPRE scheme and not when using SS since, in the SS scheme, the holder device is not required to complete the



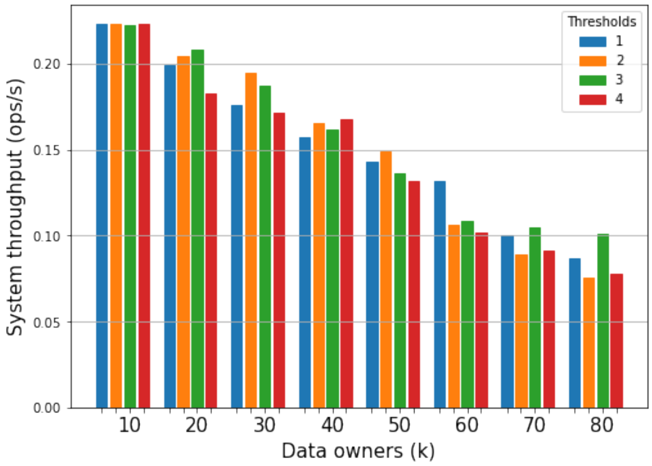
Average response latency and standard deviation for each operation in a round, varying the threshold from 1 *to* 4 *and data holders from* 10 *to* 80.

Figure 6.10: Plots showing the latency when operating with the authorization DLT.

operations. This aspect can influence the choice between the two schemes both from an operational point of view and from a performance perspective.

6.5.2.2 Authorization DLT based on GoQuorum and Umbral-Rust

Our final implementation of the authorization DLT is based on a network of nodes running the GoQuorum software, integrated with the Umbral TPRE libraries in Rust. During the test of the DLT, we used the Istanbul Byzantine Fault-Tolerant (IBFT) consensus mechanism: each block requires multiple rounds of voting by the set of validators ($> 66\%$), recorded as a collection of signatures on the block (Mazzoni et al., 2022). Four validator nodes were deployed during the tests to create the base blockchain network. Each validator node executes the consensus mechanism with parameter values set up following the recommendations in (Mazzoni et al., 2022), e.g., minimum inter-block validation time is set to 1 second. Moreover, these nodes also execute the TPRE service. One non-validator node is used to expose the APIs for external clients to interact with the blockchain. Several client nodes are created to interact with these APIs, which disseminate transactions within the network (Serena et al., 2021). The network was run on a server with a 10 cores Intel Xeon CPU and 8 GB of DDR4 RAM.



System throughput considering a round as a single operation, i.e., aggregating the results for every single operation, while varying t and k.

Figure 6.11: Histogram showing the system throughput during tests.

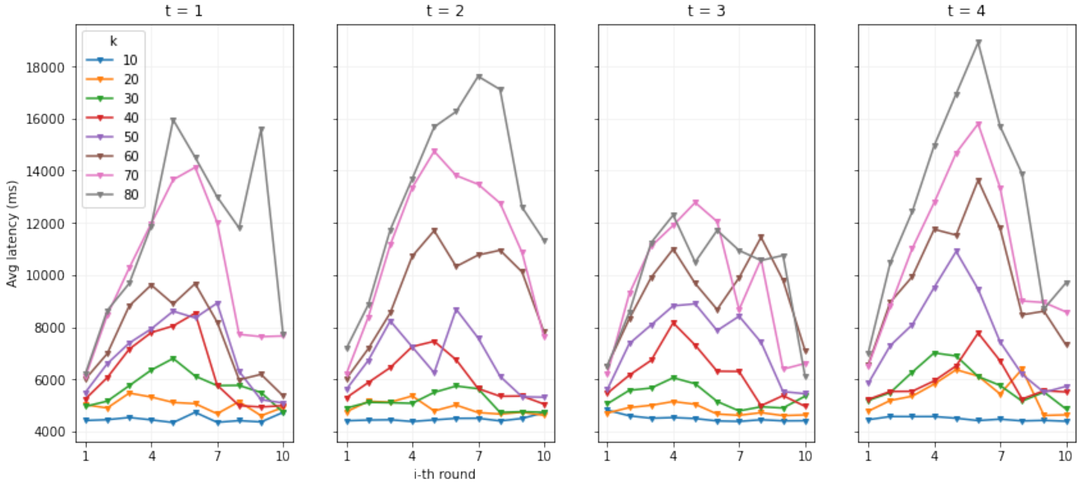
In the following, we evaluate this set of operations;

1. **Request Access** - this operation is executed by the data recipient and consists of only one method invocation of a new dedicated request smart contract, i.e., the *requestAccessToData()* method that requests access to data for many *DataHolderContract*s given as input.

2. **Grant Access** - this operation is executed by each data holder by invoking the *grantAccessRequest()* from their own *DataHolderContract*; this will store the recipient public key pk_{DR} in the smart contract ACL.

3. **Create KFrags** - this operation includes three subsequent steps; firstly, the holder generates a new set of n kfrags using the data recipient's pk_{DR}; then the holder sends a kfrag each to the n authorization servers; finally, the holder requests to the n nodes the creation of a cfrag using the kfrag just got (the encrypted capsule for the piece of data interested was sent in a pre-processing step, not accounted for the measuring).

4. **Get CFrags** - the data recipient executes the last operation to get access to the content key; the recipient firstly signs a challenge-response message using the secret key sk_{DR} associated with the pk_{DR}; then the recipient sends a get cfrag request



Average response latency when increasing the threshold t value and the number of holders k for each i-th round.

Figure 6.12: Plots showing the average latency during tests.

to k authorization servers using the signed message; each node validates the signature and checks if pk_{DR} is in the associated ACL in the *DataRecipientContract*; if so, each node returns a cfrag to the data recipient.

We recall n is the number of validator/authorization blockchain nodes and was set to 4. We consider a round of operations the successful execution of the above-described operations in order. The independent variables tested were the *threshold* t, from 1 to 4, and the *number of data holders* k, from 10 to 80 with an increase of 10 each time. We tested all the combinations of independent variables three times, then averaged the results. In each test, we initiated the round of operations 10 times for each data holder, with an interval of 3000 ms on average (value given by a Poisson Process with a mean of $3000ms$). It implies that if, overall, the set of operations lasted more than 3000 ms to be executed, probably another one was launched in parallel. This is for each data holder. The dependent metrics we measured with the tests are the *latency*, for a response to an operation, and the system *throughput*, i.e., the number of rounds of operations per second.

Results

- **Round of operations** - Figure 6.10 shows the average response latency and

Table 6.8: Average response latency and confidence interval for Create KFrags and Get CFrags operations in a round, varying *t* and *k*.

k	*t*	**Create KFrags (ms)**		**Get CFrags (ms)**	
		Average	Conf Int (95%)	Average	Conf Int (95%)
10	**1**	75.6	(72.11, 79.09)	106.63	(104.79, 108.47)
	2	86.58	(82.07, 91.09)	116.01	(113.49, 118.54)
	3	88.23	(82.38, 94.09)	120.17	(117.04, 123.3)
	4	100.48	(94.42, 106.53)	127.98	(124.23, 131.73)
20	**1**	155.96	(144.22, 167.69)	128.38	(122.94, 133.82)
	2	130.32	(122.91, 137.73)	135.27	(130.74, 139.79)
	3	144.0	(136.01, 152.0)	152.28	(146.56, 157.99)
	4	146.92	(135.99, 157.85)	163.61	(154.72, 172.49)
30	**1**	113.11	(107.89, 118.33)	119.94	(116.93, 122.95)
	2	146.23	(140.54, 151.92)	141.16	(137.83, 144.49)
	3	172.57	(163.51, 181.62)	167.19	(160.77, 173.62)
	4	162.65	(154.41, 170.89)	173.43	(167.14, 179.73)
40	**1**	211.23	(200.45, 222.01)	158.86	(152.58, 165.15)
	2	176.49	(168.25, 184.73)	166.48	(160.42, 172.53)
	3	206.08	(196.19, 215.97)	192.9	(185.59, 200.22)
	4	220.54	(210.67, 230.4)	209.77	(202.55, 216.98)
50	**1**	122.28	(117.61, 126.95)	122.32	(119.94, 124.7)
	2	189.77	(179.35, 200.2)	170.66	(163.35, 177.96)
	3	235.03	(224.69, 245.36)	215.84	(207.61, 224.08)
	4	267.82	(257.65, 277.99)	251.73	(243.17, 260.3)
60	**1**	172.14	(166.32, 177.95)	148.48	(144.76, 152.19)
	2	177.44	(169.55, 185.34)	172.77	(166.75, 178.8)
	3	225.4	(216.35, 234.45)	208.26	(201.29, 215.22)
	4	140.75	(135.36, 146.15)	159.98	(155.94, 164.03)
70	**1**	158.52	(152.33, 164.7)	141.2	(137.57, 144.83)
	2	179.65	(173.0, 186.3)	166.32	(161.58, 171.05)
	3	275.55	(264.45, 286.65)	250.54	(241.68, 259.4)
	4	230.97	(221.41, 240.53)	229.48	(221.51, 237.45)
80	**1**	178.65	(172.19, 185.1)	153.97	(149.92, 158.02)
	2	198.21	(190.55, 205.88)	178.61	(173.34, 183.89)
	3	204.39	(196.89, 211.89)	205.24	(198.95, 211.53)
	4	226.86	(217.05, 236.66)	231.71	(223.5, 239.92)

standard deviation for each operation in a round. The first result that stands out is the large difference in latency between the Request Access and Grant Access operations and the Create KFrags and Get CFrags operations. This is because the first two operations involve writing in the authorization blockchain's ledger. Thus we can already see the impact of the blockchain on the overall system response latency. As can be seen, in general, the t value does not affect the results greatly. On the other hand, as expected, the k value representing the number of data holders is the key factor. A slow but constant increase in the round response latency happens between 10 and 40 holders, starting from 2 seconds latency to 3, for both Request Access and Grant Access operations. After 40 holders, the latency increases faster per number of holders. This seems to be correlated to the fact that a new round is started on average each 3 seconds for each data holder. Thus, if the round takes approximately more than 3 seconds, as from $k = 50$ onward, many more operations start to be executed in parallel. The increase of such parallel executions seems to increase the response latency overall. While the blockchain writing dependant operations are in the order of the thousand milliseconds, i.e., seconds, the KFrags and CFrags operations are in the order of the hundreds. They can be better analyzed using Table 6.8. In both cases, we can see a direct correlation of response latency with the t and k values. With $k = 10$, latency values for the Create Kfrag operation are around 90 ms, while for the Get CFrag operation are around 110 ms. With $k = 80$ values more or less double.

- **System throughput** - Figure 6.11 shows the results obtained considering the round as a single operation, i.e., aggregating the results for every single operation. The figure thus shows the number of rounds per second, i.e., ops/s. The throughput results in more than 0.2 ops/s for the number of holders $k = 10$ and linearly decreases with the increase of k. With $k = 80$ we have on average a throughput of $\sim$ 0.07 ops/s. Even in this case, we notice how the influence of t is almost irrelevant. As we have seen before, t greatly influences the Create Kfrag and Get CFrag operations. However, these two slightly increase the round response latency with respect to the Request Access and Grant Access operations. Indeed, here too, we can see the effect of the blockchain execution in delaying the response time.

- **Threshold number** - Figure 6.12 shows the results when increasing the t value and the number of holders k for each i-th round, i.e., it shows the performances for each subsequent round instead of aggregating all rounds through their mean. In this case, the results show that the increase of t does not influence the overall response delay much. However, this temporal point of view shows the accumulation of delay in the response time when increasing k. We can see, for instance, that up to $k = 30$ each i-th round has more or less the same average latency. When increasing k, however, the latency of rounds in the middle spikes upwards due to the accumulation of operations to perform and then go back to a relatively normal value in the last rounds (i.e., 9-th and 10-th).

6.5.2.3 Discussion

Limited to the scenario we tested, the number holders around 30 and 40 induce the best ratio of completed rounds to response latency time. The system can fulfill around 0.17 rounds per second with this workload. Overall we can observe how the writing in the blockchain dramatically impacts the whole system performance and that the number of requests related only to the TPRE operations can still scale to a larger number of data holders.

In reality, the interaction of holders with the system may be much slower, making the overall round latency increase, but at the same time diminishing the system workload. We can imagine that the *NewRequest* event triggered by the *requestAccess()* method is shown to the data holder through a smartphone notification, thus requiring seconds, if not hours, to be read and accepted. In this context, using semantic web-based policy languages to express rich rules for consent and data requests could be helpful in automating (and thus speeding up) this process (Esteves et al., 2021b). This is left for the next Chapter.

Nonetheless, we argue that the results show the viability of our approach, especially having the possibility to tweak the authorization blockchain parameters and node hardware configuration. Moreover, the good response to the TPRE implementation gives reason to believe that by moving this module to another blockchain that supports smart contracts but provides better latency, even improved outcomes can be achieved.

6.6 Conclusions

In this Chapter, we presented the multi-DLT architecture for a personal information management system (PIMS). The rationale was to complement the previous two proposed systems (personal data space, PDS, in Chapter 4 and decentralized indexing in Chapter 5) with an authorization system. Our PIMS aims to provide individuals with a tool to effectively exercise (at least part of) their rights as in the General Data Protection Regulation (GDPR) and enable data sharing as the newly proposed Data Act intends. In light of the tensions already analyzed between the GDPR and DLTs, we introduced the following architectural components to the PIMS:

- an authorization DLT based on a (semi-)private permissioned ledger, where a set of smart contracts allows data subjects to define access (through an Access Control List) to their personal data stored in their PDS. The access to the data is controlled through two distributed access control mechanisms, i.e., based on secret sharing (SS) and threshold proxy re-encryption (TPRE);
- an audit DLT that consists of a permissionless DLT that provides tamper-proof security to the states of the authorization DLT.

Furthermore, we provided a security and privacy analysis based on a privacy impact assessment. We described the PIMS implementation where the authorization DLT was developed as an Ethereum private blockchain. At first, we compared the differences between SS and TPRE mechanisms; then, we analyzed the execution of a complete TPRE access control scenario experimentally in terms of execution time and system throughput. Results from our performance evaluation show that: (i) TPRE is faster when increasing the size of data to encrypt/decrypt, and it is also more scalable, as it better behaves as the number of nodes and the threshold value increases while executing the protocol; (ii) however, TPRE has the drawback of requiring the generation of re-encryption keys for each new data recipient, while SS does not; (iii) finally, the tests on the full implementation of the authorization DLT show that writing on the ledger represents a bottleneck but that in most use cases, the implementation is viable. Moreover, results beyond the ledger writing part give good reason to believe that a similar approach can be easily implemented in more performing blockchains with much better results.

6.6.1 DLT-agnostic considerations

Our implementation of PIMS is based on the use of an Ethereum network for smart contract execution. However, the multi-DLT architecture presented in this chapter can be considered agnostic to the DLT chosen. In fact, any DLT that supports the issuance of transactions that include arbitrary data can be used as an audt DLT. Any DLT that supports the execution of smart contracts can be considered for the implementation of the authorization DLT. In our case, we chose to implement the latter using an Ethereum permissioned network in which smart contracts are written in Solidity. However, the implementation of a DLT could also be done through a scripting language such as that used in Bitcoin and is not limited to quasi-Turing languages. The execution of key distribution operations is DLT-agnostic, as nodes need only read an address or public key from an immutable ledger in order to operate.

Part III

PRIVACY POLICY, SELF-SOVEREIGN IDENTITY & USE CASES

Chapter 7

Privacy-policy-based Access Control

The content of this chapter is based on the contributions presented in the below-cited publications:

- ISO/IEC JTC 1/SC 29 Technical Committee, *ISO/IEC 21000-23 Information technology — Multimedia framework (MPEG-21) — Part 23: Smart Contracts for Media.* 2022.
- (SUBMITTED) M. Zichichi, and V. Rodríguez-Doncel, *"Encoding of Media Value Chain Processes Through Blockchains and MPEG-21 Smart Contracts for Media,"* IEEE Multimedia, pp. 1-11. IEEE, 2022.

The software produced during the development of this chapter is stored in the following repositories:

- M. Zichichi (2021). Privacy Policy Generator from Smart Contracts. `github.com/miker83z/desp3d-policy-mco-generator`
- M. Zichichi (2021). Privacy Policy Parser for Smart Contracts. `github.com/miker83z/desp3d-policy-mco-parser`
- M. Zichichi (2022). Privacy Policy Smart Contract Templates. `github.com/miker83z/desp3d-policy-web3-templates`

- M. Zichichi (2022). Privacy-policy DLT manager: Manage policies based on MCO and DPV ontologies. DOI: `10.5281/zenodo.7132775`
- M. Zichichi (2022). Client tools to interact with Intelligible suite and Privacy Policy software. `github.com/miker83z/desp3d-client-tools`

In previous Chapters, has been outlined a system that can be considered a self-sufficient enclosed system, meaning that the decentralized PIMS can fully store data and transparently authorize access to them. In this Chapter, we present a way to further empower the abilities of the system's final users by defining a mechanism that enriches the definition of enforceable access policies. In the privacy-policy-based access control layer, we enrich the expressiveness of the access control mechanism to let the data subject express more effectively the privacy policies to be enacted through the access control smart contracts. Policy here is intended as a representation of the will of an individual or organization to grant access to a certain resource: in the case of a privacy-policy this resource would be a private information. Through the authorization DLT, the authorization servers can provide a meeting point through which data access rights can be shared between data holders and recipients in a transparent and verifiable manner. They primarily focus on the subject's consent expressiveness and when recipients need to enforce legitimate data access rights that may take precedence over the subject's ones, e.g., the GDPR's vital interest legal base for data processing. Authorization servers become "privacy proxies" as intended by the Data Protection Working Party (ex Article 29 WP) (Article 29 Working Party, 2014c): "data requests are confronted with predefined policies governing access to data [...]. By defining sensor and policy pairs, third parties' requests for collection or access to sensor data would be authorized, limited, or rejected". The idea is that, instead of needing new consent for each new data processing activity, the subject uses the personal device application to choose technical settings that will then be stored in the authorization DLT in the form of policy. For incoming data access requests, authorization servers respond to the request according to the rules and preferences specified by the subject's policy.

As discussed in Chapter 2 Section 2.4.1, the processing of personal data shall be "processed lawfully, fairly and in a transparent manner" (Article 5, GDPR), where lawfulness can be based on the six legal bases, among which when "the data subject has given consent to the processing of his or her personal data for one or more specific purposes". Such consent must be **easily revocable at any time**, and the controllers that initially gather personal data must always be able to **prove that the subject has consented to their processing**. Moreover, even if the GDPR does not explicitly specify how "specific" is to be interpreted, it should refer clearly and precisely to the scope of the data processing. Any technical representation of consent must, then, allow for codifying **specific purposes** at a sufficient level of detail (Ulbricht and Pallas, 2018). The acceptable forms (as per recital 32, GDPR), apart from oral or written statements, also include the choice of technical granular settings for each of the envisaged purposes (European Data Protection Board, 2020).

In this layer, the privacy-policy-based access control model complements the list-based access control (i.e., the ACL) of the previous layer to support the definition and enforcement of holder-defined policies and recipient-expressed requests for data access. We use a set of Semantic Web technologies and standards for this aim. First of all, we make use of a set of standard specifications to specify policies over assets, i.e., the Moving Pictures Expert Group's (MPEG) ISO/IEC 21000 MPEG-21 framework and the Media Contract Ontology (MCO) and Smart Contract for Media standards (Kudumakis et al., 2020). The reason for using these over some established Right Expression Languages is because the Smart Contract for Media is one of the earliest standard specifications that links rights expressions directly to DLT objects. Second, we integrate those with the Data Privacy Vocabulary (DPV), i.e., a specification that contains taxonomies related to the privacy and data protection domain and specifies terms such as purposes for processing or legal basis (Pandit et al., 2019b). The general idea is to enforce policies, enable access control mechanisms, and maintain an untamperable log of data access related to policies. When a data holder updates policies or resources, the authorization DLT is updated. Each time a data recipient requests, the authorization DLT is updated. Moreover, we also provide the tools for a GDPR "Smart" Data Processing Agreement (DPA). A DPA is an agreement between a data controller and a processor to regulate personal data processing conducted for business purposes. This privacy-policy-based access control layer aims to guarantee a series of features in favor of a transparent process that the final users can completely understand, i.e., intelligible (see Chapter 8. The act of "logging" represents a guarantee for future audits and a way to trace back all data processing activities to the original legal basis enabling it, e.g., the subject's consent.

The original contributions and novelties of this Chapter are described in the following:

- First, the main contribution of this Chapter is the design of a privacy-policy-based access control layer to place on top of the smart contract distributed authorization. These policies are enriched by the rights expression languages used in the ISO/IEC 21000-23 Smart Contract for Media and the W3C Data Privacy Vocabulary standards.

- Second, this access control mechanism is implemented into a PIMS layer that relies on smart contracts and Non Fungible Token (NFT) representation. The implementation includes a novel way of representing policies and tracing personal data exchanges thanks to the use of NFT registries.

- Third, we provide a detailed description of the use of our privacy-policy-based access control layer for the access to personal data based on GDPR's legal basis, namely consent and vital interest.

The remainder of this Chapter is organized as follows. Section 7.1 presents the background concepts behind the proposed architecture and related works. Section 7.2 provides an overview of the MPEG-21 framework and the Smart Contract for Media standard. In Section 7.3, we specify the design of the proposed access control system based on privacy policies. In Section 7.4, the implementation of this access control layer is described through two use cases describing GDPR legal bases for processing. Finally conclusions are presented in Section 7.5.

7.1 Background and Related Work

In this Section, we describe the technologies used to build the proposed policy-based access control layer and introduce the related work.

7.1.1 Non Fungible Tokens

The decentralized applications, i.e., dApps (Buterin et al., 2013), that are possible to build on top of DLTs thanks to smart contracts, exploit the verifiability of information stored on the distributed ledger and authentication based only on cryptographic primitives. This new kind of application created the need for standardized ways of representing information on DLTs. The token representation is one of the most used. It is information recorded on a DLT representing some form of right: ownership of an asset, access to a service, receipt of payment, etc. For instance, the fungible token (Vogelsteller and Buterin, 2015) is one of the most used specifications for creating second-layer cryptocurrencies. The Non Fungible Token (NFT) (Entriken et al., 2018) is a utility token usually implemented to represent and transact with (tangible or intangible) assets on DLTs, where every single token is different from the rest of the tokens, i.e., non fungible. More specifically, NFTs combine both concepts of (i) access rights to an underlying economic value (property) (Caglayan and Ozkan, 2021; Fairfield, 2021), and (ii) permission to access someone else's property or services or collective good. The asset considered here can be of many forms: (i) physical property, e.g., houses or unique artwork, (ii) virtual collectibles, e.g., unique pictures or collectible cards, (iii) negative value assets, e.g., loans, burdens, and other responsibilities. In general, NFTs are distinguishable, and the ownership of each one is tracked separately.

7.1.2 Semantic Web technologies

Semantic Web technologies (Berners-Lee et al., 2001) bring structure to the meaningful contents of the Web by promoting common data formats and exchange protocols. Linked Data is the form of its most successful incarnation: data are published in a structured manner so that information can be found, gathered, classified, and enriched using annotation and query languages. The World Wide Web Consortium (W3C) has published over the last twenty years a set of specifications to describe resources that simultaneously address these two design goals: those of the Semantic Web. Whereas these specifications were born to represent data on the Web, their use has gone beyond, and today many applications run offline but using the semantic web specifications. The most spread paradigm to represent information is RDF (Resource Description Framework). In this framework, resources are identified with URIs and described with collections of triples. The precise meaning of each resource can be formally established with OWL ontologies. An ontology is a formal representation of knowledge through a set of concepts and relations between these concepts within a specific domain. Through these ontologies, it is possible to convey the meaning of data, facilitating cross-domain applications and services. Ontologies in these scenarios effectively act as data models. Whereas new ontologies can be created whenever necessary, a set of *de facto* standard ontologies should be reused whenever possible. For example, there are ontologies to describe the basic personal contact information, such as vCard (Iannella and McKinney, 2014), or to represent policies (Ianella, 2007) or contracts (Rodríguez-Doncel et al., 2016). Other vocabularies and ontologies have recently appeared in the privacy and data protection domain (Esteves and Rodríguez-Doncel, 2022; Palmirani et al., 2018a; Pandit et al., 2018), with a special focus on representing GDPR terms Pandit et al. (2019a); Robaldo et al. (2020). These technologies bring two main advantages: "interoperability" and "reasoning". First, the ontologies above are recommended by the W3C and thus universally understood. Second, reasoning with the information represented using these data models is easy because they are mapped in a formal language. For instance, take an individual that uses the above technologies to state that his or her data must not be transacted whenever he or she finds within the United Kingdom. If properly connected to other datasets, a system knowing that the individual is in London will infer that the individual is also in the United Kingdom and should not transfer any more data.

7.1.3 Rights Expression Languages

Rights Expression Languages (RELs) are a central component of contemporary digital rights management systems (Pellegrini et al., 2018). They are applied to express permissions, obligations, and prohibitions in a machine-readable form. In this work, we are interested in their use of RELs to express policies, with particular emphasis on privacy policies but not only (Kirrane et al., 2018). The expression of policies, in this case, defines a relationship between subjects and targets within a policy domain. RELs, indeed, can be used for access control purposes and also for license management or contracting. (Pellegrini et al., 2018) propose a classification to understand their functionalities and applications, giving an outlook on how RELs are used to explicate machine-readable rights for access control, trust management, and contracting.

Among the most prominent RELs, we find the eXtensible Access Control Markup Language (XACML), the Open Digital Rights Language (ODRL), and the MPEG-21 framework. XACML (Gaaloul et al., 2008) is an OASIS standard that includes a declarative fine-grained attribute-based access control policy language, an architecture, and a processing model describing how to evaluate access requests according to the rules defined in policies. ODRL (Ianella, 2007), on the other hand, is a W3C standard that provides an information model, a vocabulary, and encoding mechanisms for representing statements about the usage of content and services. It is based on the use of Semantic Web technologies to simplify the distribution, sharing, and exploitation of statement information across the Web. Indeed, it can be argued that semantic web technologies can contribute to more intelligent and flexible handling of privacy, security, and policy issues, through supporting information integration and sense-making (Kirrane et al., 2018). Esteves and Rodríguez-Doncel (2022) surveys existing vocabularies, ontologies and policy languages that, within the semantic web, can be used to represent informational items referenced in GDPR rights and obligations. Semantic web technologies are, for the above reasons, also included in the MPEG-21 framework (Burnett et al., 2003). MPEG-21 describes an abstract content capsule, the Digital Item, and the means for its identification, description, adaptation, verification, and quality assessment. Other parts of the standard are concerned with the intellectual property of the works conveyed in the Digital Item. A REL and a Rights Data Dictionary represent under which conditions the intellectual property works can be consumed and transmitted in the context of a Digital Rights Management platform (Wang et al., 2005).

7.1.4 Related Work

Many scholars have worked on modeling policies for the informed consent of final users, especially for semantically modeled policies. Some work has also been carried out in the context of DLT usage or, in general, for the creation of PIMS. At this point, it is essential to say that, even if several related works successfully use RELs such as ODRL and XACML concerning decentralized systems, we use the MPEG-21 framework because it includes one of the earliest (if not the only one) standard specifications that links RELs directly to DLT objects, i.e., the ISO/IEC 21000-23 Smart Contract for Media (ISO/IEC IS 21000-23, 2022; Kudumakis et al., 2020).

7.1.4.1 Consent and non-DLT-based PIMS that exploit policy-based access control

Several works use policy-based access control as the primary means of protecting personal data. For instance, (Di Cerbo et al., 2018) propose an automatically-enforceable policy language for access and usage control of personal data that aims at transparent and accountable data usage. (Chhetri et al., 2022) presents a tool for automated compliance verification and auditability based on informed consent modeled with a knowledge graph.

While the general tendency to design a PIMS is to adopt distributed technologies, some solutions are not based on the use of DLTs. (Katevas et al., 2020) provide a novel use of the databox model for fine-grained access to various personal data where the control is based on the Ancile platform (Bagdasaryan et al., 2019). It is a trusted computing environment that allows third-party services to perform data calculations, enforcing data use-based privacy policies. The databox model is mainly symbolic at the moment, but it is not a theoretical model. A non-direct link to this model, which puts into practice the concept of SSI, is Solid (Sambra et al., 2016). The project was born to give users their data sovereignty and let them choose where their data resides and who is allowed to access and reuse it. Solid involves using distributed technologies and Semantic Web integration in social networks. Semantic Web technologies are used to decouple user data from the applications that use this data. Data is stored in an online storage space called Pod, a Web-accessible storage service that can be deployed on personal or public servers.

In light of the development of Solid, (Esteves et al., 2021a) focus on the notion of GDPR consent and provide a solution based on exploiting ODRL policies. In their solution, the need to ensure informed and explicit consent led to the inclusion of specific information items in the Pod so that the users can access their consent authorizations. Furthermore, their Pod implementation has methods enabling users to update or revoke the consent previously given. In line with our work, the authors debate whether implicit consent from established user preferences is enough to provide automated access to personal data. The act of "choosing technical settings for information society services" (European Data Protection Board, 2020) can be considered equivalent to letting users choose which data types and purposes they consent to enable automation.

7.1.4.2 Policy-based access control used in DLTs and smart contracts

Even if non-DLT-based PIMS can often be considered a distant topic for systems such as Solid (often ostracized by some researchers in the Semantic Web world), it has been argued that their combination can provide more features to the final users. (Kongruangkit et al., 2021) believe that DLTs can provide a platform through which data access rights can be shared between users and service providers in a transparent and verifiable manner. It is the case especially when service providers need to enforce legitimate data access rights that may take precedence over users' ones. The authors propose a hybrid access control scheme that supports the definition and enforcement of local, i.e., user-defined, and global, i.e., service provider-expressed, data access policies. (Ramachandran et al., 2020) and (Cai et al., 2020) proposed a framework to store data generated by IoT devices in Solid with a DLT for validation purposes. Through an authentication mechanism, any third-party application can gain access to the data in the Solid Pod and verify the authenticity of the data by cross-checking the hash of the data on the DLT.

Other scholars use Semantic Web technologies to create new ontologies for declaring privacy policies and then integrate a DLT. It is the case of (Banerjee and Joshi, 2017) that provides an architecture in which the DLT is used to provide a purpose-centric access-control model, and of (Mahindrakar and Joshi, 2020), that focuses explicitly on the GDPR. Other studies in the literature for GDPR compliance do not address key distribution and primarily focus on programming smart contracts for automatically managing access control policies (Davari and Bertino, 2019; Hawig et al., 2019; Koscina et al., 2019; Molina et al., 2020). (Davari and Bertino, 2019) designed a model providing access control in which only authorized parties with user consent can access user data and where all activities are recorded in a DLT. Their access control system uses policies declared based on an ontology that extends the Provenance Ontology (PROV-O) (Lebo et al., 2013) and the XACML model (Gaaloul et al., 2008). With a model following a similar approach to data interdependence, the authors in (Ahmad et al., 2017) propose an access control in Online Social Networks based on delegation. Truong et al. (Truong et al., 2020) provide a DLT-based GDPR-compliant personal data management solution where consent is handled through a token and data is stored and served through a Database Management System. In their work, (Daudén-Esmel et al., 2021) provide another DLT-based PIMS, focusing on data controllers and processors and how they fulfill the GDPR main requirements by demonstrating that they have the authorization to collect/process subjects' personal data.

7.2 The MPEG-21 framework and the Smart Contract for Media

In this Section, we will introduce the MPEG-21 framework and its RELs, since our proposed policy-based access control layer is based on them. In particular, we will define the languages and ontologies within the MPEG-21 framework that facilitate the conversion of media narrative contracts to digital ones and enable the creation of new policy sets and contracts in machine-readable electronic formats.

7.2.1 MPEG-21 framework and its RELs

MPEG (Moving Picture Experts Group), a working group of ISO/IEC, has developed several well-known media encoding standards for audio, video, and genomic information. One of its endeavors is the definition of a Multimedia Framework, known as ISO/IEC 21000 or MPEG-21. The earliest parts of MPEG-21 allowed the representation of rights in a machine-readable form, i.e., using the eXtensible Markup Language (XML). This significant advance enabled access control to audiovisual works in various information systems. The other parts of MPEG-21 leveraged the benefits of the Semantic Web to improve the interoperability of rights expressions, precisely define data models using computer ontologies, and enable description-logic-based authorization algorithms.

7.2.1.1 MPEG-21 policies and contracts representation

The first part we will describe is the part that supports the detailed description of the media value chain, i.e., Media Value Chain Ontology (MVCO) (Rodriguez-Doncel and Delgado, 2009). MVCO is an ontology used to describe the main entities in the media value chain formally: (i) IP entities, which are the objects subject to copyright law protection such as works (e.g., an original song), manifestations (e.g., its music score), instances (e.g., the performance of the song), or products (e.g., a sellable item); (ii) relevant actions that can be performed on those entities (e.g., adapt an original work, perform a specific work), and (iii) types of users whose actions are rights, obligations, or something else provided by IP law (e.g., creator, producer). The Audio Value Chain Ontology (AVCO) extends MVCO functionalities related to the description of composite IP entities in the audio domain (Kudumakis et al., 2020).

MVCO is supplemented by the representation of contracts transacting content rights, i.e., Media Contract Ontology (MCO) (Rodríguez-Doncel et al., 2016). MCO builds on MVCO's generic deontic statements (incorporating the concepts of permission, prohibition, and obligation) by providing the elements to shape the structure of media contracts (*mco-core*), to express rights to exploit media content (*mco-ipre*) and to define specific obligations for payments and notifications (*mco-pane*) (Rodríguez-Doncel et al., 2016).

In the MVCO, AVCO, and MCO cases, the use of RDF is involved. However, the MPEG-21 also includes a part involving the XML, i.e., the Contract Expression Language (CEL) (Rodríguez et al., 2015). This part can be considered equivalent to the combination of MVCO and MCO for expressing rights. CEL provides an extensible model for representing generic agreements between parties (*cel-core*) and defines the most common acts and constraints in the media field and is used in digital media contracts (*cel-ipre*) (Rodríguez et al., 2014).

Music and media value chain actors can use MPEG-21 CEL/MCO standards to share and exchange, in an interoperable manner, all metadata and contractual information related to creative works, leading to transparent payment of royalties. Furthermore, ontologies enable functional inference and reasoning capabilities to derive knowledge and data through facts and logic based on rich semantic copyright models.

Finally, the newest part of MPEG-21, titled Smart Contracts for Media (SCM), builds upon the MVCO, MCO, and CEL, and it is an ISO/IEC International Standard (IS) (ISO/IEC IS 21000-23, 2022). It is the work we will refer to for the rest of the Chapter. Before going into the detail of its specification (in the following Sub-Section), we will first describe the MCO/CEL objects that the SCM exploits.

7.2.1.2 MPEG-21 CEL/MCO objects

Narrative contracts usually share a common structure that consists of a preamble and a body. The MPEG-21 machine-readable contracts expressed in CEL and MCO/MVCO (that, from now on, we will refer to them as MPEG-21 CEL/MCO) are based on this shared structure (Kudumakis et al., 2020). The MPEG-21 MCO/CEL contract consists of a series of objects found within the contract structure. We firstly have a main *Contract* object that includes a preamble with:

- contract metadata (e.g., date, version, title);
- contract unique identifier;
- possible relationships with other *Contract* objects (e.g. amendments, prevalence or substitution);
- *Party* objects, representing signatory parties for which the contract is binding.

The *Contract* object also includes the body with:

- the *IP Entity* objects, such as an original work or a music performance, and whose rights are traded in the contract;
- the operative part containing the contact information in the form of *Deontic Expression* objects such as permissions, obligations, and prohibitions. The *Deontic Expression* (or clause) includes:
  - an *Action* object, i.e., the right;
  - a set of *Fact* objects, logically combined (i.e., using union and/or intersection operators) representing the conditions that must be satisfied;
  - an *IP Entity* object, i.e., the media (digital or not) which is the object of the right;
  - a *Party* object, representing the party the *Action* is related to.

  Deontic Expressions can be related among them. For instance, a party has an obligation of payment after broadcasting some media as specified in one of the contract's permission.

7.2.2 Smart Contract for Media

The Smart Contract for Media specification is a passthrough component designed to be the interlingua that connects the MPEG-21 framework with different DLTs. Thus, it was designed to contain a set of tools interoperable with different types of DLTs

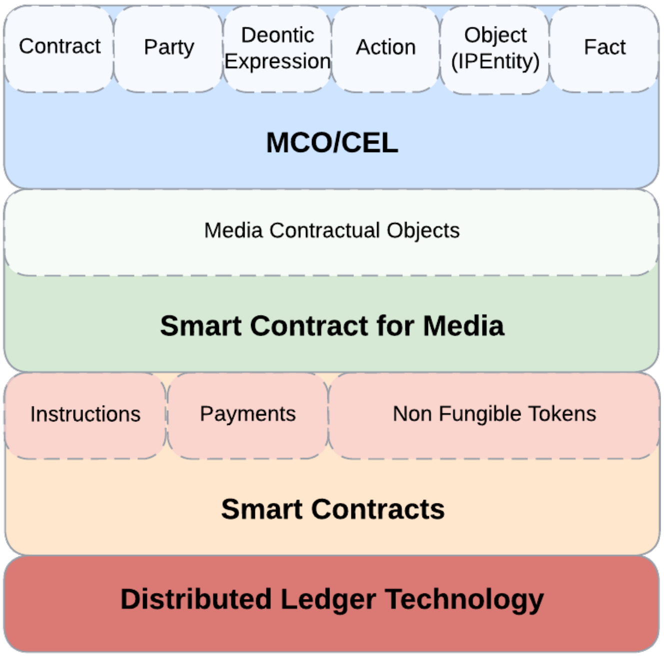
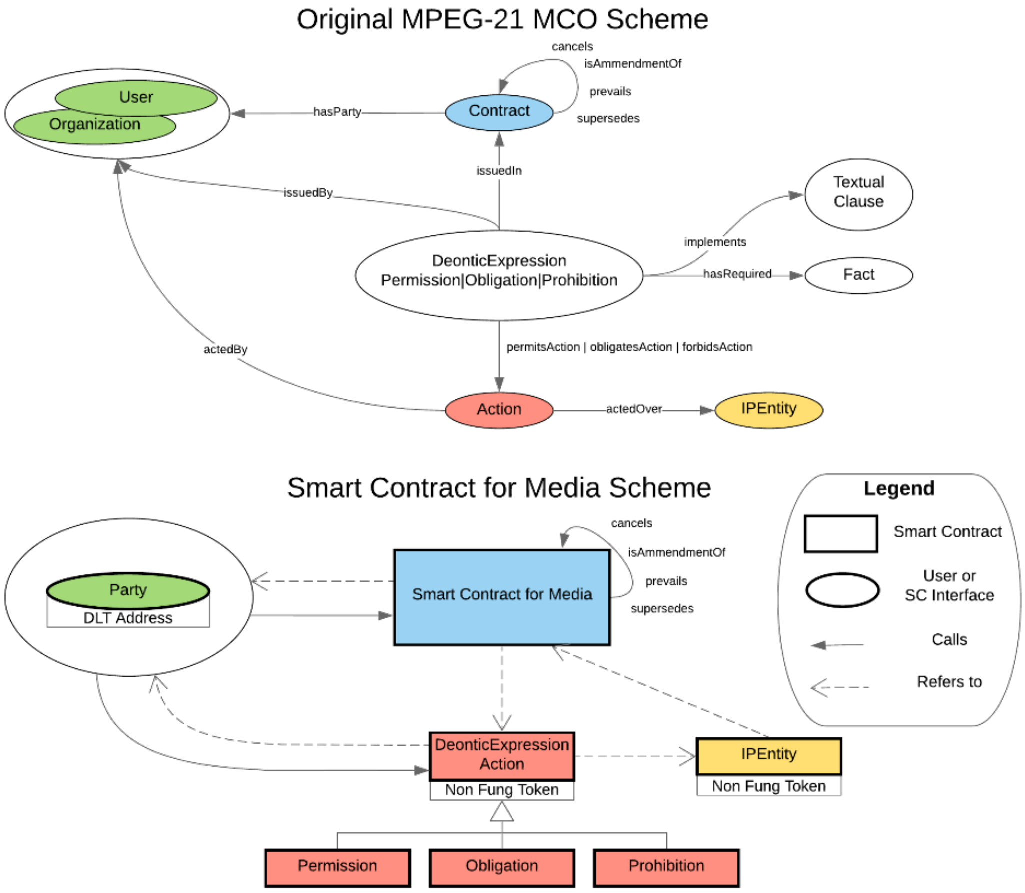


Figure 7.1: Layered view of the technologies and frameworks behind the Smart Contract for Media.

and for use in different contexts inherent in the media value chain. In combination with the MPEG-21 framework, smart contracts can be used to encode the terms and conditions of a contract for media-related asset trading. Smart contracts can be used to establish and enforce agreements such as licenses and enable the transmission of real-time access to content recipients. Rights information is protected content, then can be encoded using the MPEG-21 framework and directly and uniquely linked to a smart contract, i.e., an SCM. In other words, smart contracts could allow content policies to be administered almost instantaneously and manage usage allowances and restrictions. The SCM instructions are encoded according to agreed policies and conditions and executed as soon as an asset has to be accessed.

Figure 7.1 shows a layered view of the environment in which the SCM is executed. Its position is central with regard to the layers related to MPEG-21 MCO/CEL media contracts and the DLT. In particular, the SCM exploits several elements of the media contracts, such as Contract, Party, and Deontic Expression, and encodes new information in a DLT through smart contracts' instructions and NFTs. In this Section, we will go through each layer shown in Figure 7.1 (top to bottom), all of which constitute the SCM.



MPEG-21 CEL/MCO objects represented through the MCO schema in comparison with the SCM Media Contractual Object representation in a DLT. Original scheme taken from the SCM standard specification and edited (ISO/IEC IS 21000-23, 2022).

Figure 7.2: Diagrams comparing MPEG-21 CEL/MCO and the SCM.

7.2.2.1 The Media Contractual Objects of a Smart Contract for Media

The SCM is a smart contract that includes or refers to metadata and contractual information connected to creative works, i.e., media, and encodes a contract's terms and conditions. These metadata are called Media Contractual Objects and consist of elements already encoded using the MPEG-21 MCO/CEL representations seen in the previous Sub-Section. The use of the Media Contractual Objects in the SCM specification can be seen in Figure 7.2. The semantic and operational scopes of the original narrative contractual information are bounded to the ones provided using the MPEG-21 CEL/MCO media contracts. This aspect reflects the concept of legal isomorphism, i.e., a clear correspondence between items to be found in the source material (narrative contract version) and the representation of the information they contain in the system (SCM environment) (Bench-Capon and Coenen, 1992).

The specification in the ISO/IEC 21000 Part 23 (ISO/IEC IS 21000-23, 2022) describes a conversion process that takes a MPEG-21 CEL contract or MPEG-21 MCO contract as input and outputs a standardized set of Media Contractual Objects. These objects are unique for both CEL and MCO implementations, thus allowing too to pass from one encoding to the other. Media Contractual Objects maintain the same high-level objects definition as the one shown in Sub-Section 7.2.1.2, i.e., *Contract, Party, Deontic Expression, Action, IP Entity, Fact*, and their specializations too, e.g., a *Payment* object is a specialization of an *Action* object.

The following describes how the Media Contractual Objects are used in the SCM.

Contract The *Contract* object is the one that includes or refers to the digitalized contractual information extracted from a narrative contract, i.e., the structure, including the preamble and body. A manifestation of a *Contract* object is a unique Smart Contract for Media deployed in a specific DLT. Interoperability of data stored on the DLT can be achieved using simple references, i.e., a smart contract implementing an SCM can reference another SCM through a DLT's smart contract address.

Parties The *Party* object represents a human or juridical person bound to the narrative contract. Since identities in DLTs are generally represented through addresses, a *Party* is represented and authenticated in the SCM through a DLT address that, thus, represents this *Party*.

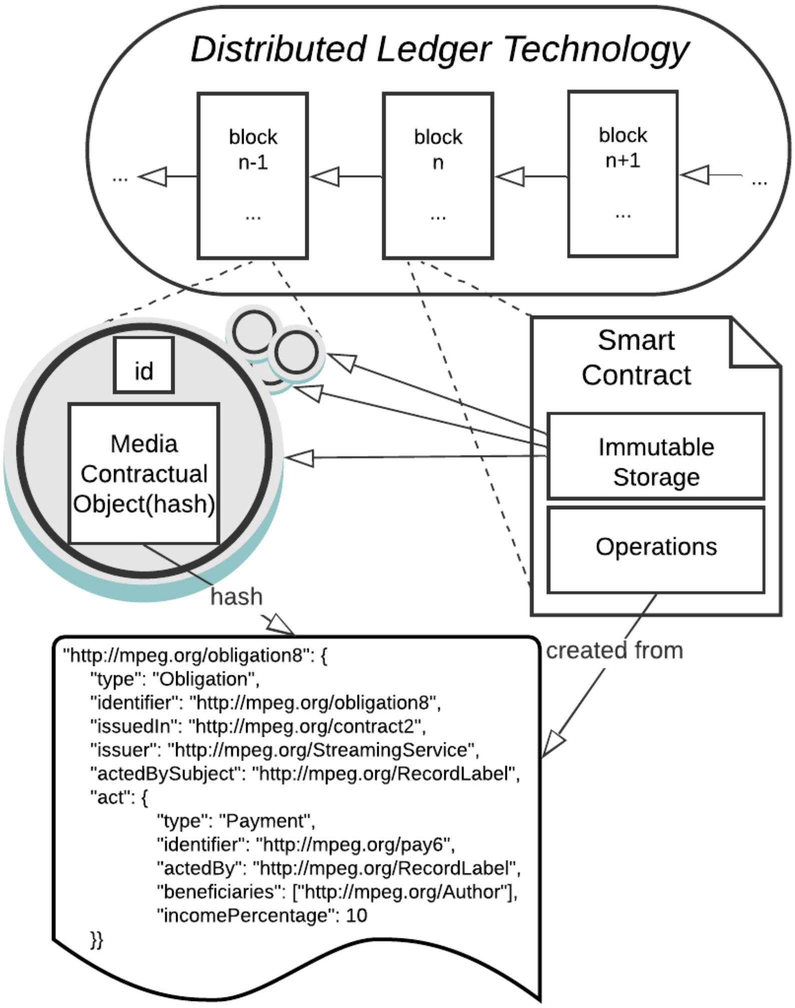
IP Entity The *IP Entity* object encapsulate one or more digital items of intellectual property in the MPEG-21 multimedia framework. Within the scope of a specific SCM contractual information, *IP Entity* objects are uniquely identified on-chain through the use of NFTs. Then, the entire set of information related to a specific *IP Entity* object is linkable to such NFT. Two reasons support this approach: (i) the linkage between IP entities and related SCM is maintained at a high level, particularly when DLTs offer append-only data storage and not a more complex one; (ii) it makes feasible the process of auditing, exploiting at best the immutability feature of DLTs; for the history of all operations executed over an *IP Entity* object, indeed, can be found in one place. For instance, an *IP Entity* object can be represented for the first time by a single NFT with an id equal to `N` in an SCM with an address equal to `X`. Then, another SCM with address `Y` that references that specific *IP Entity* object can reference the id `N` without creating a new NFT (Figure 7.3). Finally, an *Event* object is also considered an IP entity and represents an event that can influence a deontic expression.

Deontic Expression The *Deontic Expression* object is included in the body of a *Contract* object and encompasses the properties of an agreed machine-readable contract clause regulating parties' actions and rights. The uniqueness of such an agreement leads to following the same approach used for *IP Entity* objects, i.e., clauses are serialized according to the concept of NFT. The reasons for supporting this approach are: (i) it enables a unique way for storing clauses in DLTs, that is also beneficial in terms of interoperability, in terms of sharing these clauses with other DLT-based applications; (ii) it allows the transfer of value in the form of obligations, permissions and prohibitions, similarly to how cryptocurrency transfers are done.

7.2.2.2 Smart Contract and Distributed Ledger Technology

From a practical point of view, the SCM can be considered an interface that makes the MPEG-21 framework interoperable with several DLT implementations. Figure 7.3 shows graphically how the SCM can be subdivided. We can first consider the SCM as a tool to implement and passively enact the operational part of the original narrative media contract. Secondly, we can consider the SCM as immutable storage for the above-referenced MPEG-21 CEL/MCO machine-readable contract information.

Instructions One of the roles adopted by the SCM is to directly and passively enact what is "enactable" (i.e., enforceable) in a DLT, with reference to the clauses indicated in the media contract. We need to elaborate and clarify three points of the previous sentence to capture this other role of the SCM fully:



```
"http://mpeg.org/obligation8": {
    "type": "Obligation",
    "identifier": "http://mpeg.org/obligation8",
    "issuedIn": "http://mpeg.org/contract2",
    "issuer": "http://mpeg.org/StreamingService",
    "actedBySubject": "http://mpeg.org/RecordLabel",
    "act": {
            "type": "Payment",
            "identifier": "http://mpeg.org/pay6",
            "actedBy": "http://mpeg.org/RecordLabel",
            "beneficiaries": ["http://mpeg.org/Author"],
            "incomePercentage": 10
    }}
```

Figure 7.3: Smart Contract for Media Structure

1. The clauses indicated in the media contract are represented as *Deontic Expression* objects, and the implementation of instructions in the form of smart contract methods are derived from these. For instance, a specific *Payment* object, i.e., a specialization of a *Deontic Expression* object, leads to the creation of a specific payment smart contract method.

2. However, the set of actions that can be included in a media contract is greater than the set of actions that are “enactable” in a DLT’. It means, for instance, that an *Obligation* object in a contract might limit the exploitation of media in a specific country, but an SCM cannot enact the operation of verification of the exploiter

location because the DLT protocol does not allow it (Finocchiaro and Bomprezzi, 2020). This point heavily depends on the implementation of the DLT and related services, e.g., the use of oracles and proof-of-location might enable a location verification operation (Amoretti et al., 2018).

3. Finally, the 'directly and passively enact' refers to the abilities that smart contracts generally offer. A smart contract passively enacts an operation because it does not "run in the background" and automatically activates itself when needed. However, an actor (that can generally be whoever) has to "wake up" the smart contract. A smart contract directly enacts an operation because everything needed for its execution is stored on the ledger and can be validated; if a clause's condition is met on the ledger, a consequent action can be directly triggered.

NFTs and Immutable Storage The second role of the SCM is to crystallize the data encoded using the Media Contractual Object. This is due to the native immutability feature that the DLTs' ledger generally provides. Thus, once the SCM enters into action, i.e., it is deployed to the DLT, each piece of information related to the original contract can be validated against the stored SCM data, e.g., the address of a party or the fingerprint of a digital media.

Each Media Contractual Object is then stored in the SCM according to what was discussed in the previous subsection. In particular, each *IP Entity* and *Deontic Expression* is stored in a unique NFT, while the rest of the objects are stored in the SCM using an ad-hoc data structure, e.g., a hash map. To be noted is also the fact that the MPEG-21 *Contract* object preamble might include the narrative contract text version, too, in the form of an object or at clause level, making thus explicit the legal isomorphism.

We stress that NFTs are already used for encoding unique works resulting from human creativity and innovation, i.e., what intellectual property rights generally protect, that is the case of an *IP Entity* (Bamakan et al., 2021). However, what is not generally trivial is the use of NFTs to encode information related to the ownership of certain rights, such as permissions, obligations, and prohibitions. Thanks to the *Deontic Expression* object representation, we can create referable rights and duties and save the association between this reference and the relevant party directly in the ledger in an immutable way through NFTs. This same property is what we will exploit in the following Sub-Section to create the privacy-policy-based access control layer.

Table 7.1: Relationship between MPEG-21 CEL/MCO, SCM, and DPV objects.

Narrative contract object	Media Contractual Object	Description	SCM representation in a DLT	Objects that are sub-classes in DPV
Preamble and body	*Contract*	includes or refers to a set of digitalized information	Smart Contract	–
Contract parties	*Party*	human or juridical persons bound to the narrative contract	DLT Address	*DataSubject*, *DataController*, *DataProcessor* etc.
Content that is the subject of the contract	*IP Entity*	digital items and entities representing content	NFT	*Data*, *PersonalData*, *NonPersonalData* etc.
Policies and contract clauses	*Deontic Expression*	machine-readable clause/policy regulating parties' actions and rights	NFT	–
Action specified in a contractual clause or policy	*Action*	action permitted, obliged or prohibited to a party	Within a Deontic Expression NFT	*Processing* and all its sub-classes, e.g.,*Collect*, etc.
Fact specified in a contractual clause or policy	*Fact*	representing conditions to be satisfied	Within a Deontic Expression NFT	*LegalBasis* and sub-classes, e.g.,*Consent*, *Purpose* and sub-classes, e.g.,*Advertising*, *PersonalData Handling*, *Sector*

7.3 Privacy-policy-based access control layer design

This Section describes the privacy-policy-based access control layer concerning integrating a policy declaration system based on the MPEG-21 SCM with the decentralized PIMS presented in Chapter 6.

7.3.1 From media contracts to privacy policies

Up to now, we have discussed using policies concerning media contracts. We describe how this is linked to the personal data sharing policies case. First, we need to clarify that we intend to do something other than equate Intellectual Property regulations with Personal Data Protection ones. Although it can be argued that the economic characteristics of data sharing are comparable to the sharing of intangible intellectual property, property rights over data are still only partially defined, and this topic is currently much debated in the law community (Ciani, 2018; Trakman et al., 2019).

We argue that if a piece of personal data can be stored electronically and uniquely identified, then another piece of information that is also stored electronically can be associated with it as a policy that allows access to it. From a strictly practical standpoint, the piece of personal data can be conceptually treated as an intellectual property entity. For example, both can be represented by a file, and that file can be associated with a designated access policy for its use in a computer operating system.

Second, this layer is intended to overcome the limitations of the Access Control List (ACL) we proposed in our PIMS. Indeed, the ACL authorizations are sufficient for establishing access policies about an identified party. However, these cannot be used to specify prohibitions or obligations over data and cannot define more complex rules. For this reason, we extend the ACL with MPEG-21 CEL/MCO policies that specify permissions and prohibitions and integrate the DPV to align the access control to data protection and privacy concepts. Media Contractual Objects such as *Deontic Expression*s and *Action*s profile enable to express granular access control policies, while the DPV enables the modeling of specific data protection terms, such as the subject's consent.

7.3.1.1 Privacy Policy Objects: mapping Media Contractual Objects to the W3C Data Privacy Vocabulary

The DPV provides top-down taxonomies based on the GDPR but is intended to be jurisdiction-independent (Pandit et al., 2019b). In such a way, it can represent personal data processing practices in terms of categories of personal data, purposes, processing, technical and organizational measures, legal entities, legal bases, rights, and risks (Esteves et al., 2021a).

Table 7.1 shows the MPEG-21 CEL/MCO (high-level) objects and their description on the left. Each object has a representation in terms of DLTs thanks to the Smart Contract for Media specification. We stress that this specification is one of the earliest ones that bring a widely used REL to the DLT context. Thus, we decided to associate high-level objects of the DPV to Media Contractual Objects for two reasons: (i) to

Table 7.2: Namespaces used in the RDF listings

Prefix	Namespace
rdfs	`http://www.w3.org/2000/01/rdf-schema#`
mvco	`http://purl.oclc.org/NET/mvco.owl#`
mco-core	`urn:mpeg:mpeg21:mco:core:2015#`
dpv	`http://www.w3.org/ns/dpv#`

The namespaces associate a prefix to a URI representing a specific schema.

include a comprehensive privacy and data protection terminology that can exploit the MPEG-21 CEL/MCO policy making; (ii) to have a standardized direct link with DLTs elements such as addresses and NFTs.

The model we will use to create Privacy Policy Objects consists of a subset of Media Contractual Objects and DPV objects. We firstly have a main *Contract* object that includes a preamble and a body, as in MPEG-21 MCO/CEL. This has no direct links to a DPV equivalent but will be used later in this subsection to represent a "Smart" DPA. The other high-level objects are:

- the *Party* is a super-class of a series of DPV objects representing a natural person or a legal entity, e.g., *Data Subject*, *Data Controller*, *Data Processor*;

- the *IP Entity* is a super-class of DPV objects representing content that, in our case, is *Data*, *Personal Data*, *Non-Personal Data* and all the sub-types;

- the *Deontic Expression* has no direct link to a DPV object, but its information includes:
  - the *Action* is a super-class of DPV objects representing processing activities, e.g., *Collect*, *Share*, *Transform*;
  - the *Fact* is a super-class of a series of DPV objects for representing legal bases, e.g., *Consent*, the processing purpose, e.g., *Advertising*, the properties of a personal data handling and other miscellaneous DPV terms, e.g., the *Sector* of a processor company.

Listing 7.1 represents an example of the use of Privacy Policy Objects for location data sharing policy with a constraint of targeted advertising in social media only. In this listing and the following ones, we will use the namespaces depicted in Table 7.2.

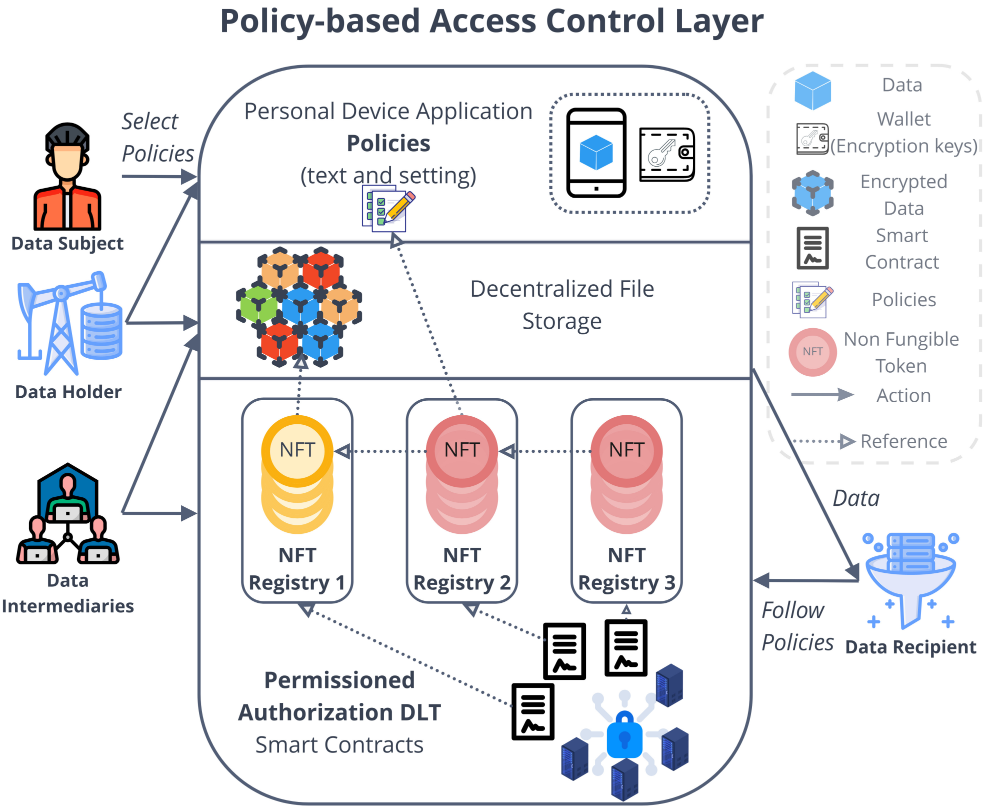
```
<uri:txt001>
        a                  mco-core:TextualClause ;
        mco-core:text      "Location data read-only policy for
                            Targeted Advertising in Social Media" .
<did:iid:holder1>
        a           dpv:DataController ;
        rdfs:label  "Data Holder" .

<did:iid:subject1>
        a           dpv:DataSubject ;
        rdfs:label  "Data Subject" .

<did:nft:eip155:1_erc721:0xa437b30051601bd54ffee7de357b28e1488929rt_32>
        a               dpv:PseudoAnonymisedData .

<did:nft:eip155:1_erc721:0xa437b30051601bd54ffee7de357b28e1488929rt_43>
        a               dpv:SensitivePersonalData .

<did:nft:eip155:1_erc721:0xa437b30051601bd54ffee7de357b28e1488929rt_1>
        a                       dpv:PersonalData ;
        mvco:isMadeUpOf         <did:nft:eip155:1_erc721:0xa4...29rt_32>,
                                    <did:nft:eip155:1_erc721:0xa4...29rt_43> .

<did:nft:cnsnt_givn1>
        a                       mco-core:Event ;
        mvco:hasRightsOwner     <did:iid:subject1> ;
        mco-core:makesTrue      <uri:aef001> ;
        rdfs:label              "Subject's consent given event
                                    (can be withdrawn)" .
<uri:aef001>
        a               mvco:ActionEventFact .

<did:nft:per001>
        a                       mvco:Permission ;
        mco-core:implements     <uri:txt001> ;
        mvco:issuedBy           <did:iid:subject1> ;
        mco-core:permitsAction  <uri:act001> ;
        mco-core:hasRequired    <uri:fac001>.

<uri:act001>
        a                   dpv:Share ;
        mvco:actedBy        <did:iid:holder1> ;
        mvco:actedOver      <did:nft:eip155:1_erc721:0xa437b...488929rt_1> ;
        mco-core:makesTrue  <uri:aef002> .

<uri:fac001>
        a               mvco:FactIntersection ;
        mvco:hasFact    <uri:aef001>,
                            <uri:con001> ;

<uri:aef002>
        a               mvco:ActionEventFact .

<uri:con001>
        a                       dpv:Consent ;
        dpv:hasDataSubject      <did:iid:subject1> ;
        dpv:hasDataController   <did:iid:holder1> ;
        dpv:hasPurpose          <uri:repo001>,
                                    <uri:repo002> ;
        dpv:hasProcessing       <uri:repo003> .

<uri:repo001>
        a               dpv:SocialMediaMarketing .

<uri:repo002>
        a               dpv:TargetedAdvertising .

<uri:repo003>
        a               dpv:Consult .
```

Example 7.1: RDF representation of a location data sharing policy for Targeted Advertising in Social Media

The above example represents concepts in RDF, namely in order:

- a *subject*, represented by a Unique *ID* in angle brackets "<>";
- a *predicate* applied to the *subject*, represented using an ontology property, e.g. *mvco:issuedBy*;
- the *object* of the *predicate*, represented by an object of the ontology (or by a *URI* that links to this one).

In this example, some *URI*s (Berners-Lee et al., 2004) starts with "*did:*," but we will see why in Chapter 8. For now, we use these kinds of *URI* to refer to unique parties "*did:iid*" or piece of data "*did:nft*". The other *URI*s identify facts or actions or the remaining kinds of Privacy Policy Object. Then, we have the primary element of interest, i.e., the object with *URI* equal to `did:nft:per001`. It represents a MPEG-21 *Permission* object issued by the data subject, i.e., `did:iid:subject1`, to perform the DPV *Share* action, i.e., `uri:act001`, but requiring the DPV *Consent* with *URI* `uri:con001`. The *Share* action can be acted by the data holder, i.e., `did:iid:holder1`, but with the conditions indicated



The top-right legend identifies the elements in the diagram. Solid arrows represent interactions between an actor or a set of actors and a system or network. Dashed arrows represent hash pointers to elements.

Figure 7.4: Components of the privacy-policy-based access control layer.

in the *Consent*, i.e., when the processing activity after sharing the data is of type DPV *Consult* and with DPV *SocialMediaMarketing* and *TargetedAdvertising* purposes. Note, too, that the textual clause is accompanied by the Deontic Expression, making a strong link between the narrative and execution parts of the policy in this case and contracts in others (we will see why this is important in Chapter 8).

7.3.2 Components Design

This Section describes the components that form the privacy-policy-based access control layer. These are graphically described in Figure 7.4, showing their relations and interaction with actors. In particular, components include: (i) the personal device application for setting the policies and the DFS for storing them; (ii) the authorization

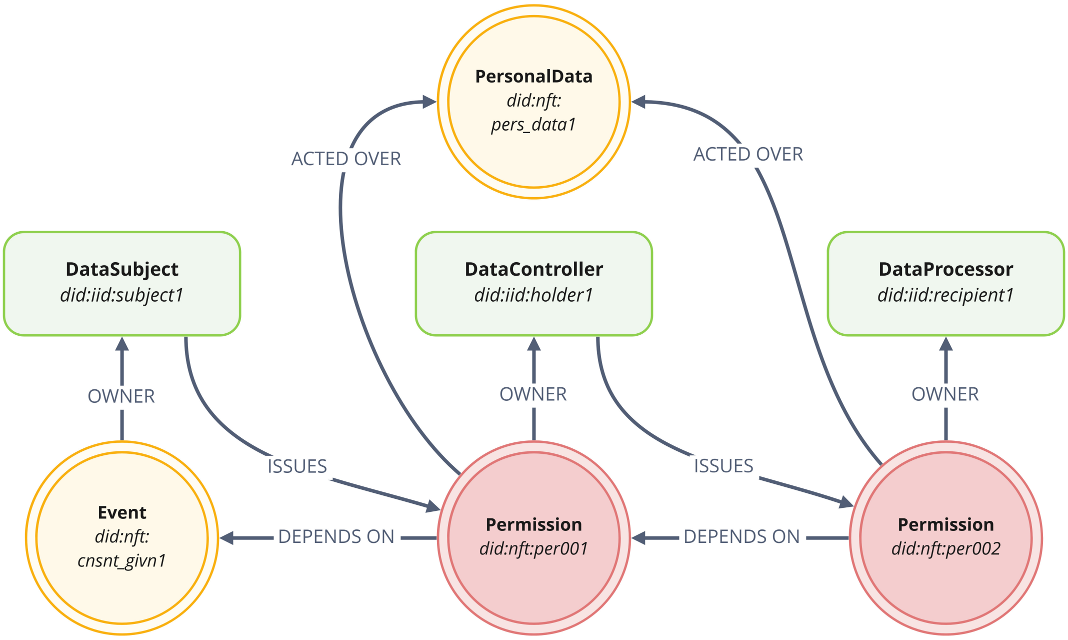
DLT or executing authorization smart contract and storing policies-related NFTs.

7.3.2.1 Personal device application and Decentralized File Storage

As we have seen previously, the personal device application is used for gathering and managing personal data and encryption keys, while the DFS is used for storing encrypted personal data. These two components can be easily extended to manage and store policies.

Information presentation The personal device application enables data subjects to select policies and make informed choices. In Sub-Section 2.1.3.1 of Chapter 2, we have discussed the stages in consent-based data processing and, in particular, the information receiving stage. In particular, the fact that users have difficulties in cognitively processing data access request information due to complexity and lack of complete vision over the context. In our PIMS, we used these findings as an architectural driver by making information transparency (and untamperability) one of the main pillars. All that happens in the authorization DLT concerning the subject can be visioned by this one, thus also all the possible personal data processing activities carried out and related policies/access requests. In the personal device application, we provide the ability to show all this information; however, we do not deal with the form of how these are presented due to possible cognitive overloading of the application users. We do not go into this detail as this topic requires other aspects that are out of the scope of this work.

Privacy Policy Objects creation Once the subject selects a series of policies displayed in natural language, we assume that the conversion from natural language to a policy file encoded in MPEG-21 MCO would be feasible due to the nature of this MPEG-21 framework specification (Kudumakis et al., 2020). The personal device application would generate an RDF encoded file such as the one in Listing 7.1. This file is given as input to a parser module that generates a set of Privacy Policy Objects (as per the ISO/IEC 21000-23 specification for Media Contractual Objects, (ISO/IEC IS 21000-23, 2022)). Each Privacy Policy Object is then stored in a DFS.



Rounded double-edge boxes are used to represent NFTs, while the rectangular ones represent information contained in those.

Figure 7.5: Graphical representation of actors, the NFTs they own, and the relations among the policies contained in those.

Decentralized File Storage and Immutable URIs Several configurations of DFS can be deployed for the use described here. For instance, the same DFS used in the PDS (see Chapter 4) can be used. However, since the authorization servers must access most of the data stored in the form of a Privacy Policy Object to perform the privacy-policy-based access control, a DFS maintained by these would be the most efficient implementation.

Based on the specifications we already described, the DFS stores objects identified by an immutable unique id. In the case of IPFS (Benet, 2014), for instance, an IPFS object shared in the network will be identified by the CID retrieved from the object hash, e.g., a document with CID equal to `QmUA3Nn...` (truncated). If any other node in the network tries to share the same document, the CID will always be the same. We exploit this feature to share policies as Privacy Policy Object to all the authorization servers (we remind them that they act as joint data controllers).

7.3.2.2 Authorization DLT

The authorization DLT maintains the same features as the one described for the PIMS and is extended with a series of smart contracts for managing NFTs and more complex access control smart contracts.

Non Fungible Tokens for linking policies and tracing data Also here, storing hash pointers on-chain and Privacy Policy Objects off-chain (i.e., in the DFS) allows us to maintain the verifiability property even if the access to the information is not public. Indeed each Privacy Policy Object is stored in documents that can be maintained private, while their hash, and thus their immutability verification, can be made public. An NFT is used to store on-chain the hash pointer to retrieve the policy from the DFS. Thus, anyone with access to the DFS object can verify if it has been altered, and, at the same time, the content is not shared publicly with anyone without access.

There can be many NFT registries, i.e., smart contracts containing a registry for enumerating NFTs. Such a registry contains a list of NFTs uniquely identifying a Privacy Policy Object, e.g., *IP Entity* object or *Deontic Expression* object. In particular, the registry maps the NFT alphanumeric id to the DLT address of the NFT owner. The NFT owner is a party, e.g., the data subject, holder, or recipient, that can decide to transfer or burn the NFT. The NFT registry owner is the party that gives the authorization to mint, i.e., create, a new NFT in that registry. When a new NFT is created, the NFT registry smart contracts bind the NFT owner to the NFT alphanumeric id and then set the NFT metadata as the hash pointer to the Privacy Policy Object. An example is shown in Figure 7.5. In that Figure, red circles are NFTs representing *Deontic Expression* objects, yellow ones represent *IP Entity* objects, and the actors are displayed in green boxes. The relations shown through arrows are encoded using the combination of MPEG-21 MCO and DPV properties in the corresponding Privacy Policy Object. The Figure shows a scenario in which a subject `did:iid:subject1` gives consent to a holder `did:iid:holder1` to share personal data by minting some NFTs in one or more owned registries. The four NFTs in Figure 7.5 represent the Privacy Policy Objects of type *Event* and *Personal Data* (i.e., sub-classes of *IP Entity*) and of type *Permission* (i.e., sub-class of *Deontic Expression*) of which encoding is shown in Listing 7.1 and 7.4. Generally, this Figure shows a piece of *Personal Data* `did:nft:pers_data1` and two *Permissions* to act over it, which on turn depend on each other and on an *Event* `did:nft:cnsnt_givn1`.

As we will see in detail in the following, the ownership of the NFTs representing these Privacy Policy Objects allows:

- the recipient to demonstrate the permission to act over the data;
- the holder to demonstrate the permission to share the data with the recipient in compliance;
- the subject to control the consent given, i.e., as the NFT can be burned to revoke consent.

We emphasize that this is one of the many possible scenarios, and different policies involving obligations or other legal bases for processing can be encoded in other scenarios. Finally, using a single structure representation on-chain, i.e., the NFT registry, enables smart contract-level interoperability. For instance, different holders can have their registry smart contract, or there can be a single registry open for more holders and recipients. At the same time, smart contracts deployed for other purposes can reference NFTs in those registries to have a direct link on-chain. In fact, the access control smart contract is extended based on this property.

From Privacy Policy Objects to Non Fungible Tokens We focus now on the process creation of NFTs, taking as reference Figure 7.6 and the permission of the scenario seen up to now. We use Listing 7.2 to show an excerpt of Listing 7.1 that encodes the central *Permission* NFT in Figure 7.5.

```
<did:nft:per001>
        a                       mvco:Permission ;
        mco-core:implements     <uri:txt001> ;
        mvco:issuedBy           <did:iid:subject1> ;
        mco-core:permitsAction  <uri:act001> ;
        mco-core:hasRequired    <uri:fac001>.
```

Example 7.2: RDF representation of a permission for data sharing policy

The process works as follows:

- On the basis of the data subject's input (action 1.1 in Figure 7.6), a set of Privacy Policy Objects is created (action 1.2).
- The set is returned (response 1.3), and then each object is stored in the DFS.

- What action 1.4 does is storing, for each Privacy Policy Object, the nested RDF *object*s first. For instance, for the *Permission* object in Listing 7.2, the RDF *object* associated with the *predicate mco-core:permitsAction*, i.e., `uri:act001`, could return (response 1.5) a hash pointer when stored in the DFS such as `QmeveuwF5wWBSgUX...AAwHUoVsx6E` (truncated). The exact process has to be executed recursively for the nested RDF *object*s within this `uri:act001` *object*. At the end of this sub-process, the *Permission* object `did:nft:per001` is stored in a form like the one in Listing 7.3. Objects that do not have an on-chain representation, such as an NFT, have their URI in this form `ipfs://QmSrPmbaUKA3Zod4...bkWfEptTBJd`.

```
<did:nft:eip155:1_erc721:0xb300a43751601bd54ffee7de35929537b28e1488_2>
        a                       mvco:Permission ;
        mco-core:implements     <ipfs://QmYNQJoKGNHT...uMa3aF6ytMpHdao> ;
        mvco:issuedBy           <did:iid:subject1> ;
        mco-core:permitsAction  <ipfs://QmeveuwF5wWBSgUX...AAwHUoVsx6E> ;
        mco-core:hasRequired    <ipfs://QmSrPmbaUKA3Zod4...bkWfEptTBJd>.
```

Example 7.3: RDF representation of a permission for data sharing policy

- The personal device application directly interacts with an NFT registry smart contract for minting (action 1.6) each *Deontic Expression* and *IP Entity* objects found in the set of Privacy Policy Objects. To maintain the immutability of information, the NFT metadata provided for minting is the hash pointer returned from the DFS. The NFT owners are indicated in the Privacy Policy Object, e.g., *mvco:actedBy*; how the DLT address is derived from an *URI* such as `did:iid:holder1` is shown in Chapter 8.

- The NFT registry smart contract returns the id of the NFT (response 1.7). Objects that do have an on-chain representation as an NFT have their URI in the following form:
  `did:nft:eip155:1_erc721:0xb300a43751601bd54ffee7de35929537b28e1488_22`,
  where the `0xb300a43751601bd54ffee7de35929537b28e1488` would stand for the NFT registry smart contract address and 22 would be the id of the NFT within the registry. This notation will be further explained in Chapter 8.

- After the last action, the NFT registry owned by the subject stores an NFT owned by the holder representing the *Permission* object for sharing data. Thus, the holder can use the received NFT id (response 1.8) to include the related *Permission* object in a new policy (action 2.1). This policy will create a new NFT in a similar way to provide access to the data recipient `did:iid:recipient1`, as we will see in the next paragraph.

- In this case, the holder device application, in order to create the new policy, it first gets the *Permission* object issued by the subject from the DFS (action 2.4) by using the NFT metadata (action 2.2).

Privacy-policy-based access control smart contract The access control smart contract presented in Chapter 6 is extended with a reference to one or many NFT registry smart contracts. The idea is that these registries, which include permissions, obligations, and prohibitions in the form of NFT, substitute the Access Control List (ACL) of data recipient addresses. Thus, from the design point of view, the access control smart contract will include one or more addresses that refer to the NFT registries smart contract, and authorization servers will use that to check the access policy. Moreover, an NFT registry will be used to "organize" encrypted personal data hash pointers in another way. Indeed, with the proposed policy model, pieces of personal data are represented as Privacy Policy Object, i.e., *IP Entity* objects. Since *IP Entity* objects have their NFT registry too, each encrypted personal data hash pointer is stored in an NFT metadata and referenced as the others, e.g., :
`did:nft:eip155:1_erc721:0xa437b30051601bd54ffee7de357b28e1488929rt_1`.

What changes the most in this part of the privacy-policy-based access control is the operational point of view, as the authorization servers execute a reasoning process over the policies encoded using the Privacy Policy Objects. The output of such a reasoning process is a boolean response as well as before, thus, the capsule distribution operation remains the same. Starting from the example Listing 7.1 again, a data recipient would need to receive permission from the data holder, i.e., `did:iid:holder1`, in line with the subject consent. The one in Listing 7.4, for instance.

```
<uri:text2>
        a                   mco-core:TextualClause ;
        mco-core:text       "Consult sensitive personal data for
                                 Targeted Advertising in Social Media" .
```

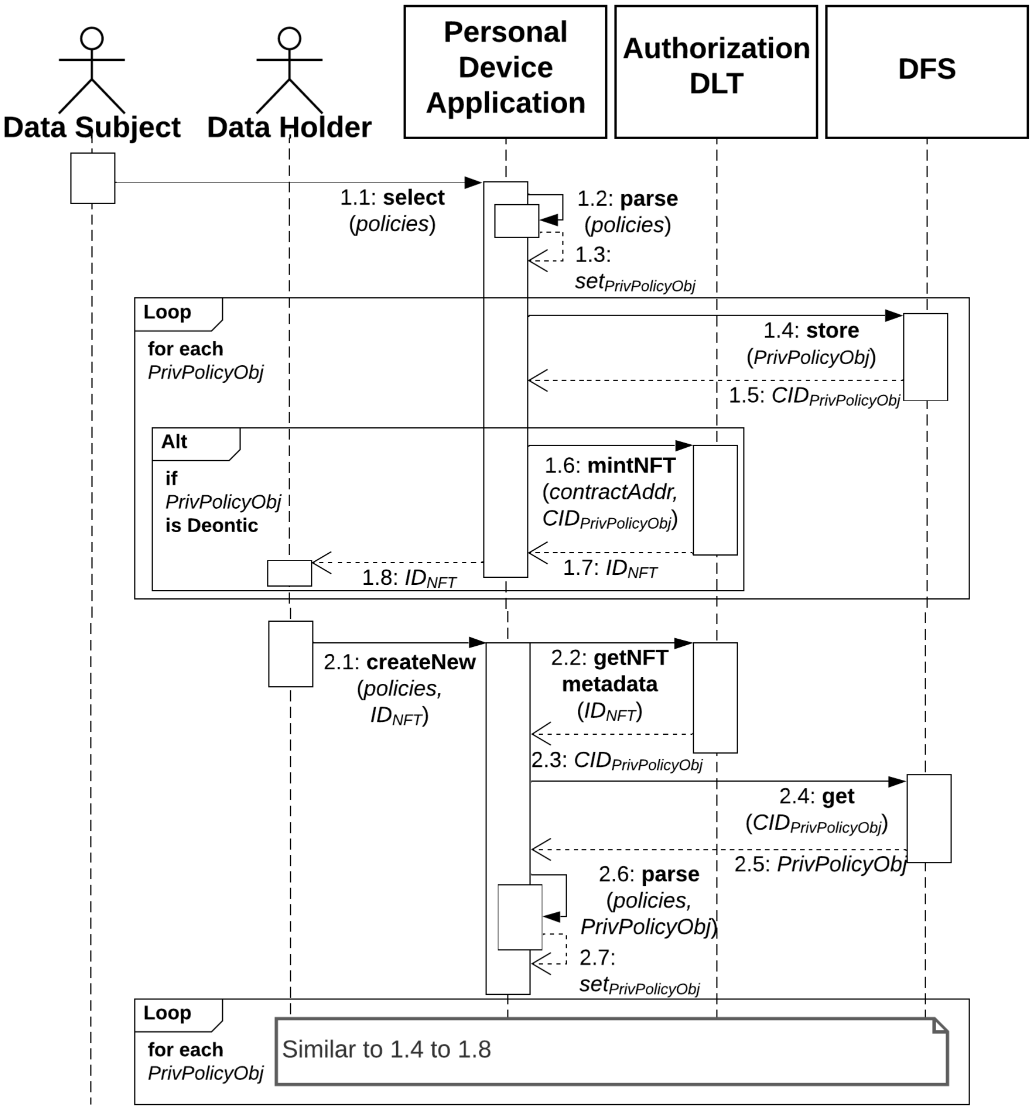

Personal Device Application
Authorization DLT
DFS
Data Subject
Data Holder
1.1: select (policies)
1.2: parse (policies)
1.3: set$_{PrivPolicyObj}$
Loop
for each PrivPolicyObj
1.4: store (PrivPolicyObj)
1.5: CID$_{PrivPolicyObj}$
Alt
if PrivPolicyObj is Deontic
1.6: mintNFT (contractAddr, CID$_{PrivPolicyObj}$)
1.7: ID$_{NFT}$
1.8: ID$_{NFT}$
2.1: createNew (policies, ID$_{NFT}$)
2.2: getNFT metadata (ID$_{NFT}$)
2.3: CID$_{PrivPolicyObj}$
2.4: get (CID$_{PrivPolicyObj}$)
2.5: PrivPolicyObj
2.6: parse (policies, PrivPolicyObj)
2.7: set$_{PrivPolicyObj}$
Loop
for each PrivPolicyObj
Similar to 1.4 to 1.8

Figure 7.6: UML sequence diagram describing the process of creating a policy.

```

<did:iid:recipient1>
        a           dpv:DataProcessor ;
        rdfs:label  "Data Recipient" .

<did:nft:per002>
        a                       mvco:Permission ;
        mco-core:implements     <uri:text2> ;
        mvco:issuedBy           <did:iid:holder1> ;
        mco-core:permitsAction  <uri:act002> ;
        mco-core:hasRequired    <uri:fac002> .

<uri:act002>
        a               dpv:Consult ;
        mvco:actedBy    <did:iid:recipient1> ;
        mvco:actedOver  <did:nft:eip155:1_erc721:0xa437b300...1488929rt_1> .

<uri:fac002>
        a               mvco:FactIntersection .
        mvco:hasFact    <uri:aef002>,
                            <uri:pdh001> ;

<uri:aef002>
        a               mvco:ActionEventFact .

<uri:act001>
        a                   dpv:Share ;
        mco-core:makesTrue  <uri:aef002> .

<did:nft:per001>
        a                       mvco:Permission ;
        mco-core:permitsAction  <uri:act001> ;

<uri:pdh001>
        a                   dpv:PersonalDataHandling ;
        dpv:hasPurpose      <uri:repo001>,
                                <uri:repo002> ;
        dpv:hasProcessing   <uri:repo003> .
```

Example 7.4: RDF representation of permission for data consult processing activity.

The *Permission* object `did:nft:per002` is represented through an NFT of which the owner is the recipient, i.e., `did:iid:recipient1`. In order to access data, the recipient provides the NFT's id, i.e., `did:nft:per002`, to the authorization servers as a data access request. This request passes through only if this NFT "fits in the privacy policy puzzle". Indeed, before releasing the capsule to the recipient, each authorization server reconstructs the full privacy policy based on the URIs found in the data access request and checks for logical errors. We use the diagram in Figure 7.7 to render the process described in the following clearer. In the Listing 7.4 example, each server:

- starts from the permission `did:nft:per002`;

- checks the required facts, i.e., the intersection between fact `uri:aef002` and fact `uri:pdh001`;

- the `uri:aef002` fact is associated with the *Share* action in permission `did:nft:per001`; then this *Permission* object is obtained from the DFS, and its information merged with the one already got (Listing 7.1 and 7.4 merged).

- the `uri:pdh001` fact is associated with a series of purposes and processing objects that can be considered common to many other policies and thus are included in a terminology repository in the DFS.

In this case, the logical breaking points could be two:

1. when the *dpv:PersonalDataHandling* `uri:pdh001` object does not have the same purposes included in the subject's *Consent* `uri:con001` object, i.e., *dpv:SocialMediaMarketing* and *dpv:TargetedAdvertising*;

2. when the *dpv:Share* `uri:act001` object does not make true *ActionEventFact* `uri:fact007` because it does not exist anymore; this means the subject could have withdrawn the consent through an *Event* object.

This example can be easily generalized to a process that authorization servers execute to reconstruct privacy policies and then try to validate data access requests. Moreover, thanks to a DLT and the NFT representation, all the access control operations can be traced up to fine-grained detail.

The implementation of the above presented components can be found in (Zichichi, 2021c,d, 2022a,e,f).

7.4 Use cases

In this Section, we describe two use cases to validate, within the context of our proposed PIMS, the Privacy-policy-based access control layer against the ACL-based access control. Two use cases will demonstrate the richer expressiveness of the proposed approach in opposition to the limited ACL approach, in which a list of addresses regulates access.

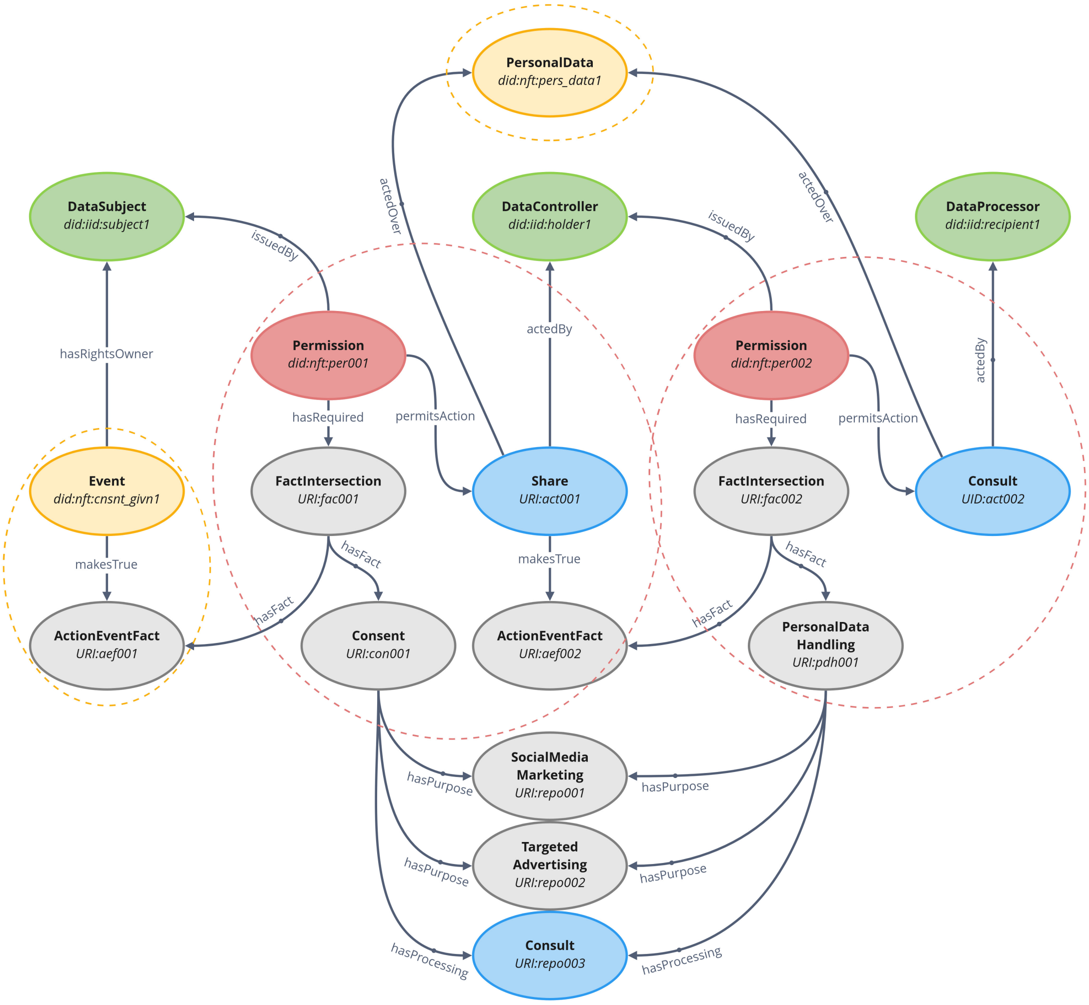
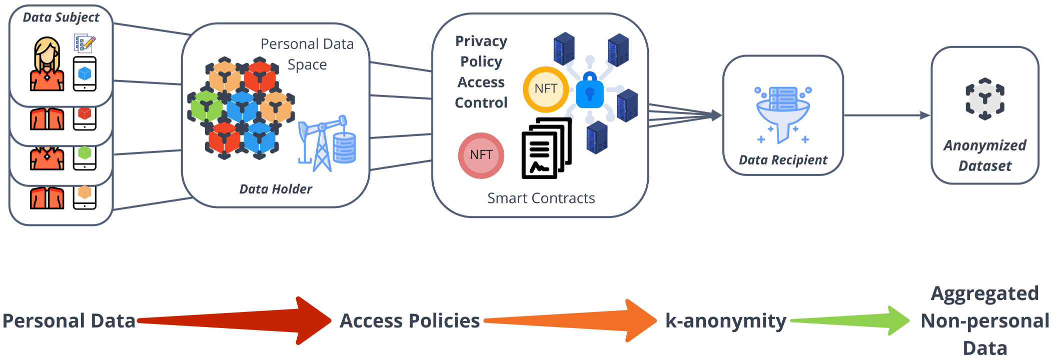


Figure 7.7: Graphical representation of a privacy policy and a data access request relations expressed in RDF.



Bubbles represent RDF subjects/objects, while arrows represent RDF predicates.

Figure 7.8: Graphical representation of a citizen-generated data use case involving a Smart Data Processing Agreement.

7.4.1 "Smart" Data Processing Agreement

A use case can be instantiated in which, based on the approach used in the previous Section for a subject providing consent to access some personal data, a "Smart" Data Processing Agreement (DPA) is contracted between the data holder and the recipient. We describe a citizen-generated data scenario for facilitating the use of privately held data for the public interest. This is in line with the vision of the European Union's strategy on data sharing for public interest (High-Level Expert Group on Business-to-Government Data Sharing, 2021). More specifically, in our use case, a data holder gives access to a data recipient that uses data transformed into a non-personal form to create value and make better decisions in the public domain. In most cases, citizen-generated data should be made orthogonal to the application of data protection laws and regulations, e.g., GDPR (Corcho et al., 2022). Therefore, citizen-generated data should not contain personal data, or personal data shall be appropriately anonymized or aggregated. We imagine a concrete scenario of citizen-generated hiking trails or pedestrian travel routes produced using GPS-enabled smartphones using an application such as Komoot or AllTrails (Campbell et al., 2019). Citizen's data, thus, consist of travel traces, i.e., a set of latitude and longitude points associated with a timestamp.

With this in mind, we describe the use case with the help of Figure 7.8. At the highest level, the flow of data is as follows: (i) data subjects, i.e., citizens, generate

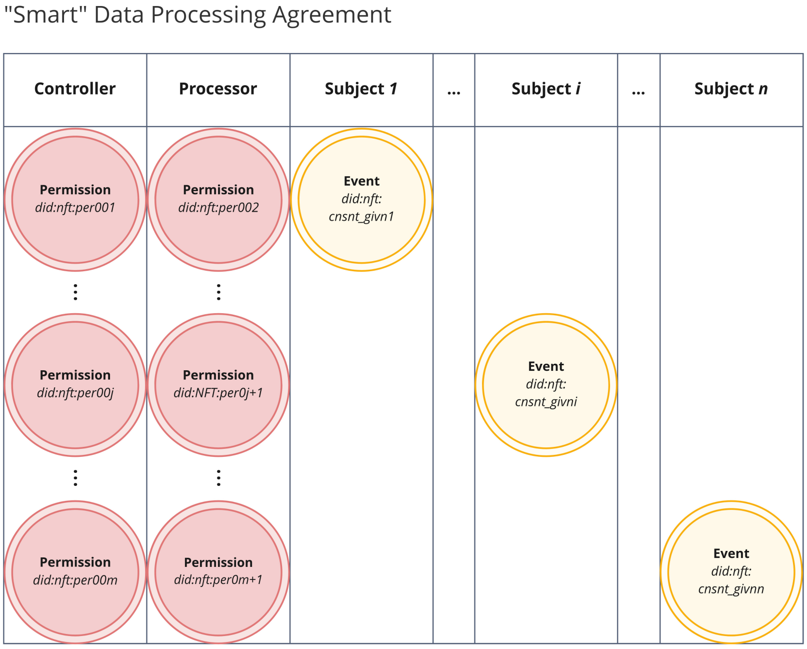


Figure 7.9: Table representing NFTs associated with a Smart DPA linked to several subjects' privacy policies.

personal data through their hiking app and interact with their PDS through their device application; (ii) a data holder, i.e., the manufacturer of the hiking app, stores and maintain personal data in the PDS associated to each subject; (iii) a data recipient undertakes the task of aggregating a specific kind of data and accesses PDS through a Smart DPA based on privacy policies; (iii) the recipient uses algorithms such as the k-anonymity (Samarati and Sweeney, 1998) to render the input personal data anonymous; (iv) the citizen-generated anonymized aggregated dataset is published as Open Data containing non-personal data. The main idea is to have subjects select their privacy policies, as seen in the previous Section, and then have the data holder aggregate these policies for signing a Smart DPT with the recipient.

7.4.1.1 Use case steps

- The data holder maintains personal data in a PDS implemented within the PIMS. The decentralized index used for each piece of data is the id of the NFT representing the related *IP Entity* Privacy Policy Object in the authorization DLT, e.g., `did:nft:eip155:1_erc721:0xa437b30051601bd54ffee7de357b28e1488929rt`

_1.

- Each data subject owns a privacy-policy-based access control smart contract that points to NFT registries that include NFTs representing those *IP Entity* objects, but also *Deontic Expression* objects. For this use case, we imagine that each subject permits to share the hiking travel trace data with the data holder through a *Permission* object similar to the one in Listing 1 of Section 7.3.

- The data recipient sends a request to the holder with a *PersonalDataHandling* object indicating the type of processing (Listing 7.5).

```
<uri:text3>
        a                  mco-core:TextualClause ;
        mco-core:text      "Combine and anonymize personal data" .

<did:iid:recipient1>
        a           dpv:DataProcessor ;
        rdfs:label  "Data Recipient" .

<uri:pdh002>
        a                       dpv:PersonalDataHandling ;
        dpv:hasProcessing          <uri:repo004>,
                                    <uri:repo005> .
<uri:repo004>
        a               dpv:Combine .

<uri:repo005>
        a               dpv:Anonymise .
```

Example 7.5: RDF representation of a personal data handling based on combine and anonymize processing activity.

- The data holder creates a Smart DPA that aggregates all the NFTs representing *Permission* objects issued by the subjects, where the *Consent* object include the processing activities *Combine* and *Anonymise*. For each subject's permission to share, the holder mints a new NFT representing a *Permission* object for the recipient (see Figure 7.9). It is done to enable subjects to burn the NFT representing the consent given *Event* in order to opt out from the processing activity.

- The Smart DPA includes all the references to minted *Personal Data* and *Permission* NFTs but can also be encoded with other clauses. For instance, the holder can obligate the recipient to make data available after the processing activity, i.e., make it open data. This is done through an *Obligation Deontic - Expression* that obligates the *Make Available - Action*. The complete encoding of the Smart DPA in RDF is shown in Listing 7.6.

- If the recipient manages to access the data of at least k data holders, this one should be able to perform the data aggregation producing a dataset that presents properties of k-anonymity, i.e., where each release of data must be indistinguishably related to no less than a certain number (k) of individuals in the population (Samarati and Sweeney, 1998). This dataset can then be released as open data containing non-personal data.

```
<did:iid:holder1>
    a           dpv:DataController ;
    rdfs:label  "Data Holder" .

<did:iid:recipient1>
    a           dpv:DataProcessor ;
    rdfs:label  "Data Recipient" .

<uri:smartdpa1>
    a                  mco-core:Contract ;
    rdfs:label         "Smart Data Processing Agreement for combining and
                            anonymizing location data" ;
    mco-core:hasParty  <did:iid:holder1> , <did:iid:recipient1> .

<uri:text4>
    a                  mco-core:TextualClause ;
    mco-core:text      "Obligate the recipient to make data available after
                            the processing activity" .

<did:nft:obl001>
    a                           mco-core:Obligation ;
    mco-core:implements         <uri:text4> ;
    mco-core:issuedIn           <uri:smartdpa1> ;
    mvco:issuedBy               <did:iid:holder1> ;
    mco-core:obligatesAction    <uri:act003> .

<uri:act003>
    a                   dpv:MakeAvailable ;
    mvco:actedBy        <did:iid:recipient1> ;
    mvco:actedOver      <did:nft:eip155:1_erc721:0xa43...29rt_664> .

<did:nft:eip155:1_erc721:0xa43...29rt_664>
    a                       dpv:AnonymisedData ;
    mvco:hasRightsOwner     <did:iid:recipient1> ;
    mvco:resultedFrom       <uri:act301> .

<uri:act301>
    a               dpv:Anonymise ;
    mvco:actedBy    <did:iid:recipient1> .


<did:nft:cnsnt_givn1>
    a                       mco-core:Event ;
    mvco:hasRightsOwner     <did:iid:subject1> ;
    mco-core:makesTrue      <uri:aef001> ;
    rdfs:label              "Subject's consent given event" .

<uri:aef001>
    a               mco-core:ActionEventFact .

<did:nft:per001>
    a                       mvco:Permission ;
    mvco:issuedBy           <did:iid:subject1> ;
    mco-core:permitsAction  <uri:act001> ;
    mco-core:hasRequired    <uri:fac001>.

<uri:act001>
    a                   dpv:Share ;
    mvco:actedBy        <did:iid:holder1> ;
    mvco:actedOver      <did:nft:eip155:1_erc721:0xa43...29rt_1> ;
    mco-core:makesTrue  <uri:aef002> .

<uri:fac001>
    a               mco-core:FactIntersection ;
    mvco:hasFact    <uri:aef001>, <uri:con001> .

<uri:aef002>
    a              mco-core:ActionEventFact .

<uri:con001>
    a                       dpv:Consent ;
    dpv:hasDataSubject      <did:iid:subject1> ;
    dpv:hasDataController   <did:iid:holder1> ;
    dpv:hasProcessing       <uri:repo004>, <uri:repo005> .

<uri:text3>
    a                  mco-core:TextualClause ;
    mco-core:text      "Combine and anonymize personal data" .

<did:nft:per002>
    a                       mvco:Permission ;
    mco-core:implements     <uri:text3> ;
    mco-core:issuedIn       <uri:smartdpa1> ;
    mvco:issuedBy           <did:iid:holder1> ;
    mco-core:permitsAction  <uri:act002> ;
    mco-core:hasRequired    <uri:fac002> .

<uri:act002>
    a               dpv:Combine ;
    mvco:actedBy    <did:iid:recipient1> ;
    mvco:actedOver  <did:nft:eip155:1_erc721:0xa43...29rt_1> .

<uri:fac002>
    a               mco-core:FactIntersection ;
    mvco:hasFact    <uri:aef002>, <uri:pdh002> .

<did:nft:cnsnt_givn2>
    a                       mco-core:Event ;
    mvco:hasRightsOwner     <did:iid:subject2> ;
    mco-core:makesTrue      <uri:aef201> ;
    rdfs:label              "Subject's consent given event" .

<uri:aef201>
    a               mco-core:ActionEventFact .

<did:nft:per003>
    a                       mvco:Permission ;
    mvco:issuedBy           <did:iid:subject2> ;
    mco-core:permitsAction  <uri:act201> ;
    mco-core:hasRequired    <uri:fac201>.

<uri:act201>
    a                   dpv:Share ;
    mvco:actedBy        <did:iid:holder1> ;
    mvco:actedOver      <did:nft:eip155:1_erc721:0xa43...29rt_2> ;
    mco-core:makesTrue  <uri:aef202> .

<uri:fac201>
    a               mco-core:FactIntersection ;
    mvco:hasFact    <uri:aef201>, <uri:con201> .

<uri:aef202>
    a               mco-core:ActionEventFact .

<uri:con201>
    a                       dpv:Consent ;
    dpv:hasDataSubject      <did:iid:subject2> ;
    dpv:hasDataController   <did:iid:holder1> ;
    dpv:hasProcessing       <uri:repo004>,  <uri:repo005> .

<did:nft:per004>
```

```
    a                       mvco:Permission ;
    mco-core:implements     <uri:text3> ;
    mco-core:issuedIn       <uri:smartdpa1> ;
    mvco:issuedBy           <did:iid:holder1> ;
    mco-core:permitsAction  <uri:act202> ;
    mco-core:hasRequired    <uri:fac202> .

<uri:act202>
    a               dpv:Combine ;
    mvco:actedBy    <did:iid:recipient1> ;
    mvco:actedOver  <did:nft:eip155:1_erc721:0xa43...29rt_2> .

<uri:fac202>
    a               mco-core:FactIntersection ;
    mvco:hasFact    <uri:aef202>, <uri:pdh002> .

<uri:pdh002>
    a                   dpv:PersonalDataHandling ;
    dpv:hasProcessing      <uri:repo004>,
                            <uri:repo005> .
<uri:repo004>
    a               dpv:Combine .

<uri:repo005>
    a               dpv:Anonymise .
```

Example 7.6: RDF representation of a Smart DPA for combining and anonymizing citizens' crowdsourced personal data.

Finally, the Smart DPA encoded in RDF undertakes the same conversion process into a smart contract as in the ISO/IEC 21000-23 Smart Contract for Media (ISO/IEC IS 21000-23, 2022). The result would be a smart contract in the authorization DLT that points to all the NFTs representing the Privacy Policy Objects and possibly implementing other functionalities, e.g., direct payment from the recipient to the holder.

7.4.2 Not only consent: personal data access based on vital interest legal basis

A use case can be instantiated in which, instead of using the consent approach used in the previous Section for providing consent to access personal data, another legal basis is involved and encoded thanks to Privacy Policies Objects. In this use case, we refer to the "vital interest" legal basis, i.e., the processing activity that could be necessary to save someone's life. In this case, everything revolves around an *Event* object. The event represents the need for the processing activity based on a vital interest (see Listing 7.7). This event can be generated by the data holder as a result of a request from the data recipient when it can rely on the latter being trusted in declaring a vital interest. Alternatively, it may be declared by a third party, such as an oracle (Beniiche, 2020), reporting a real-world event in the authorization DLT.

```
<did:nft:vital_interest1>
    a                       mco-core:Event ;
    mvco:hasRightsOwner     <did:iid:subject1> ;
    mco-core:makesTrue      <uri:aef051> ;
    rdfs:label              "Real world vital interest event" .

<uri:aef051>
    a               mco-core:ActionEventFact .

<did:nft:per005>
    a                       mvco:Permission ;
    mvco:issuedBy           <did:iid:holder1> ;
    mco-core:permitsAction  <uri:act501> ;
    mco-core:hasRequired    <uri:fac501> .

<uri:act501>
        a               dpv:Consult ;
        mvco:actedBy    <did:iid:recipient1> ;
        mvco:actedOver  <did:nft:eip155:1_erc721:0xa43...29rt_50> .

<uri:act001>
    a                   dpv:Share ;
    mvco:actedBy        <did:iid:holder1> ;
    mvco:actedOver      <did:nft:eip155:1_erc721:0xa43...29rt_1> ;
    mco-core:makesTrue  <uri:aef002> .

<uri:fac501>
    a               mco-core:FactIntersection ;
    mvco:hasFact    <uri:aef051>, <uri:vin001> .

<uri:vin001>
        a                   dpv:VitalInterest .
```

Example 7.7: RDF representation of a Smart DPA for combining and anonymizing citizens' crowdsourced personal data.

In any case, after the NFT representing the *Event* object has been minted, the subject can burn it if the vital interest information is false.

7.4.3 Privacy-policy-based and ACL-based access control

We use Table 7.3 as a reference for comparing the approach of the privacy-policy-based access control layer with the approach presented in Chapter 6.

Both approaches are based on smart contracts executed on the authorization DLT in compliance with the GDPR, trace the sharing of personal data, and allow to declare the permission to access such data. On the other hand, as seen in this Section, the privacy-policy-based approach enables the declaration of obligations and prohibitions regarding the processing activity or personal data handling. Sub-Section 7.4.1 shows an implementation of a Smart Data processing Agreement that cannot be implemented using the ACL approach. The same goes for supporting all legal bases, as shown in an example in Sub-Section 7.4.2.

Table 7.3: Validation of the privacy-policy-based access control vs. ACL-based.

	Authorization DLT Access Control	
	Privacy policy based	**ACL based**
Smart contract use	✓	✓
Data sharing tracing	✓	✓
Enable to declare permissions	✓	✓
Enable to declare obligations	✓	×
Enable to declare prohibitions	✓	×
GDPR-compliant	✓	✓
Enable Smart DPAs	✓	×
Supports all GDPR legal bases	✓	× (only consent)

7.5 Conclusions

In this Chapter, we presented the model for privacy-policy-based access control based on the use of several Semantic Web standards and their integration with the PIMS shown in previous chapters. The rationale was to design and implement a tool to effectively exercise GDPR legal bases for accessing personal data, where the data subject is at the center of the decision-making.

A privacy-policy-based access control model complements the list-based access control presented in Chapter 7. The ISO/IEC 21000 MPEG-21 framework and Smart Contract for Media International Standards are used to specify policies over assets. This is because the Smart Contract for Media specifies how to link rights expressions directly to DLT objects. Then, we employed the W3C Data Privacy Vocabulary specification in order to represent data privacy and GDPR-related concepts. The implementation of the proposed access control model relies on smart contracts and Non Fungible Token (NFT) representation, to uniquely trace data and policies.

The implementation of the "Smart" Data Processing Agreement and different legal bases use cases led us to assess the value of the proposed access control model for an authorization based on a specific set of privacy policies. This privacy-policy-based access control layer aims to guarantee a series of features in favor of a transparent personal data access process. Indeed it allows us to enforce policies and trace the operations when they are used.

Finally, some problems remain open as they are outside the scope of this paper or have been omitted for the sake of brevity of the entire work so as to make it more readable. The first problem can be considered sociotechnical in nature, as an entity that adopts privacy-policy access control may run into legal problems arising from bugs or malfunctions, e.g., imagine a bug in the immutable source code of smart contract access control. This issue is raised by many in the general case of smart contract-based services because the source code may have vulnerabilities that are not easily fixed. In addition, several limitations can be identified in the proposed approach, such as the compulsion to refer only to the GDPR, but also to the other legal bases of data processing within the GDPR. In either case, the most favorable approach would be to structure the software in a modular way, so that GDPR-based policy languages and other languages are used interchangeably.

Chapter 8

Self-Sovereign Identity Model

The software produced during the development of this chapter is stored in the following repositories:

- M. Zichichi (2021). Intelligible Decentralized Identity implementation. `github.com/miker83z/desp3d-intelligible-identity`

- M. Zichichi (2021). Intelligible Verifiable Certificate implementation. `github.com/miker83z/desp3d-intelligible-certificate`

- M. Zichichi (2022). Intelligible packages based on DID and VC. DOI: `10.5281/zenodo.7132777`

- M. Zichichi (2022). Client tools to interact with Intelligible suite and Privacy Policy software. `github.com/miker83z/desp3d-client-tools`

This Chapter describes the last layer of our user-centered model, i.e., the Self-Sovereign Identity (SSI) layer. Such a layer is transversal to all the others as the final user needs to be identified during the execution of the decentralized systems. However, this layer also needs all the other inner layers, as the final user's identity will also be made up of those personal data which he or she may control more directly. The SSI layer closes the circle of the user-centered PIMS and creates a port to let any ICTs service interact with the *onlife* identity of an individual Floridi (2014).

The surge of DLTs and decentralized technologies have also brought regulators to reevaluate the concept of a "trusted" transaction of online credentials. In particular, the new proposal of the EU Commission (European Commission, 2021) introduces new modifications to the regulation on Electronic Identification, Authentication, and Trust Services (eIDAS) (European Parliament, 2014). The eIDAS was enacted into law in 2014 for the assurances of juridical traffic performed electronically in the EU internal market. Such a juridical act is an expression of will, intended to have legal consequences, and it is performed electronically by natural and legal persons Preukschat and Reed (2021). These include digital signatures, i.e., a cryptographical scheme verifying the authenticity of digital messages or documents. Since this scheme uses asymmetric encryption, the public key needs to be associated with a digital certificate, regulated as a specific trust service by the eIDAS Regulation. PKIs are formed by certification authorities (CA) that issue digital certificates binding a public key with an identified entity. The EU Commission's new proposal introduces modifications to the eIDAS to manage the qualified identity management based on DLT, with regulations for new trust services linked to the qualified electronic ledgers. In such form, legislation at the EU level would support the use of specific Decentralized Identifiers (DIDs) Longley et al. (2021) and a specific type of Verifiable Credentials Longley et al. (2019) as a qualified certificate for both natural and legal persons. These two standards, i.e., the DID and VC, are currently the most used implementation of Self-Sovereign Identity. The DID is a type of identifier entirely under the control of the identity subject, independent from any centralized registry, identity provider, or certificate authority. It relates an entity to means for trustable interactions with that entity, i.e., VCs. A VC is a tamper-evident credential with authorship that can be cryptographically verified.

We present in this Chapter, as an implementation of the SSI layer, a new identity model called Intelligible Decentralized Identity (Intelligible DID), which we implemented as a set of technological components that are deployed in decentralized environments for the purpose of providing, requesting and obtaining qualified data in order to negotiate and/or execute electronic transactions, extending the DID and VC implementations Preukschat and Reed (2021). Our proposal adds the possibility of bringing with it the relevant operational and legal context of this identity and easily tracing the processes that involve it. We argue that a fundamental challenge exists in balancing the powerful capabilities of everyday systems with the development of technologies that can let people be empowered. For instance, the lack of transparency in the decisions taken by intelligent systems can negatively impact user acceptance and satisfaction with system results Eiband et al. (2018). Governments have begun establishing rules of conduct by complex systems, such as emphasizing the importance of a Right to Explanation in the GDPR AI HLEG (2019); ICO (2019). The general idea is that people should be able to understand how technology can affect them, trust it, and feel in control Abdul et al. (2018). In this work, we exploit DLTs to build models that focus on the concepts of user empowerment and system intelligibility, i.e., the ability to represent to their users what they know, how they know it, and what they are doing about it Bellotti and Edwards (2001).

The Intelligible DID model is a combination of (i) asymmetric cryptography key pairs, i.e., a public key and a private key; (ii) a Non Fungible Token (NFT) stored on the ledger of a public smart-contract-enabled DLT, such as the blockchain provided at EU level by the European Blockchain Services Infrastructure (EBSI) (European Blockchain Partnership EBP, 2022); (iii) an intelligible identity document stored outside of the ledger, i.e., off-chain. The intelligible identity document is the heart of the Intelligible DID, as it references all the other documents involved in the legal and operational context. Through its use, a built-in linkability property allows interoperability between data structures and possible decentralized application services based on it. This document delivers a specification to mark the relevant legal prose of identity assertions and digital resources, defining the bridge from this to the corresponding operational code. Intelligibility is conveyed by linking (i) the resources that make up the document or define their legal contexts, (ii) the agents that are involved in the document life cycle, and (iii) the digital resources that describe how to perform operations with the identities.

The original contributions and novelties of this Chapter are described in the following:

- First, the main contribution of this Chapter consists of the design of an SSI model for verifying the authenticity of some claims in digital interactions, where information about an entity, be it a physical or digital object or an identity, has to be shared with third parties, i.e., the Intelligible Decentralized Identity and Verifiable Certificate.

- Second, the implementation of these two models into a PIMS layer that relies on its functioning on smart contracts and standard technologies is described. The stack of technologies includes: (i) a set of smart contracts implemented to govern the digital interactions for the exchange of claims; (ii) an application of the Akoma Ntoso (AKN) OASIS standard Palmirani and Vitali (2011) for structuring documents and linking entities; (iii) a document storage layer based on a DFS.
- Third, we provide a detailed description of the use of our SSI layer for the certification of Open Data to improve the reusability and shareability of such data while providing the possibility to track the legal basis, consistency, and scope.

The remainder of this Chapter is organized as follows. Section 8.1 presents the background concepts behind the proposed systems and related works. In Section 8.3.2, we introduce the Intelligible Decentralized Identity and Certificate models. In Section 8.4, we provide a possible implementation of an Intelligible System involving both. In Section 8.4.2, we evaluate the use of such a system, while conclusions are presented in Section 8.5.

8.1 Background and Related Works

In this Section, we describe the models, technologies, and related work that led to the concept of SSI and that we embrace in our SSI layer.

8.1.1 Intelligible Systems

The problem of intelligibility of systems is related to a more widely investigated problem, that is, the explainability for complex systems, including eXplainable Artificial Intelligence (XAI) and explanatorY Artificial Intelligence (YAI) Sovrano and Vitali (2021). In particular, in literature, we find that intelligibility and fidelity are both necessary to reach explainability, where the former captures how understandable an explanation is for humans. The latter expresses how accurately an explanation describes model behavior Markus et al. (2020). From the point of view of an individual, i.e., the human interacting with a system, rendering the latter intelligible means that the individual is provided with the means to be made fully aware of the provided content and behavior. We also mainly focus on the operational context and the legal framework in which the system operates, which are necessary for anyone to understand what he or she is operating with entirely but also to allow any other external system to interact with it and deepen the content verification. From the point of view of a machine processing a digital document, for instance, the readability of the information would ensure that the processing environment is provided with an operational, legal context.

In the case of smart contracts, this concept was already presented in the Intelligible Contract proposal Cervone et al. (2020). The Intelligible Contract is a unique collection of linked legal resources describing a contract, its legal prose, its legal context, and information on which parts can be automatically processed and how to do it. Not only do these contracts enable the creation of a bridge between the human-readable legal prose of contracts and the operational code that automatically executes such prose, but they also enable the linking of (i) contract resources and their legal contexts; (ii) parties involved in the life-cycle of the contract; (iii) digital resources that describe how to execute the operational code and that report its execution.

8.1.2 Verified information transaction in light of DLTs

The surge of DLTs has led to a renewed consideration of some of the problems that have accompanied the Internet and computer systems since their inception, such as the verifiability of shared information. For instance, if we limit the scope to the digital information that regards the process of certification, i.e., a formal attestation or confirmation of specific characteristics of an object, person, or organization, it is essential for the digital document containing the certificate information to remain untampered and unambiguously identifiable Preukschat and Reed (2021). When intelligible certificate documents are translated from a physical to a digital form, usually, a single entity is needed to act as guarantor of the non-counterfeiting of the digital document, e.g., through a digital signature. This paradigm implies that the verifier of a digital certificate puts its trust in the guarantor or in another entity that, in turn, trusts the guarantor Bradbury (2012). The DLTs' decentralization and un-tamperability reduce the likelihood of certificate forgery and, in addition, make the certification and automatic licensing processes open and transparent (when needed). Cheng et al. Cheng et al. (2018) present, for example, a DLT-based system where companies or organizations can request information on any certificate directly from the ledger. While, Gräther et al. Gräther et al. (2018) present a platform implementation for education that, through the use of a DLT and smart contracts, manages certification authorities and certificates, as well as services for certifiers, learners and third parties.

8.1.3 Self-Sovereign Identity

The Internet was built without a way to know to whom and to whom people are connecting. Since its inception, Internet service providers have had to resort to the fallback of making the Internet based on a “mosaic of disposable identities” Cameron (2005). Online identity models have evolved in the years through a few major phases Christopher Allen (2016): (i) centralized identity, locked by a single authority; (ii) federated identity, locked by multiple federated authorities; (iii) user-centric identity, with the user in the middle of the identity process but still locked into a single authority central control. A new fourth phase is emerging thanks to the DLTs’ paradigms, referred to as Self-Sovereign Identity (SSI) Wang and De Filippi (2020).

SSI consists of the complete control of individuals’ digital identities and their data through decentralization Kondova and Erbguth (2020). This is a solution that enables any subject to share information with third parties by proving to those the ownership of certain attestations or attributes that are self-asserted or issued by trusted[6] entity. Any verifier is provided with the means to verify the validity of claims independently of any centralized registry, identity provider, or certification authority, given the employment of cryptographic methods. SSI has been generically implemented as a set of technological components deployed in decentralized environments to provide, request, and obtain qualified data to negotiate and/or execute electronic transactions Preukschat and Reed (2021). Scholars and companies have already produced some contributions leveraging public and/or private DLTs for SSI. Sovrin Sovrin Foundation (2020) is an open-source identity network built on a layered architecture involving a public permissioned DLT that only stores identity transactions without personal data and where only trusted institutions, i.e., banks, universities, and governments, can be DLT nodes. uPort Lundkvist et al. (2017) is based on Ethereum’s public permissionless DLT and provides a platform for user-centric identity and communication, consisting of a mobile app, smart contracts, and a set of open protocols for message flow.

[6] trust here refers to the subjective viewpoint of an individual who has a measure of confidence in another individual or entity

8.2 DID and VC based Self Sovereign Identity ecosystem

In this Section, we will introduce the DID and VC elements since our proposed SSI layer is based on them. In particular, we will define the actions and roles within the SSI ecosystem that facilitate the transaction of VCs.

8.2.1 SSI Actions

So-called "SSI actions" can be defined as the core actions affecting the exchange and management of relevant data and the qualifications and other assurances provided and/or required. These actions aim to provide, request and obtain qualified data to negotiate and/or execute electronic transactions. In the following, we describe the general model:

1. **Request the data** - The first participant request qualified data for (i) making a commitment or acceptance decision for a transaction (ii) executing part of a transaction.

2. **Provide the response** - The other participant(s) provide the data as a response to the request

3. **Verify** - the data obtained from the response is *verified* by the requestor

4. **Validate** - the data obtained from the verified response is then *validated*

5. **Continue/terminate transaction** - the requestor state any intermediate and/or final result of the transaction to the other participant(s)

8.2.2 SSI Roles

This Sub-Section describes the roles and relationships between the SSI ecosystem actors. Figure 8.1 displays that information graphically. The subject is again the entity about which claims are made but can also not participate in the processes.

- **Issuer** - A role an entity performs by asserting claims about one or more subjects, i.e., by issuing VCs. The issued credentials are composed of: (i) a set of (related) statements/claims; (ii) metadata, e.g., date of issuing; (iii) a digital signature, by

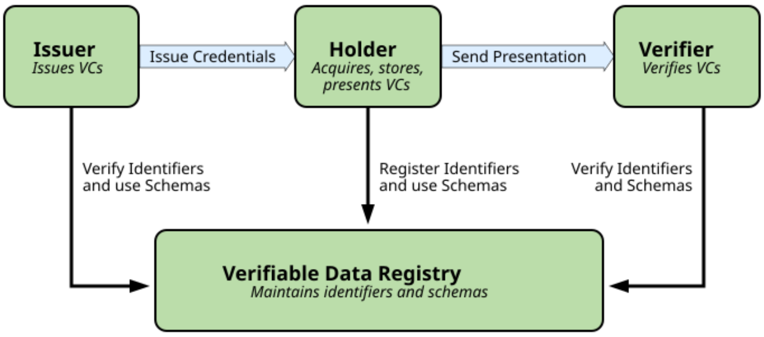

Issuer
Issues VCs
Issue Credentials
Holder
Acquires, stores, presents VCs
Send Presentation
Verifier
Verifies VCs
Verify Identifiers and use Schemas
Register Identifiers and use Schemas
Verify Identifiers and Schemas
Verifiable Data Registry
Maintains identifiers and schemas

Figure 8.1: SSI roles in the Verifiable Credentials specification

which third parties can prove its provenance and integrity. The issuer can also revocate and (un)suspend VC already issued.

- **Holder** - stores VCs securely under its own control through a wallet. This component stores self-signed credentials, or VCs, obtained from third-party issuers. It also stores private keys that can be used to sign or seal data. The holder handles credentials requests that it receives from a verifier, looking for the requested data in the wallet, i.e., they provide the response SSI action. If the wallet does not store the requested data, the holder may negotiate a transaction with an issuer to obtain the needed VCs.

- **Verifier** - requests and verify VCs. The verifier executes most of the SSI actions, i.e., it requests the data by creating a request that asks for specific VCs, then it verifies the VCs received as a response, i.e., checking the signature and other proofs of the veracity of both the construction as well as its contents.

- **Verifiable data registry** - mediates the creation and validation of identifiers (DIDs), keys, and other relevant data required for issuing, exchanging, and revoking VCs. Example verifiable data registries include DLTs.

8.2.3 SSI Elements

This Sub-Section describes the SSI core elements presented in the Verifiable Credentials specification Longley et al. (2019).

- **claim** - An assertion made about a subject, i.e., a subject-property-value relationships such as “Alice is student of the University of Bologna”.

- **credential** - A set of one or more claims made by an issuer. Then this is rendered tamper-evident, and it is authored through a cryptographically verifiable method, e.g., a digital signature, it can be defined Verifiable Credential (VC).

- **presentation** - Data derived from one or more verifiable credentials, issued by one or more issuers, that is shared with a specific verifier. Verifiable presentation is data that may be different from the VC from which it is derived while also being used to verify the credential claim, e.g., through zero-knowledge proofs Feige et al. (1988), and the issuer's authorship.

- **proof** - One or more cryptographic proofs can be used to detect tampering and verify the authorship of a credential or presentation, e.g., a digital signature.

- **data schema** - used to enforce a specific structure on a given collection of data, e.g., a specific type of VC. Data encoding schemas map the contents of a VC to a machine-readable format for specific attributes. Data verification schemas are used to verify that a VC's structure and contents conform.

- **decentralized identifier (DID)** - A portable URL-based identifier associated with an entity. These identifiers are most often used in a VC and are associated with subjects such that a VC itself can be easily ported from one repository to another without the need to reissue the credential. An example of a DID is the one shown in Chapter 7 for the *URI* of an *IP Entity* object: `did:nft:eip155:1_erc721:0xa437b30051601bd54ffee7de357b28e1488929rt_1`.

- **DID document** - a document accessible using a verifiable data registry and containing information related to a specific DID and describing the DID subject, i.e., the entity identified by the DID. The DID document also includes mechanisms, such as cryptographic public keys, that the DID subject can use to authenticate itself and prove its association with the DID.

- **DID controller** - an entity that can make changes to a DID document. A DID might have more than one DID controller.

- **DID method** - specifies the precise operations by which DIDs and DID documents are created, resolved, updated, and deactivated.

- **DID scheme** - the formal syntax of a DID, i.e., the DID scheme provides the prefix `did:scheme:` to associate with a DID method. In a specific DID scheme, the DID method name follows the first colon and terminates with the second colon. The DID method has a specification that defines a specific DID method scheme that works with that specific DID method.

8.3 Self-Sovereign Identity layer design

In this Section, we provide an overview of the SSI layer design, where the components of our decentralized PIMS are integrated into the SSI ecosystem. Moreover, we describe how we expanded the DID and VC models with an Intelligible Model.

8.3.1 Actors and architectural components

Our first aim is to compare the SSI roles to the roles of the actors involved in the architecture of our decentralized PIMS. Then we will describe the use of PIMS components in the SSI ecosystem. We use Figure 8.2 as a reference for the description.

8.3.1.1 Actors

Based on the identified actors of Chapter 6, we reconsider the roles described in previous Section:

- ***Data subject*** (DS) - the same subject protagonist of a claim. Indeed, in most cases, the claim is a piece of personal data, and thus it can be treated similarly. This means that in the following, the handling of VCs and DID documents is compared to the handling of personal data.

- ***Data holder*** (DH) - the SSI holder, as it holds a piece of personal data, i.e., the VC. Some data holders can also be VC issuers.

- ***Data intermediary*** (DI) - the one at the center (together with other intermediaries) of the verified transaction execution. They can assume the role of the holder, verifier, or verifiable data registry maintainer depending on the service provided:

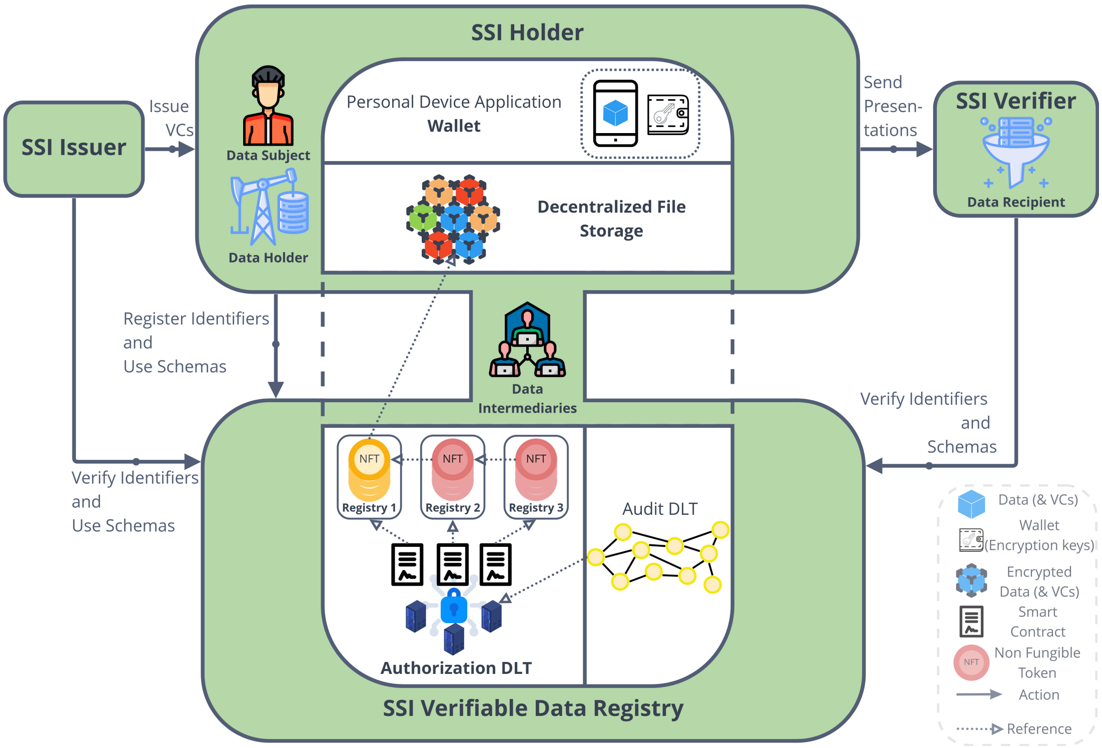


Diagram with reference to the diagram of Figure 8.1.

Figure 8.2: Decentralized PIMS components within the SSI ecosystem

  - ***DFS provider*** (SP) - a SSI holder that cannot access VCs because the files in the DFS are encrypted. They help the subject and/or holder store the VCs.
  - ***Authorization server*** (AS) - SSI holder that receives credential requests from verifiers. This VC request will be treated exactly as the data access request was treated (thus, as seen in Chapters 6 and 7). Moreover, the authorization servers are also verifiable data registry maintainers.
  - ***Audit DLT node*** (AN) - Since our decentralized PIMS is based on a multi-DLT architecture, the permissionless audit DLT is also included in the verifiable data registry functioning. Thus the audit DLT nodes are verifiable data registry maintainers too.

- ***Data recipient*** (DR) - the SSI verifier.

8.3.1.2 Components

Regarding Figure 8.2, in the following, we describe the components of our PIMS architecture in the SSI ecosystem:

- **Personal device application** - Described in Chapter 4. The personal device application (for the subject) or the device application (for the data holder and recipient) maintains the wallet for managing asymmetric keys and digital signatures. This wallet can also temporarily store the VC issued by SSI issuers. The application is also used to interact with the DFS and authorization DLT as described previously, e.g., to register DIDs similarly as encrypted personal data hash pointers are registered on-chain. Moreover, it is also used to send verified presentations if the associated VCs are stored in the wallet.

- **Decentralized File Storage** - Described in Chapter 4. Since it already embodies the PDS, the DFS can be used as the SSI wallet definition to store VCs. Moreover, it also stored DID documents. However, these are encrypted, and the access control is based on using the authorization DLT.

- **Distributed Authorization**

  - **Authorization DLT** - Also, in this case, we leverage the (semi-)private permissioned authorization DLT to orchestrate the access control to VCs and DID documents. In Chapter 7, we have seen how data (and policies) references are stored in the ledger using an NFT representation. This is also used for storing VCs and DID document references, i.e., on-chain hash pointers. The authorization DLT is also used for indexing DIDs and storing data schemas.
  - **Audit DLT** - The public permissionless audit DLT is used as a security mechanism for the authorization DLT because it stores periodical commitments. The audit DLT can also be used for indexing DIDs, e.g., institutional SSI issuer DIDs, and storing data schemas.

8.3.2 The Intelligible Model

In this Chapter, our contribution is not limited to the integration of the PIMS to the SSI ecosystem; in fact, we also provide an expansion of the DID and VCs model towards the aim of intelligibility for both users and machines. We argue that the intelligibility and machine readability of what is contained in digital documents and processes based on them must be ensured as much as possible. Using an Intelligible model means that users can understand how processes and decisions affect them, feel in control, and trust the system they interact with. The first step is to provide resources that are interconnected and provided with an operational and legal context. We stem from the properties that make contracts intelligible Cervone et al. (2020) in a distributed environment and derive the models for digital identities and certificates. These two are often intertwined in processes that involve the verification of data in electronic transactions: a digital certificate result in a digital document storing a statement of conformance and is signed by the certifying party, represented by a digital identity. An intelligible model embedded within the SSI ecosystem is in an excellent position to embody both credential verification and human-intelligible artifacts management. Moreover, such a model considers that digital certificates and identities are usually not standalone but are embedded within a context and may refer to or may have to comply with a complex set of local, national, and international legislation and regulations. Thus, it provides the possibility to track the legal basis (e.g., purpose, public interest), the scope (e.g., data altruism), and the legal consistency with some norms (e.g., competition law).

8.3.2.1 Intelligible Decentralized Identity

The Intelligible Decentralized Identity (Intelligible DID) model is used to represent the identity of a person (but also object and organization) by unequivocally binding the identity information to a public key (and the derived DLT address). It can be compared to the traditional public key certificate within Public Key Infrastructures (PKI) since its main role is to prove the ownership of a public key used for authentication purposes Preukschat and Reed (2021). However, while PKIs are formed by certification authorities (CA) that issue digital certificates binding a public key with an identified entity, the Intelligible DID follows the SSI vision. The Intelligible DID is ultimately implemented as a DID document, and thus, a DID is used to refer to an Intelligible DID. What our proposal adds to the DID document is the possibility of bringing with it the relevant operational and legal context of this identity and easily tracing the processes that involve it.

The Intelligible DID model is a combination of (i) asymmetric key pairs, i.e., a public key and a private key; (ii) an NFT stored on a dedicated NFT registry in the authorization DLT; (iii) an intelligible identity document stored in a DFS.

Key Pair The key pair is at the core of identity authentication. Using a digital signature as a binding cryptographic method enables any entity to be represented by its public key since, by signing a digital document using the associated private key, anyone can verify that the signature is associated with the key pair's public key. The public key is used to authenticate the identity publicly and is used in any DLT in the form of an address. Conversely, the private key is stored (and protected) in the wallet of the (personal) device application of the identified entity, i.e., subject, holder, recipient, or service provider. The private key is kept safe since it is the only information needed to digitally sign on behalf of that entity, i.e., authenticate. Examples of a private key, public key, and the DLT address generated from the latter using the Ethereum protocol Buterin et al. (2013) are shown in Figure 8.3.

Non Fungible Token In this model, an NFT is used as a decentralized representation of the Intelligible DID, as one of such can be considered unique. We have already seen how NFTs can uniquely represent a piece of information in Chapter 7. Moreover, using the NFTs allows interoperability between platforms that already implement interfaces with these, thus making it possible to use Intelligible DID in different contexts.

The Intelligible DID model involves using an NFT registry smart contract that can be considered a registry for issuing new identities. In particular, the registry maps the DLT address derived from the key pair to a specific NFT id, i.e., the address is the owner of the NFT. As shown in Figure 8.3, the NFT is used to derive a DID representing the Intelligible DID. We use a modified version of the NFT DID method proposed by Thorstensson (2021). The NFT DID Method converts any NFT on any DLT into a DID where the owner of the NFT is the DID controller. This is achieved by referring to the NFT owner address and (in our case) to the NFT metadata. When an NFT is minted, the

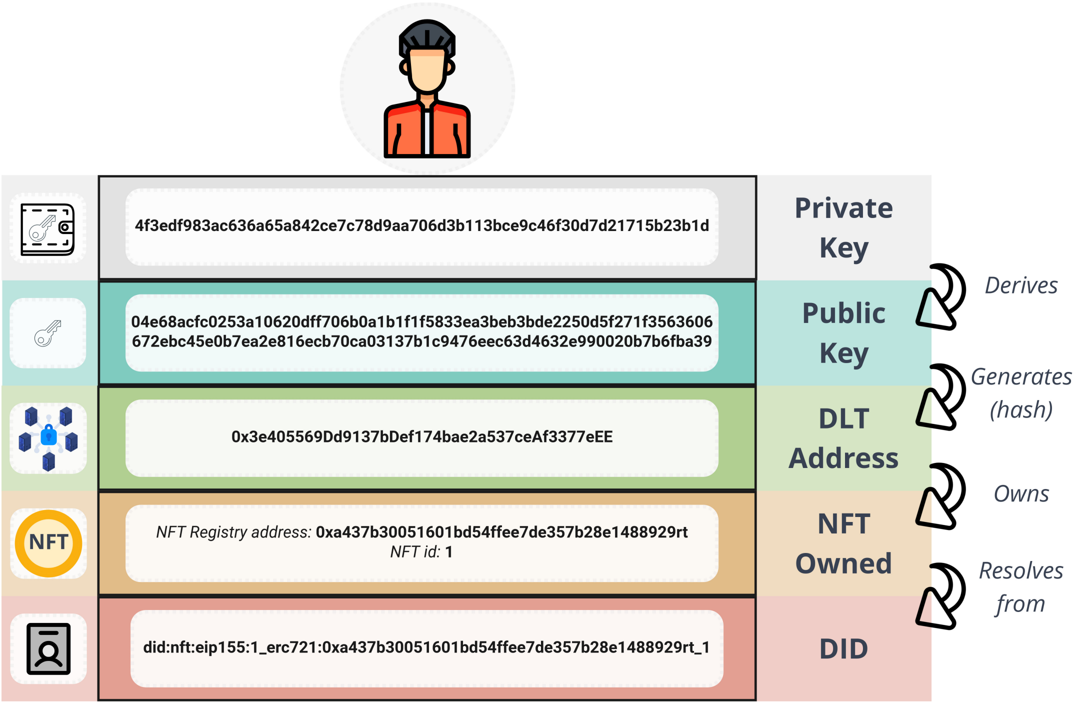

4f3edf983ac636a65a842ce7c78d9aa706d3b113bce9c46f30d7d21715b23b1d
Private Key
04e68acfc0253a10620dff706b0a1b1f1f5833ea3beb3bde2250d5f271f3563606672ebc45e0b7ea2e816ecb70ca03137b1c9476eec63d4632e990020b7b6fba39
Public Key
Derives
0x3e405569Dd9137bDef174bae2a537ceAf3377eEE
DLT Address
Generates (hash)
NFT
NFT Registry address: 0xa437b30051601bd54ffee7de357b28e1488929rt
NFT id: 1
NFT Owned
Owns
did:nft:eip155:1_erc721:0xa437b30051601bd54ffee7de357b28e1488929rt_1
DID
Resolves from

Figure 8.3: Diagram showing the generation of an NFT DID from a public and private key pair.

NFT id and NFT metadata are specified and cannot be further modified. The metadata contains an immutable URI to an off-chain intelligible identity document.

Intelligible Identity Document The core of the Intelligible DID consists of a document (or set of documents) structured in a format that is both intelligible and machine-readable, e.g., using XML. This document includes all the data related to the identity, in the form of VCs, together with references to parties, software, and other legal and operational documents involved in creating and issuing the Intelligible DID. The intelligible identity document is divided in:

- **Meta**: includes references to external documents representing persons, organizations, and objects referenced in the Body section of the document. The intelligible identity document generally supports generic and legal documents, which can be included in creating an Intelligible DID. References may also link to ontologies, for instance, for defining participants' roles. Finally, references also include the link to the NFT in the registry.

- **Body**: includes the actual contents that identify the subject entity, i.e., a set of VCs to be shared with verifiers that prove that the identified entity has ownership of claims. In the case of a person, those would be, for instance, name and email. In particular, the VCs are extended as Intelligible Verifiable Certificates (Intelligible VCs), a model described in the following Sub-Section.
- **Signatures**: digital signatures of the parties involved, such as the issuer of the Intelligible DID (which can be the subject itself) and the signature of the software issuing the Intelligible DID.

It is important to stress that, even if logically indicated as a single document, it can be considered a document storing a collection of documents. For instance, the Meta and Body parts can be included in one document, and the Signatures in another since the digital signature need to include the hash of the first document Damgård (1989).

8.3.2.2 Intelligible Verifiable Certificate

The Intelligible Verifiable Certificate (Intelligible VC) model is used to expand verifiable credentials or, in general, to represent a digital certificate. In its most elementary definition, it represents an act of attestation by a third party, i.e., certification of content. In addition, the certificate issuer attests to specific claims concerning an object, a digital asset, or an Intelligible DID.

The creation of this model, indeed, stems from the fact that usually existing digital documents representing credentials do not provide (i) an operational and legal context, (ii) intelligible and transparent information regarding the boundaries, or (iii) both. Our model aims to produce a trusted and accountable system. These enable a set of rules to become self-executable via artificial artifacts (e.g., sensors, IoT, app) in order to create disintermediated services that are reliable and legally valid.

An Intelligible VC consists of a combination of a set of (i) Intelligible DIDs, (ii) a Non Fungible Token, and (iii) an intelligible certificate document stored off-chain.

Intelligible DIDs and Signatures An Intelligible VC is linked to one or more identities represented by Intelligible DIDs, e.g., the ones who digitally sign the VC. An Intelligible VC might be issued by an organization represented by an Intelligible DID and act, i.e., sign, through a person with his or her own Intelligible DID. The VC receiver owns another Intelligible DID, and an object associated with him or her may constitute the purpose of the Intelligible VC. All the parties involved might be requested to sign the Intelligible VC, and such a sequential or parallel process is traced in the DLT, considering the appropriate unique identifiers for each new edit. Finally, it is important also to consider the software environments and user interface renderings when a signature is made, particularly in the case of a receiver signing. Indeed, for verifiability and tracing back in the case of legal dispute, it is essential to make a "snapshot" of the graphical user interface by using an encoding that allows reproducing the same environment a user was presented to when the signature was made. This operational context document is included or referenced in the intelligible certificate document.

Non Fungible Token Even in this case, an NFT is leveraged to represent the uniqueness of an Intelligible VC in a DLT. It is important to stress that here (and for the Intelligible DID in the previous subsubsection), the NFT is used as a smart contract standard to represent the uniqueness of information, i.e., the certificate (or the identity). A smart contract is also issued and considered as a registry, such as in the case of identities. For Intelligible VCs, however, writing to such a registry might be allowed to the issuer organization and its members. Therefore, each organization or public registry can have a registry (i.e., a smart contract). In this case too, the NFT is used to create a DID representing the Intelligible VC, and the NFT metadata will point to an intelligible certificate document stored in the DFS.

Intelligible Certificate Document The document structure is the same as the Intelligible DID one, as we tried to keep it as simple as possible to be extended. However, the Intelligible VC body can contain more diversified information. Intelligible certificate documents can reference more or less complex hierarchies of documents for the legal and operational context.

The legal context references the legal environment under which the Intelligible VC is described. Thus, it references legal documents of diverse nature, such as acts cited within the certificate claims or licenses. We stress that legal documents must be serialized in a way that makes them human and machine-readable, structured, serialized through standard technologies, and linkable to other resources. More precisely, metadata must be specified for each legal document, and the markup language used for legal documents must provide features to identify and give semantic meaning to all data contained in documents that have legal significance. For instance, legal metadata may include the date of issue, date of execution, date of signature, changes to the document, information about its versions, and so on.

The operational context of Intelligible VCs is a set of references to operational parts for the exploitation of the certificate. The performed operations mainly involve verifying the claims and actions based on the parameters set.

8.3.2.3 Features that make the model Intelligible

The first step to convey intelligibility in our proposed models involves using different technologies. In our specification and implementation, we exploit: (a) a technology that executes source code written in plain English; and (b) three specifications of the Akoma Ntoso (AKN) LegalDocML OASIS standard Palmirani and Vitali (2011) for (i) human readable URIs naming convention Vitali et al. (2019), (ii) structuring legal documents Palmirani et al. (2018b), (iii) describing and connecting entities using an informal ontology Vitali et al. (2018).

Zenroom Zenroom Roio (2019) is a language interpreter that can be integrated into any application to authenticate and restrict access to data and execute human-readable smart contracts. These smart contracts are written in the specific language of Zencode, which reads in a natural language, similar to English, and makes smart contracts look closer to legal ones. The Zencode can be used to implement signature and verification according to the W3C VC specification. For this reason, we refer to this language to create the VCs used in the Intelligible DID and VC. An example of key pair creation for an Intelligible DID is shown in Listing 8.1.

```
rule check version 2.0.0
Scenario ’ecdh’: Create the keypair
Given that I am known as ’DIDController’
When I create the keypair
Then print my data
```

Example 8.1: Source code executed in Zen room for the issuing of an Intelligible DID key pair

The zencode for issuing a credential is included in the document referenced by the intelligible identity and certificate documents.

Akoma Ntoso The Akoma Ntoso (AKN) LegalDocML OASIS standard for modeling legal resources Palmirani and Vitali (2011) has been effective in some different legal and non-legal contexts Gen et al. (2015). We leverage the AKN XML vocabulary to mark up the legal documents that intelligible identity and certificate documents may refer to. This vocabulary's primary purpose is associating XML markup with legal and parliamentary resources such as acts, bills, and debate reports. However, it is an extensible and customizable language that currently supports more than three hundred elements and twelve document types Palmirani et al. (2018b). For instance, in our implementation, we use "documentCollection" that is "*used to represent documents which are collections of other independent documents.*" Palmirani et al. (2018b). In the case of the Intelligible VC, other legal documents that belong to the certificate can be represented using the doc element, which is intended for all the documents with legal validity but whose structure is not restricted to any predefined one. Not only legal documents and concepts belonging to the legal context can be identified using the AKN convention, but also entities and concepts belonging to the operational context. This can be made through the informal ontology supplied with AKN Vitali et al. (2018).

Finally, AKN offers the possibility for the AKN document collections to include graphical and informational documents related to the main document, e.g., a website source code and environmental variables snapshot taken during the digital signature placement moment. This greatly helps trace the relevant document and increases what can be displayed to the user as an explanation, thus augmenting the system's intelligibility.

Requirements for an "Immutable" Document Identification A key property of the models presented above is the immutability of their constituent documents and the unique identification of the documents composing the Intelligible DID and VC. DLTs provide the means to store immutable information, but it needs to be specified how to leverage this in document identification. We use a combination of URI Berners-Lee et al. (2004), and the result of a hash function applied to the document involved Damgård (1989).

The URI must be specified in a manner that is human and computer readable and can make intelligible the hierarchical semantics of the resource it points to, the documents versioning, the references to legal documents, and operational context documents. For instance, URIs must be designed to point to folders, subfolders, files, file partitions, and so on, and if a resource is managed within a specific environment or protocol, then the URI must specify them.

Hashes computed on the content of resources specified by URIs should be self-describing, too, or prefixed by the protocol name, e.g., *keccak-256:3ca6eb64994(...)* Kavun and Yalcin (2010). Since hash functions are computed on the content of resources, URIs should resolve to immutable resources that were available at a specific time and thus should be designed to express this concept.

We refer to the AKN specifications for our intelligible models to link legal documents and concepts to pairs of URIs and hash pointers. For instance, using the AKN naming convention enables the following: (i) to represent intelligible identities and certificate documents and all the other documents involved, both legal and operational, through IRIs, i.e., International Resource Identifier; (ii) to specify hierarchical relations between the documents; (iii) to allow versioning of the documents; (iv) to express references of entities, whether Intelligible DIDs represent them or not. A hierarchy of four concepts is used for the naming convention, following the FRBR conceptual model Vitali et al. (2019):

1. **work** - represents the most abstract concept of the resource, e.g., `/akn/EU/doc/2022-06-09/did-nft-eip155-1_erc721-0xa437b30051601bd54ffee7de357b28e1488929rt_1/main`;

2. **expression** - used to identify specific temporal and linguistic versions of the resource, e.g., `/akn/EU/doc/2022-06-09/did-nft-eip155-1_erc721-0xa437b30051601bd54ffee7de357b28e1488929rt_1/eng@!main`;

3. **manifestation** - a serialization of a version in a specific data format, e.g., `/akn/eu/doc/2022-06-09/did-nft-eip155-1_erc721-0xa437b30051601bd54ffee7de357b28e1488929rt_1/eng@/main.xml`;

4. **item** - a specific file where a manifestation is stored, e.g., `ipfs://QmdruQGEeXugWYWQ7Co5H5XjGqoitC59tDc3DLpCrCp35x/akn/eu/doc/2022-06-09/did-nft-eip155-1_erc721-0xa437b30051601bd54ffee7de357b28e1488929rt_1/eng@/main.xml`.

As can be seen from the examples, the item is the one that includes the hash pointer information, and the AKN IRI follows it. Again we refer to IPFS as the implementation of the DFS to use the IPFS CID in the following examples. The AKN IRI has a specific set of rules for its composition. For a work, this is the conceptual structure: /akn/{jurisdiction}/{docType}/{subType}/{actor}/{date}/{docNumber}. Since AKN allows setting a unique string of characters as the *docNumber*, we use this field to insert the NFT DID that refer to an intelligible identity or certificate document (and we substitute the column ":" character with the dash character "-").

For instance, the metadata of the NFT resolved from did:nft:eip155:1_erc721:0xa437b30051601bd54ffee7de357b28e1488929rt_1 would be the item ipfs://QmdruQGEeXugWYWQ7Co5H5XjGqoitC59tDc3DLpCrCp35x/akn/eu/doc/2022-06-09/did-nft-eip155-1_erc721-0xa437b30051601bd54ffee7de357b28e1488929rt_1/eng@/main.xml. Thanks to the use of the hash pointer ipfs://QmdruQGEeXugWYWQ7Co5H5XjGqoitC59tDc3DLpCrCp35x it is possible to point to the package including all the other documents referred by the intelligible identity or certificate document. Figure 8.4 shows an example of the combination of AKN IRI and hash pointers for an Intelligible DID and VC.

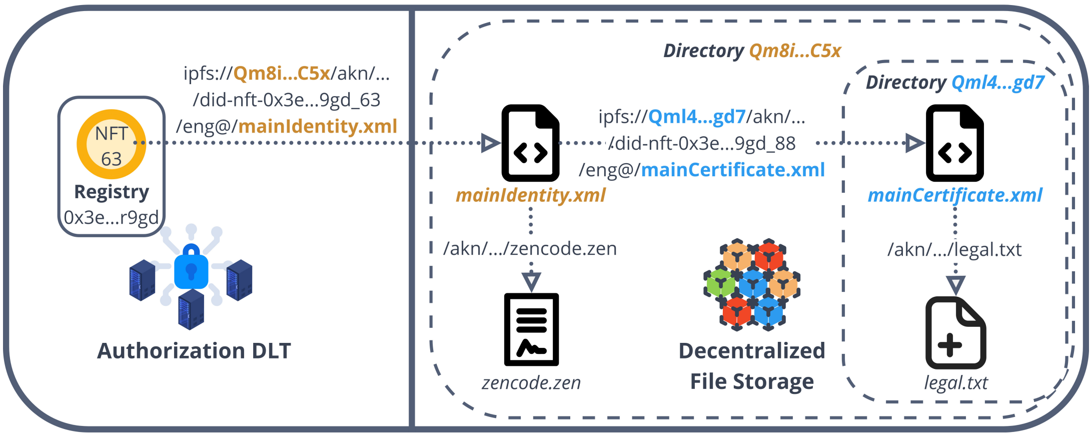


Figure 8.4: Example of a combination of AKN IRI and hash pointers for an Intelligible DID and VC.

8.4 Implementation and Use Cases

This Section describes the implementation and use cases to evaluate the proposed SSI layer. Our prototype of Intelligible DID and VC can be found in Zenodo Zichichi

(2022d). Here we are interested in the main advantages that an intelligible model can bring together with an SSI ecosystem. In particular, we implemented the smart contracts for the Intelligible DID and VC and related document creation, and we described two use cases to show their functionalities.

In the following, we first evaluate the implementation of the smart contracts, then we provide a use case for the authentication through the Intelligible DID and a data sharing scenario where datasets are certified through the use of the Intelligible DID and Certificate.

8.4.1 Implementation

In this Sub-Section, we provide a brief description of the Intelligible models' implementation, an evaluation by measuring our experiments in terms of gas (as per the Ethereum protocol), and a use case for discussing the usability of the Intelligible DID for the user.

8.4.1.1 Intelligible Decentralized Identity and Verifiable Certificate

The Intelligible DID and VC implementation has been developed as the composition of several modules interacting with the decentralized PIMS that can be in common or specific:

- The **intelligible-doc** module allows the creation and parsing of XML documents. Two specializations of this module are the **intelligible-akn**, for managing AKN documents, and **intelligible-sign**, for managing signature documents.

- The **intelligible-storage** module manages the interaction with a DFS where to store documents and handle hash pointers. Its specialization is **intelligible-storage-ipfs** for interacting with an IPFS network.

- The **intelligible-nft-web3** module handles smart contract templates and the interaction with NFT registries in the authorization DLT.

- The **intelligible-identity** module implements the Intelligible DID model by combining the previous three modules and specializing in creating documents for identities. Moreover, it includes sub-modules for issuing key pairs (using Zencode) for creating the DID and the signature of documents using the key pair.

Table 8.1: Methods' gas usage for the NFT registry Smart Contract.

Method	Gas Usage
newToken()	216046
newTokenFromReserved()	157204
reserveId(1)	53562
reserveId(2)	77162
reserveId(3)	100762

- The **intelligible-certificate** module implements the Intelligible VC model by combining the previous four modules and specializing in creating documents for certificates.

The implementations can be found in (Zichichi, 2021a,b, 2022a,d).

8.4.1.2 Smart Contracts implementing NFT registry

The results in table 8.1 show the use of gas for the methods involved in executing the application and their related Smart Contracts. Gas is a unit that measures the amount of computational effort it takes to execute operations in Ethereum Smart Contracts. Thus, the higher the gas usage for a method, the more intense the computation of a blockchain node to execute the method is. The gas usage of the NFT registry smart contract methods is relatively low and does not deviate much from the other similar application implementations in Ethereum. For instance, the $newToken()$ method has a gas usage of $\sim 216k$. We implemented a method for reserving registry ids in batch and then creating new NFTs from reserved id, using the $newTokenFromId()$ method. Results in table 8.1 show that this approach is more convenient in the case of the creation of several tokens in the same operation. For instance, creating 3 tokens with $newToken()$ has a gas usage of $\sim 216k \times 3 =\sim 648k$, while creating those with the second approach has a gas cost of $\sim 100k + (\sim 157k \times 3) =\sim 571k$, i.e., the cost of $reserveId(3)$ plus three times $newTokenFromId()$.

8.4.1.3 Intelligible Decentralized Identity authentication

In this Sub-Section, we demonstrate how the Intelligible DID can be used for authentication purposes. The NFT representation enables interoperability between web platforms that already implement software interfaces with this standard. Indeed, the NFT, in the

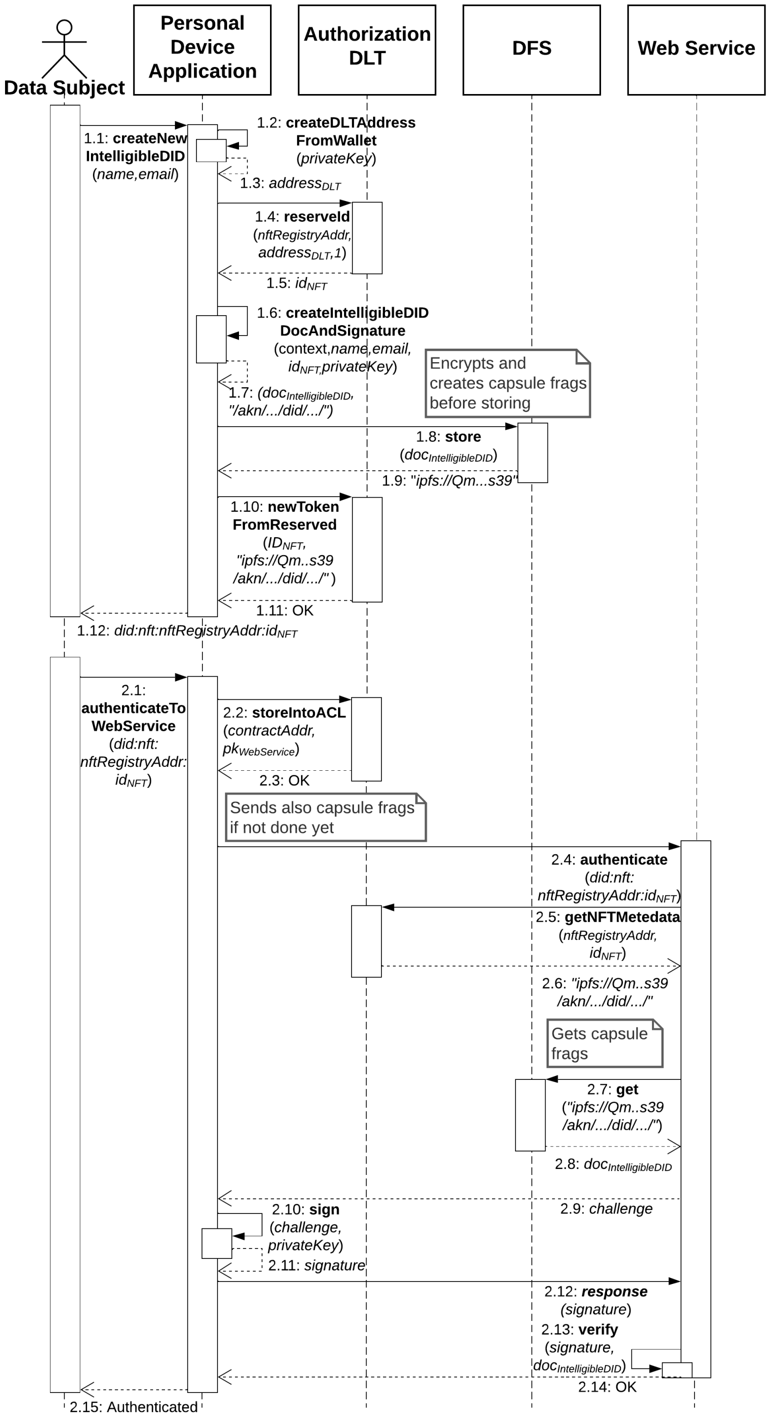

Data Subject
Personal Device Application
Authorization DLT
DFS
Web Service
1.1: createNew IntelligibleDID (name,email)
1.2: createDLTAddress FromWallet (privateKey)
1.3: address$_{DLT}$
1.4: reserveId (nftRegistryAddr, address$_{DLT}$,1)
1.5: id$_{NFT}$
1.6: createIntelligibleDID DocAndSignature (context,name,email, id$_{NFT}$,privateKey)
1.7: (doc$_{IntelligibleDID}$, "/akn/.../did/.../")
Encrypts and creates capsule frags before storing
1.8: store (doc$_{IntelligibleDID}$)
1.9: "ipfs://Qm...s39"
1.10: newToken FromReserved (ID$_{NFT}$, "ipfs://Qm..s39 /akn/.../did/.../")
1.11: OK
1.12: did:nft:nftRegistryAddr:id$_{NFT}$
2.1: authenticateTo WebService (did:nft: nftRegistryAddr: id$_{NFT}$)
2.2: storeIntoACL (contractAddr, pk$_{WebService}$)
2.3: OK
Sends also capsule frags if not done yet
2.4: authenticate (did:nft: nftRegistryAddr:id$_{NFT}$)
2.5: getNFTMetedata (nftRegistryAddr, id$_{NFT}$)
2.6: "ipfs://Qm..s39 /akn/.../did/.../"
Gets capsule frags
2.7: get ("ipfs://Qm..s39 /akn/.../did/.../")
2.8: doc$_{IntelligibleDID}$
2.9: challenge
2.10: sign (challenge, privateKey)
2.11: signature
2.12: response (signature)
2.13: verify (signature, doc$_{IntelligibleDID}$)
2.14: OK
2.15: Authenticated

Figure 8.5: Intelligible DID challenge-response authentication

form of ERC 721 Entriken et al. (2018), is currently widely used, and many websites already support its interface. This means that Intelligible DID can be implemented in several already functioning web services with low effort. In web services, users can operate through their software personal device application wallet, e.g., Metamask Metamask (2021), and prove that they hold an Identity DID simply by revealing their NFT DID. The web service will access the intelligible identity document in the DFS and check the user's claims.

The authentication method is based on a simple challenge-response authentication shown in Figure 8.5. Any service that supports the NFT standard can perform an NFT lookup after a user authentication request. The service can verify the Intelligible DID by checking it against the user signature through the retrieved data. Users interact only with the personal device application wallet that locally secures private keys and executes signature operations. A working demo can be found in `https://intelligible-demo.herokuapp.com`.

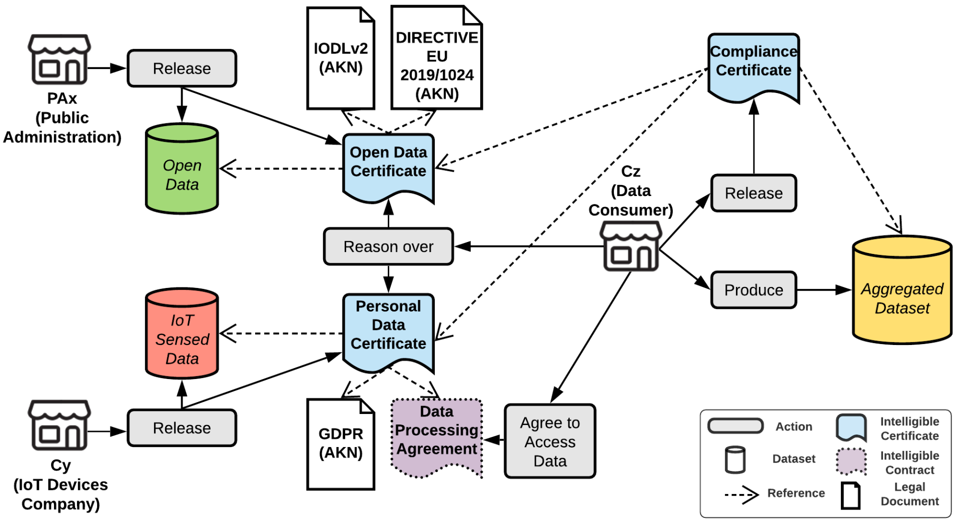


Figure 8.6: Open government data use case diagram

8.4.2 The Use of an Intelligible System for the Sharing of Open Government Data

Specifically, we involve the use of Open Government Data that can be defined as "*public sector information, offered paid or non-paid for the non-commercial and commercial re-use,* *available in a machine-readable format, interoperable, and likely covered by open licenses or minimal restrictions to re-use it"* Mockus (2017). Open data shareability is usually limited by different legal norms defined in different acts. For instance, they must comply with the GDPR, as they cannot create discrimination or affect some specific group. Thus, they have to be released under license conditions. Moreover, companies' open data should be used under special agreements or contracts to respect the competition law or confidentiality clauses. All these constraints limit the circulation of open data and their release. The existing platforms releasing open data do not provide intelligible and transparent information about these limits.

The idea of using blockchain for managing open data has been introduced previously. Nevertheless, with the following use case, we want to show the innovation of our solution, that is (i) to include a smart contract for modeling normative rules using certificates and (ii) to manage the modifications over time of the normative system in an automatic way using AKN and LegalRuleML standards. Our solution could help to remove resistance in the end-user to share open data, in the PAs to publish them, and in the companies to disclose big data.

8.4.2.1 Scenario

Intelligible VCs help to maintain a high level of trust in such a data sharing scenario because of the authentication and traceability capabilities they provide. For instance, an entity might be interested in releasing a dataset with a certificate stating that the data included has undergone a process of anonymization. In order to be fully exploitable, this certificate must be readable by both an individual and a machine, i.e., the case of an Intelligible VC. In particular, the certificate will reference several AKN marked-up legal documents that provide the use limitations. Another entity might be interested in aggregating such a dataset with another dataset containing personal data released under certain conditions by a data controller. In the following, with the aid of Figure 8.6, we will show how these entities can interact to reach their goals through an Intelligible System.

8.4.2.2 Public Administration Releases Open Data

A public administration henceforth referred to as PAx, produces a dataset and releases it in the form of Open Data. We particularly focus on this use case of datasets released by Public Sector Institutions due to its importance among researchers. Thus, PAx releases this dataset that we imagine in this simple scenario to be the routes of buses of Milan city in Italy. The URI of such dataset *item* could be, for instance, `ipfs://QmU...J4mN/akn/references/expression/dataset/PAx/`, making use of the AKN naming convention and IPFS hash pointer. PAx then can release an Open Data (Intelligible Verifiable) Certificate where it refers to the dataset, together with reference to two other documents that form the legal context, namely the Italian Open Data License (IODL) v2.0 Formez PA (2020) and the EU Directive 2019/1024 European Parliament (2019). PAx references the IODL to enable users to freely share, modify, use, and re-use the dataset. While in the second case, the EU directive 2019/1024 on open data and the re-use of public sector information is referenced to explicitly define the anonymization process enacted for the dataset[7]. An excerpt of the references included in the Intelligible VC document would be:

```
<references source="#editor">
 <TLCReference
   eID="dataset-expression"
    name="Dataset"
    showAs="Bus Routes Dataset"
    href="ipfs://QmU...J4mN/akn/references/expression/dataset/PAx/">
 </TLCReference>
 <TLCReference
   eID="iodl-expression"
    name="IODLv2"
    showAs="Italian Open Data License v2.0"
    href="ipfs://QmU...k2/akn/it/doc/IODL/2012-03-05/2/ita@2020-05-05/PAx.akn">
 </TLCReference>
 <TLCReference
   eID="odpsi-work"
    name="odpsi"
    showAs="EU directive 2019/1024 on open data and the re-use
    of public sector information"
    href="/akn/EU/act/regulation/EU/2019-06-20/1024@2019-06-20">
 </TLCReference>
</references>
```

Example 8.2: References in the intelligible certificate document

With the use of those references, an instance of the body part for the intelligible certificate document can include this data:

[7] Article 2(7): *"'anonymization' means the process of changing documents into anonymous documents which do not relate to an identified or identifiable natural person"*

```xml
<mainBody>
 ...omissis...
 <blockList id="lst1">
  <listIntroduction>Legal Information</listIntroduction>
  <item id="lst1-itma">
   <num>(a) </num>
   <p>This data is available under <ref refersTo=
   "#iodl-expression"> Italian Open Data License v2.0 </ref>
   </p>
  </item>
  <item id="lst1-itmb">
   <num>(b) </num>
   <p>This data contains documents that comply with the
   requirements of anonymization in relation with the
   <ref refersTo="#odpsi-work"> EU directive 2019/1024 on
   open data and the re-use of public sector information</ref>
   </p>
  </item>
  ...omissis...
 </blockList>
</mainbody>
```

Example 8.3: Mark-up of the intelligible certificate document body

8.4.2.3 IoT Devices Company Releases Personal Data

In the same scenario, we consider an IoT Devices Company, henceforth referred to as Cy, that collects data from the sensors of the devices it produces. For this use case, we consider a dataset of information sensed from smart card devices when the users are on board a bus. Since this data consists of information relating to an identified or identifiable natural person, it falls under the definition of personal data of the GDPR. Thus Cy makes use of a Data Processor Agreement to release such dataset. AKN's naming convention may be used for Intelligible Contracts that similarly serialize a DPA template to what we showed in the previous subsection. Here we focus more on the operational part of the Intelligible VC. Indeed, together with the dataset containing personal data, the certificate can be released, and its document can directly link to the Smart Data Processing Agreement (DPA) shown in Chapter 7. Both model implementations will live in the same platform, i.e., the authorization DLT, and, thanks to the linkability properties shown in this Chapter and the previous one, a Smart DPA can reference a personal data Intelligible VC NFT directly. This means that the first phase of the verification happens on-chain. Furthermore, the authorization servers can be involved in providing access to personal data using the distributed mechanism of the decentralized PIMS. For the implementation in this use case, we consider the dataset stored in encrypted storage owned by Cy. To access the decryption keys and thus access the data, a privacy policy will be implemented and stored in the authorization DLT. In the following Sub-Section, we will show how a data recipient can prove to be eligible to be enlisted in such a list.

8.4.2.4 Data Recipient Aggregates the Datasets

We consider a third actor in our scenario interested in access to the dataset released by both PAx and Cy. This Data Recipient Company henceforth referred to as Cz, has access to the Intelligible VCs presented above and to all the referenced documents, including the Intelligible DIDs for PAx and Cy. Cz is thus able to reason over the two datasets' certificates from both a human and a machine perspective. For instance, any individual representative of Cz can go through the certificate and the AKN legal text referenced via URIs using software such as Lime[8]. From the point of view of machine readability, software such as one based on LegalRuleML can bridge the legal prose to operational code, maintaining the same intelligibility. LegalRuleML is an OASIS standard and a human-readable and machine-readable interchange language for rules in the legal domain Athan et al. (2013). It enables Legal Knowledge Engineers to highlight and connect business rules contained in legal documents to automatic legal reasoners enriched by Linked Open Data information. In our use case, LegalRuleML allows the reasoning over the AKN documents that form the certificate's legal context; for instance, the case for the Data Processing Agreement has been shown in detail in Cervone et al. (2020). From the Open Data Intelligible VC, Cz can infer that PAx's dataset is anonymous regarding the EU directive 2019/1024, but most importantly, that it is possible to create a Derivative Work through combination with other information, i.e., a mashup Mockus (2017), due to the IODLv2 Formez PA (2020). However, such a license explicitly states that it does not constitute authorization to violate personal data regulations. Thus the reasoning of the Personal Data Certificate for the IoT Sensed dataset must lead to compliance with the GDPR. That is the case if Cz agrees with the Data Processing Agreement. Given the high risk of location privacy intrusion Keßler

[8] `http://lime.cirsfid.unibo.it/`

and McKenzie (2018), a particular clause can be arranged in the agreement stating that the processing must *"render all or part of Customer Personal Data anonymous in such a manner that the data no longer constitutes personal data"*. In such a case, we can imagine a scenario where Cz wants to produce an aggregated dataset mixing the bus hop on and off times and the bus routes to get a heatmap of mass gathering for the time of the day and city zone. For instance, this could be the case with a COVID-19 Risk Assessment Tool. Suppose no other information is shared in the resulting aggregated dataset other than time slot, zone id, and a mass gathering value, e.g., based on density. In that case, the dataset complies with the requirements found in the two processed datasets' Intelligible VC. In turn, Cz can produce a third Intelligible VC associated with the newly created dataset, where compliance with the PAx's dataset and, more importantly, compliance with Cz's Personal Data Certificate and Data Processing Agreement is noted. If for Cy, the execution of the LegalRuleML processing over these documents leads to compliance, then the new Cz's certificate can be signed by a Cy representative using the Intelligible DID, and Cz can be enlisted in the Access Control List for accessing the personal data dataset.

8.5 Conclusions

In this Chapter, we discussed the role that intelligibility and trust play in developing digital transactions through implementation of Intelligible DIDs and Certificates. Our technical solution and the use case described provides us with an evaluation of the goodness of the two models in verifying the authenticity of the claims of a digital document while providing intelligibility to the system.

The idea of using DLTs, particularly smart contracts, for managing this kind of information is not particularly new. The innovation of our solution is to include smart contract modeling the normative rules using certificates and to manage the modifications over time of the normative system in an automatic way using Akoma Ntoso and LegalRuleML standards. The implementation based on the decentralized PIMS and the Akoma Ntoso standard, together with the Public Sector Information and Personal Data Processing use case, led us to assess the value of an intelligible model for automatic claims verification based on a specific context, while maintaining intelligibility for the human interaction.

With the broad aim of supporting the digital economy while, in the meantime, respecting the fundamental legal rights of the system users, our solution provides a set of tools for implementing this approach. The use case presented for evaluating the system also shows the market opportunities for companies to release open data according to their data strategy.

Part IV

CONCLUSIONS

Chapter 9

Conclusions and Future Work

In this thesis, we made the first step toward creating a user-centered decentralized personal information management system, where data management logic is pushed toward the data subjects' control. The European Union (EU) regulatory framework helps to promote a view in favor of the interests of individuals instead of large corporations. The General Data Protection Regulation (GDPR) is the principal example. However, dedicated technologies are still needed to help companies comply while enabling data subjects to exercise fully. We tackled this issue through decentralized systems, i.e., Distributed Ledger Technologies (DLTs) and Decentralized File Storage (DFS). The structure of this dissertation followed an incremental approach to describing a set of decentralized systems and models that revolve around the data subject. The main benefit of such decentralized systems is that, by bringing together regulations and technologies, these can provide subjects with the ability to record their data in some interoperable personal data spaces (PDS), guarantee a path towards data sovereignty, and enable users to control what personal data they want to share. In this work, we referred to EU regulations, such as GDPR, eIDAS, and Data Governance Act, to build our PIMS architecture's functional and non-functional drivers. Moreover, we also analyze the GDPR compliance of our proposed PIMS at different levels.

9.1 Discussion

In our design, personal data is kept in a PDS consisting of encrypted personal data referring to the subject. A PDS based on centralized or decentralized file storage enables data persistence and a place where data can be shared after being encrypted. A DLT brings immutability and transparency feature to the whole process. We provided a prototype implementation developing the DLT component as an IOTA public DLT and leveraging IPFS and Sia as DFS components. Our implementation evaluation shows that: (i) a proprietary IPSA service node appears to provide stronger assurances for responsiveness and reliability; (ii) using IOTA as DLT is not viable for real-time applications but acceptable for less demanding services.

We also used IOTA as a decentralized indexing layer to guarantee the integrity, verifiability, linkability, and indexing of the encrypted personal data stored in the PDS. In this layer, we follow the approach to reference data and their content on a DLT, i.e., on-chain hash digests. Then, we associate to such hash digest reference a keyword set that is exploited to lookup for specific kinds of contents, all thanks to the use of a Distributed Hash Table (DHT), i.e., Hypercube DHT. We first showed the design of such a decentralized system and its implementation. Being r the Hypercube DHT dimension, i.e., the logarithm on the number of DHT nodes, on average $\frac{r}{2}$ number of hops (i.e., when a query message is passed from one DHT node to the next) are required for a punctual search based on a specific keyword set. When a broader search based on a particular keyword set and its supersets, i.e., Superset search, is executed, we have shown how the number of hops depends on the distribution of objects between DHT nodes.

On top of the decentralized indexing layer, a network of authorization servers acts as a data intermediary to provide data access to potential data recipients. Access to the data stored on a PDS can be allowed by the data holder through smart contracts. These maintain a data structure to record eligible data recipients, i.e., those to whom to issue the keys needed to access the encrypted data. Also, in this case, the GDPR tensions with DLTs drove the architecture to be a multi-DLT one. In light of such tensions, we introduced two architectural components to the PIMS. The first one is an authorization DLT based on a "tightly controlled" (semi-)private permissioned ledger, where a set of smart contracts allows data subjects to define access (through an Access Control List) to their data stored in their PDS. The access to the data is controlled through two distributed access control mechanisms, i.e., based on secret sharing (SS) and threshold proxy re-encryption (TPRE). The second one is an audit DLT that consists of a permissionless DLT that provides tamper-proof security to the states of the authorization DLT. Furthermore, we provided a security and privacy analysis based on a privacy impact assessment. We described the PIMS implementation where the authorization DLT was developed as an Ethereum private blockchain. At first, we compared the differences between SS and TPRE mechanisms; then, we analyzed the execution of a complete TPRE access control scenario experimentally in terms of execution time and system throughput. Results from our performance evaluation show that: (i) TPRE is faster when increasing the size of data to encrypt/decrypt and also more scalable; (ii) the implementation of the authorization DLT has shown that writing on the ledger represents a bottleneck but that in most use cases, the implementation is viable.

In the policy-based access control layer, we enrich the expressiveness of the access control mechanism to let the data subjects and/or holders express privacy policies enacted through smart contracts. We use a set of Semantic Web technologies and standards for this aim. First, we use a standard to specify access control policies over assets, i.e., the Moving Pictures Expert Group's (MPEG) ISO/IEC 21000 MPEG-21 framework and Smart Contract for Media standards. Second, we integrate those with the Data Privacy Vocabulary (DPV), i.e., a specification that specifies terms such as purposes for processing or legal basis. The implementation of the "Smart" Data Processing Agreement and different legal bases use cases led us to assess the value of the proposed access control model for an authorization based on a specific set of privacy policies. This privacy-policy-based access control layer aims to guarantee a series of features in favor of a transparent personal data access process. Indeed it allows us to enforce policies and trace the operations when used.

The last layer is the one that the final user needs to be identified during the execution of the decentralized systems. We present a Self-Sovereign Identity (SSI) model called Intelligible identity that we implemented as a set of technological components to provide, request, and obtain qualified data to negotiate and/or execute electronic transactions. It, in particular, follows the new proposal of modification to the eIDAS regulation to manage the qualified identity management based on DLT. Our model is a specialization of a W3C DID that adds the possibility to bring with it the relevant operational and legal context of this identity and to trace the processes that involve it effortlessly. The implementation based on the decentralized PIMS and the Akoma

Ntoso standard, together with the Public Sector Information and Personal Data Processing use case, led us to assess the value of an intelligible model for automatic claims verification based on a specific context, while maintaining intelligibility for the human interaction.

9.2 Limitations and future works

Some limitations of this work are presented in the following:

- DLTs still have scalability issues when real-time mass users ledger writing is needed.
- Decentralized indexing in DLTs and other decentralized systems still needs widespread approaches to enable complex queries. We had to implement the software by ourselves by the community might not support it.
- The proposed authorization mechanism is provided through a permissioned network. Such a service must be carefully crafted to abide by the law and guarantee the correct working to the final users. This also includes the use of "safe" and/or "legal" smart contracts, but the legal and technical debate on how to define a smart contract is still open.
- The policy-based access control layer has not been tested against the final user. The standardization bodies involved in their development vouched for the rich privacy policy expressiveness of the used languages. However, the policy creation possible complexity that the final user might face has yet to be tested out.
- The technical implementation of an online identity based on the Intelligible Identity paradigm might be influenced by the legal framework that is continuing to build up in these years.

Based on these limitations, we foresee some future works that our research will cover:

- Continuing studying the evolution of the EU legal framework with respect to online users' generated data and adapting/replacing parts of the proposed architecture to remain compliant.

- Evaluate the feasibility of proposing a newer system that can be modularized to face several data protection laws worldwide.

- Prototype the parts of the system that involve the use of a DLT by benefiting from the latest advancements in this field, e.g., by adopting the new features that IOTA DLT provide in their 2.0 version or new smart contract languages.

- Move the multi-party computation at the level of data instead of encryption keys. This means using algorithms that analyze and manipulate in a decentralized way only the encrypted version of the personal data, i.e., without decryption.

- Design of a technical assistant to provide information on personal data access and possible information inferences. Such software should act as a guide for the individual by giving a clear view of the data flowing through the processing chains; thus, it will act as an interface to the information stored in DLTs. Moreover, it will give an overview of possible information inferred from the analysis of personal data, acting as an attacker trying to violate his informational privacy. Finally, the software shall help users adjust their privacy policies to relieve cognitive overload.

Bibliography

Abdul, A., Vermeulen, J., Wang, D., Lim, B. Y., and Kankanhalli, M. (2018). Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–18. 211

Acquisti, A., Taylor, C., and Wagman, L. (2016). The economics of privacy. *Journal of economic Literature*, 54(2):442–92. 18, 19, 28, 29, 30

Agencia Española De Protección De Datos (2019). Introduction to the Hash Function as a Personal Data Pseudonymisation Technique. Technical report, Agencia Española De Protección De Datos. 43, 45, 69, 70, 78, 79, 134

Ahmad, A., Whitworth, B., Zeshan, F., Bertino, E., and Friedman, R. (2017). Extending social networks with delegation. *Computers & Security*, 70:546–564. 41, 173

Ahmed, J., Yildirim, S., Nowostawski, M., Abomhara, M., Ramachandra, R., and Elezaj, O. (2020). Towards blockchain-based gdpr-compliant online social networks: Challenges, opportunities and way forward. In *Future of Information and Communication Conference*, pages 113–129. Springer. 40, 68

AI HLEG (2019). Ethics guidelines for trustworthy ai. 211

Aiello, M., Cambiaso, E., Canonico, R., Maccari, L., Mellia, M., Pescapè, A., and Vaccari, I. (2020). Ippo: A privacy-aware architecture for decentralized data-sharing. *arXiv preprint arXiv:2001.06420*. 47, 68, 116

Ajao, O., Hong, J., and Liu, W. (2015). A survey of location inference techniques on twitter. *Journal of Information Science*, 41(6):855–864. 20

Ali, M., Shea, R., Nelson, J., and Freedman, M. J. (2017). Blockstack: A new decentralized internet. *Whitepaper, May*. 67, 118

Altshuler, Y., Aharony, N., Fire, M., Elovici, Y., and Pentland, A. (2012). Incremental learning with accuracy prediction of social and individual properties from mobile-phone data. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 969–974. IEEE. 18

Amoretti, M., Brambilla, G., Medioli, F., and Zanichelli, F. (2018). Blockchain-based proof of location. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 146–153. IEEE. 182

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, pages 1–15. 64, 65, 127

Article 29 Working Party (2014a). Opinion 05/2014 on anonymisation techniques. `https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/index_en.htm`. 45, 46, 47, 48, 70, 78

Article 29 Working Party (2014b). Opinion 06/2014 on the notion of legitimate interests of the data controller under article 7 of directive 95/46/ec. 23

Article 29 Working Party (2014c). Opinion 8/2014 on the on recent developments on the internet of things. `https://www.pdpjournals.com/docs/88440.pdf`. 166

Ateniese, G., Fu, K., Green, M., and Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30. 116

Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., and Wyner, A. (2013). Oasis legalruleml. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, pages 3–12. 237

Backx, P., Wauters, T., Dhoedt, B., and Demeester, P. (2002). A comparison of peer-to-peer architectures. In *Eurescom Summit*, volume 2. Citeseer. 31

Bagdasaryan, E., Berlstein, G., Waterman, J., Birrell, E., Foster, N., Schneider, F. B., and Estrin, D. (2019). Ancile: Enhancing privacy for ubiquitous computing with use-based privacy. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, pages 111–124. 172

Bakarich, K. and Castonguay, J. J. (2021). Using a permissioned blockchain? *The CPA Journal*, 91(6/7):48–51. 37

Bakos, Y. and Halaburda, H. (2021). Tradeoffs in permissioned vs permissionless blockchains: Trust and performance. *NYU Stern School of Business working paper*. 37

Bakos, Y., Halaburda, H., and Mueller-Bloch, C. (2021). When permissioned blockchains deliver more decentralization than permissionless. *Communications of the ACM*, 64(2):20–22. 31, 33, 37

Bamakan, S. M. H., Nezhadsistani, N., Bodaghi, O., and Qu, Q. (2021). A decentralized framework for patents and intellectual property as NFT in blockchain networks. 182

Banerjee, A. and Joshi, K. P. (2017). Link before you share: Managing privacy policies through blockchain. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 4438–4447. IEEE. 173

Banerjee, S. (2019). Geosurveillance, location privacy, and personalization. *Journal of Public Policy & Marketing*, 38(4):484–499. 18, 30

Beck, R., Czepluch, J. S., Lollike, N., and Malone, S. (2016). Blockchain–the gateway to trust-free cryptographic transactions. In *Twenty-Fourth European Conference on Information Systems (ECIS), İstanbul, Turkey, 2016*, pages 1–14. Springer Publishing Company. 36

Becker, M. and Bodó, B. (2021). Trust in blockchain-based systems. *Internet Policy Review*, 10(2):1–10. 35, 36

Bellotti, V. and Edwards, K. (2001). Intelligibility and accountability: human considerations in context-aware systems. *Human–Computer Interaction*, 16(2-4):193–212. 211

Belotti, M., Božić, N., Pujolle, G., and Secci, S. (2019). A vademecum on blockchain technologies: When, which, and how. *IEEE Communications Surveys & Tutorials*, 21(4):3796–3838. 92

Bench-Capon, T. J. and Coenen, F. P. (1992). Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law*, 1(1):65–86. 179

Benet, J. (2014). Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*. 47, 64, 66, 95, 96, 190

Benet, J. and Greco, N. (2018). Filecoin: A decentralized storage network. *Protoc. Labs*. 66

Beniiche, A. (2020). A study of blockchain oracles. *arXiv preprint arXiv:2004.07140*. 206

Berners-Lee, T., Fielding, R., and Masinter, L. (2004). Uniform resource identifier (uri): Generic syntax. Technical report. 187, 227

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37. 169

Bez, M., Fornari, G., and Vardanega, T. (2019). The scalability challenge of ethereum: An initial quantitative analysis. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 167–176. IEEE. 87, 133, 134

Bigini, G. and Lattanzi, E. (2022). Toward the interplanetary health layer for the internet of medical things with distributed ledgers and storages. *IEEE Access*, 10:82883–82895. 127

Biryukov, A. and Pustogarov, I. (2015). Bitcoin over tor isn't a good idea. In *2015 IEEE Symposium on Security and Privacy*, pages 122–134. IEEE. 34

Blakley, G. R. (1979). Safeguarding cryptographic keys. In *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318. IEEE. 116, 126

Blockchain.com (2020). Blockchain explorer. 93

Bonneau, J., Anderson, J., Anderson, R., and Stajano, F. (2009). Eight friends are enough: social graph approximation via public listings. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 13–18. 19

Bothorel, C., Lathia, N., Picot-Clemente, R., and Noulas, A. (2018). Location recommendation with social media data. In *Social Information Access*, pages 624–653. Springer. 19

Bradbury, D. (2012). Digital certificates: worth the paper they're written on? *Computer Fraud & Security*, 2012(10):12–16. 213

Brogan, J., Baskaran, I., and Ramachandran, N. (2018). Authenticating health activity data using distributed ledger technologies. *Computational and Structural Biotechnology Journal*, 16. 66, 82

Burnett, I., Van de Walle, R., Hill, K., Bormans, J., and Pereira, F. (2003). Mpeg-21: goals and achievements. *IEEE MultiMedia*, 10(4):60–70. 171

Buterin, V. et al. (2013). Ethereum white paper. 64, 65, 115, 121, 126, 127, 169, 222

Cadwalladr, C. and Graham-Harrison, E. (2018). Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. *The guardian*, 17:22. 21

Caglayan, P. and Ozkan, Z. (2021). NFTs and copyright: challenges and opportunities. *Journal of Intellectual Property Law& Practice*, 16(10). 169

Cai, T., Yang, Z., Chen, W., Zheng, Z., and Yu, Y. (2020). A blockchain-assisted trust access authentication system for solid. *IEEE Access*, 8:71605–71616. 173

California State Legislature (2020). California consumer privacy act. 25

Cameron, K. (2005). The laws of identity. *https://docs.microsoft.com/en-us/previous-versions/ dotnet/articles/ms996456(v=msdn.10).* Online. Accessed 2021. 214

Campanile, L., Cantiello, P., Iacono, M., Marulli, F., and Mastroianni, M. (2021). Risk analysis of a gdpr-compliant deletion technique for consortium blockchains based on pseudonymization. In *International Conference on Computational Science and Its Applications*, pages 3–14. Springer. 138

Campbell, M. J., Dennison, P. E., Butler, B. W., and Page, W. G. (2019). Using crowdsourced fitness tracker data to model the relationship between slope and travel rates. *Applied geography*, 106:93–107. 199

Cavoukian, A. (2009). Privacy by design. *Take the challenge. Information and privacy commissioner of Ontario, Canada*. 68

Cervone, L., Palmirani, M., and Vitali, F. (2020). The intelligible contract. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*. 41, 213, 221, 237

Chang, E. Y., Liao, S.-W., Liu, C.-T., Lin, W.-C., Liao, P.-W., Fu, W.-K., Mei, C.-H., and Chang, E. J. (2018). Deeplinq: distributed multi-layer ledgers for privacy-preserving data sharing. In *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 173–178. IEEE. 117, 118, 133

Cheng, J.-C., Lee, N.-Y., Chi, C., and Chen, Y.-H. (2018). Blockchain and smart contract for digital certificate. In *2018 IEEE international conference on applied system invention (ICASI)*, pages 1046–1051. IEEE. 213

Chhetri, T. R., Kurteva, A., DeLong, R. J., Hilscher, R., Korte, K., and Fensel, A. (2022). Data protection by design tool for automated gdpr compliance verification based on semantically modeled informed consent. *Sensors*, 22(7):2763. 172

Cho, E., Myers, S. A., and Leskovec, J. (2011). Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. 19

Choudhury, O., Rudolph, N., Sylla, I., Fairoza, N., and Das, A. (2018). Auto-generation of smart contracts from domain-specific ontologies and semantic rules. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 963–970. IEEE. 41

Christensen, S. (2018). A comparative study of privacy-preserving cryptocurrencies: Monero and zcash. Master's thesis, University of Birmingham. 47, 79

Christl, W., Kopp, K., and Riechert, P. U. (2017). How companies use personal data against people. *Automated Disadvantage, Personalized Persuasion, and the Societal Ramifications of the Commercial Use of Personal Information. Wien: Cracked Labs*. 21

Christopher Allen (2016). The path to self-sovereign identity. 5, 41, 214

Ciani, J. (2018). Governing data trade in intelligent environments: A taxonomy of possible regulatory regimes between property and access rights. In *Intelligent Environments 2018, Workshop Proceedings of the 14th International Conference on Intelligent Environments, Ambient Intelligence and Smart Environments Series*, volume 23, pages 285–297. 184

CNIL (2018a). Commission nationale de l'informatique et des libertés - solutions for a responsible use of the blockchain in the context of personal data. 45, 46, 47, 48, 49, 69, 71, 79, 80, 137

CNIL (2018b). French Data Protection Authority - Privacy Impact Assessment (PIA) - Knowledge Bases (2018. Technical report, French Data Protection Authority (CNIL). 138, 139, 140

Conti, M. and Passarella, A. (2018). The internet of people: A human and data-centric paradigm for the next generation internet. *Computer Communications*, 131:51–65. 5

Corcho, O., Jiménez, J., Morote, C., and Simperl, E. (2022). Data.europa.eu and citizen-generated data. 199

Courtois, N. T. and Mercer, R. (2017). Stealth address and key management techniques in blockchain systems. *ICISSP*, 2017:559–566. 47, 67, 79

Custers, B., van Der Hof, S., Schermer, B., Appleby-Arnold, S., and Brockdorff, N. (2013). Informed consent in social media use-the gap between user expectations and eu personal data protection law. *SCRIPTed*, 10:435. 27

Damgård, I. B. (1989). A design principle for hash functions. In *Conference on the Theory and Application of Cryptology*, pages 416–427. Springer. 224, 227

D'Angelo, G. and Ferretti, S. (2011). Lunes: Agent-based simulation of p2p systems. In *2011 International Conference on High Performance Computing & Simulation*, pages 593–599. IEEE. 102

D'Angelo, G., Ferretti, S., and Marzolla, M. (2018). A blockchain-based flight data recorder for cloud accountability. In *Proc. of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock)*. 142

Daudén-Esmel, C., Castellà-Roca, J., Viejo, A., and Domingo-Ferrer, J. (2021). Lightweight blockchain-based platform for gdpr-compliant personal data management. In *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)*, pages 68–73. IEEE. 173

Davari, M. and Bertino, E. (2019). Access control model extensions to support data privacy protection based on gdpr. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4017–4024. IEEE. 41, 173

De Filippi, P., Mannan, M., and Reijers, W. (2020). Blockchain as a confidence machine: The problem of trust & challenges of governance. *Technology in Society*, 62:101284. 30, 32, 36

De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., and Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376. 19

Di Cerbo, F., Martinelli, F., Matteucci, I., and Mori, P. (2018). Towards a declarative approach to stateful and stateless usage control for data protection. In *WEBIST*, pages 308–315. 172

Dias, D. and Costa, L. H. M. K. (2018). CRAWDAD dataset coppe-ufrj/riobuses (v. 2018-03-19). Downloaded from `https://crawdad.org/coppe-ufrj/RioBuses/20180319`. 80

Dobson, J. E. and Fisher, P. F. (2003). Geoslavery. *IEEE Technology and Society Magazine*, 22(1):47–52. 21

D'Angelo, G. and Ferretti, S. (2017). Highly intensive data dissemination in complex networks. *Journal of Parallel and Distributed Computing*, 99:28–50. 95

EDPB, E. D. P. S. (2016). Opinion on personal information management systems. 4, 39, 62

Edwards, L., Finck, M., Veale, M., and Zingales, N. (2019). Data subjects as data controllers: a fashion (able) concept? *Internet Policy Review*. 48

Egorov, M., Wilkison, M., and Nuñez, D. (2017). Nucypher kms: decentralized key management system. *arXiv preprint arXiv:1707.06140*. 118, 126, 146

Eiband, M., Schneider, H., Bilandzic, M., Fazekas-Con, J., Haug, M., and Hussmann, H. (2018). Bringing transparency design into practice. In *23rd international conference on intelligent user interfaces*, pages 211–223. 211

Elsts, A., Mitskas, E., and Oikonomou, G. (2018). Distributed ledger technology and the Internet of Things: A feasibility study. In *Proc. of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (BlockSys)*. 82

ENISA, E. U. A. F. C. (2017). Guidelines for SMEs on the security of personal data processing. Technical report, European Union Agency for Cybersecurity. 70

ENISA, E. U. A. F. C. (2021). Data Pseudonymisation: Advanced Techniques & Use Cases. Technical report, European Union Agency for Cybersecurity. 4, 62, 70, 79, 126

Entriken, W., Shirley, D., Evans, J., and Sachs, N. (2018). Erc-721 non-fungible token standard. 169, 233

Erbguth, J. (2019a). Five ways to gdpr-compliant use of blockchains. *Eur. Data Prot. L. Rev.*, 5:427. 45, 46

Erbguth, J. (2019b). Practitioner's Corner - Five Ways to GDPR-Compliant Use of Blockchains. *European Data Protection Law Review*, 5(3):427–433. 69

Esteves, B., Pandit, H. J., and Rodríguez-Doncel, V. (2021a). Odrl profile for expressing consent through granular access control policies in solid. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 298–306. IEEE. 172, 184

Esteves, B., Pandit, H. J., and Rodríguez-Doncel, V. (2021b). Odrl profile for expressing consent through granular access control policies in solid. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pages 298–306. 159

Esteves, B. and Rodríguez-Doncel, V. (2022). Analysis of ontologies and policy languages to represent information flows in gdpr. *Semantic Web*, (Preprint):1–35. 170, 171

European Blockchain Partnership EBP (2022). European blockchain services infrastructure. `https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home`. 38, 211

European Commission (2020). A european strategy for data. 4, 5, 39, 62, 68, 112

European Commission (2021). On the evaluation of regulation (eu) no 910/2014 on electronic identification and trust services for electronic transactions in the internal market (eidas). 210

European Commission (2022). Data act proposal. 113

European Data Protection Board (2020). Guidelines 05/2020 on consent under regulation 2016/679. https://edpb.europa.eu/sites/default/files/files/file1/edpb_guidelines_202005_consent_en.pdf. 167, 172

European Parliament (2002). Privacy and electronic communications directive 2002/58/ec. 21

European Parliament (2014). Regulation (eu) 2014/910 - directive 95/46. 210

European Parliament (2016). Regulation (eu) 2016/679. 4, 21

European Parliament (2017). European parliament resolution of 3 october 2018 on distributed ledger technologies and blockchains: building trust with disintermediation. 5, 6, 63

European Parliament (2019). Directive eu 2019/1024. 235

European Parliament (2022). Regulation (eu) 2022/868. 62, 63, 72, 137

Fairfield, J. (2021). Tokenized: The Law of Non-Fungible Tokens and Unique Digital Property. *Indiana Law Journal, Forthcoming*. 169

Farshid, S., Reitz, A., and Roßbach, P. (2019). Design of a forgetting blockchain: A possible way to accomplish gdpr compatibility. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*. 46

Feige, U., Fiat, A., and Shamir, A. (1988). Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2). 45, 217

Finck, M. (2019). *Blockchain and the General Data Protection Regulation: Can Distributed Ledgers be Squared with European Data Protection Law?: Study*. European Parliament. 43, 45, 46, 47, 49, 69, 78, 79, 137

Finck, M. and Pallas, F. (2020). They who must not be identified—distinguishing personal from non-personal data under the GDPR. *International Data Privacy Law*, 10(1):11–36. 43, 45, 46, 47, 48, 69

Finocchiaro, G. and Bomprezzi, C. (2020). Legal analysis of the use of blockchain technology for the formation of smart legal contracts. *Media Laws Rivista di Diritto dei Media*. 32, 182

Fishbein, M. and Ajzen, I. (1977). Belief, attitude, intention, and behavior: An introduction to theory and research. *Philosophy and Rhetoric*, 10(2). 29

Florian, M., Henningsen, S., Beaucamp, S., and Scheuermann, B. (2019). Erasing data from blockchain nodes. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 367–376. IEEE. 46

Floridi, L. (2014). *The fourth revolution: How the infosphere is reshaping human reality*. OUP Oxford. 5, 17, 18, 20, 41, 209

Forbrukerrådet (2020). Out of control – how consumers are exploited by the online advertising industry. 19, 21

Formez PA (2020). Italian open data license v2.0. *https://www.dati.gov.it/content/italian-open-data -license-v20*. Online. Accessed 2021. 235, 237

Furini, M., Mirri, S., Montangero, M., and Prandi, C. (2020). Privacy perception when using smartphone applications. *Mobile Networks and Applications*, 25:1055–1061. 62

Gaaloul, K., Schaad, A., Flegel, U., and Charoy, F. (2008). A secure task delegation model for workflows. In *2008 Second International Conference on Emerging Security Information, Systems and Technologies*, pages 10–15. IEEE. 171, 173

Gal, M. S. and Aviv, O. (2020). The competitive effects of the gdpr. *Journal of Competition Law & Economics*, 16(3):349–391. 25, 26

Gen, K., Akira, N., Makoto, M., Yasuhiro, O., Tomohiro, O., and Katsuhiko, T. (2015). Applying the akoma ntoso xml schema to japanese legislation. *JL Inf. & Sci.*, 24:49. 227

Giannopoulou, A. (2020). Data protection compliance challenges for self-sovereign identity. In Prieto, J., Pinto, A., Das, A. K., and Ferretti, S., editors, *Blockchain and Applications*, pages 91–100, Cham. Springer International Publishing. 5, 6, 41, 45, 47, 48, 63, 69, 71

Giannopoulou, A. and Wang, F. (2021). Self-sovereign identity. *Internet Policy Review*, 10(2):1–10. 5

Giansante, C. and Zichichi, M. (2021). Hypercube DHT Simulation. DOI: `10.5281/zenodo.6548266`. 102

Gräther, W., Kolvenbach, S., Ruland, R., Schütte, J., Torres, C., and Wendland, F. (2018). Blockchain for education: lifelong learning passport. In *Proceedings of 1st ERCIM Blockchain workshop 2018*. European Society for Socially Embedded Technologies (EUSSET). 213

Grinberg, M. (2018). *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.". 106

Hardjono, T., Shrier, D. L., and Pentland, A. (2019). *2 TOWARDS AN INTERNET OF TRUSTED DATA*, pages 15–40. MIT Press. 37

Hasan, S., Ukkusuri, S. V., and Zhan, X. (2016). Understanding social influence in activity location choice and lifestyle patterns using geolocation data from social media. *Frontiers in ICT*, 3:10. 19

Hassanzadeh-Nazarabadi, Y., Taheri-Boshrooyeh, S., Otoum, S., Ucar, S., and Özkasap, Ö. (2021). Dht-based communications survey: Architectures and use cases. *arXiv preprint arXiv:2109.10787*. 116

Hawig, D., Zhou, C., Fuhrhop, S., Fialho, A. S., and Ramachandran, N. (2019). Designing a distributed ledger technology system for interoperable and general data protection regulation–compliant health data exchange: a use case in blood glucose data. *Journal of medical Internet research*, 21(6):e13665. 40, 67, 117, 173

He, R., Cao, J., Zhang, L., and Lee, D. (2019). Statistical enrichment models for activity inference from imprecise location data. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 946–954. IEEE. 19

Herranz, J., Hofheinz, D., and Kiltz, E. (2006). Kem/dem: Necessary and sufficient conditions for secure hybrid encryption. *IACR Cryptology ePrint Archive*. 75

High-Level Expert Group on Business-to-Government Data Sharing (2021). Towards a european strategy on business-to-government data sharing for the public interest. 199

Hur, J. and Noh, D. K. (2010). Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1214–1221. 119

Ianella, R. (2007). Open digital rights language (odrl). *Open Content Licensing: Cultivating the Creative Commons*. 170, 171

Iannella, R. and McKinney, J. (2014). vcard ontology-for describing people and organizations. *W3C Group Note NOTE-vcard-rdf-20140522*. 170

ICO (2019). Project explain interim report. *https://ico.org.uk/about-the-ico/research-and-reports/ project-explain-interim-report/*. 211

Infura Inc (2020). Infura: Secure and scalable access to ethereum apis and ipfs gateways. 81

IPFS Community (2021). Search engine for the interplanetary file system. `https://github.com/ipfs-search/ipfs-search`. 95

IPLD Team (2016). Interplanetary linked data (ipld). *https://specs.ipld.io/*. 92

ISO/IEC IS 21000-23 (2022). ISO/IEC IS 21000-23 Information technology - Multimedia framework (MPEG-21) - Part 23: Smart Contracts for Media. Standard, International Organization for Standardization, Geneva, CH. 171, 175, 178, 179, 189, 205

Jacobson, V. (1988). Congestion avoidance and control. In *ACM SIGCOMM Computer Communication Review*, volume 18. ACM. 82

Janssen, H., Cobbe, J., Norval, C., and Singh, J. (2020). Decentralized data processing: personal data stores and the gdpr. *International Data Privacy Law*, 10(4):356–384. 4

Janssen, H. and Singh, J. (2022a). The data intermediary. *Internet Policy Review*, 11(2):1–6. 62

Janssen, H. and Singh, J. (2022b). Personal information management systems. *Internet Policy Review*, 11(2):1–6. 4

Jemel, M. and Serhrouchni, A. (2017). Decentralized access control mechanism with temporal dimension based on blockchain. In *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, pages 177–182. IEEE. 115, 116, 119

Jiang, P., Guo, F., Liang, K., Lai, J., and Wen, Q. (2020). Searchain: Blockchain-based private keyword search in decentralized storage. *Future Generation Computer Systems*, 107:781–792. 96

Jiang, S., Cao, J., Wu, H., Yang, Y., Ma, M., and He, J. (2018). Blochie: a blockchain-based platform for healthcare information exchange. In *2018 ieee international conference on smart computing (smartcomp)*, pages 49–56. IEEE. 117

Jiang, S., Liu, J., Wang, L., and Yoo, S.-M. (2019). Verifiable search meets blockchain: A privacy-preserving framework for outsourced encrypted data. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE. 67, 118

Joung, Y.-J., Yang, L.-W., and Fang, C.-T. (2007). Keyword search in dht-based peer-to-peer networks. *IEEE Journal on Selected Areas in Communications*, 25(1):46–61. 93, 94, 99, 100

Jurgens, D. (2013). That's what friends are for: Inferring location in online social media platforms based on social relationships. In *Seventh International AAAI Conference on Weblogs and Social Media*. 19

Kamleitner, B. and Mitchell, V. (2019). Your data is my data: A framework for addressing interdependent privacy infringements. *Journal of Public Policy & Marketing*, 38(4):433–450. 19, 20

Kaplan, A. M. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68. 17

Katevas, K., Bagdasaryan, E., Waterman, J., Safadieh, M. M., Haddadi, H., and Estrin, D. (2020). Decentralized policy-based private analytics. *arXiv preprint arXiv:2003.06612*. 41, 172

Kavun, E. B. and Yalcin, T. (2010). A lightweight implementation of keccak hash function for radio-frequency identification applications. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pages 258–269. Springer. 228

Kayem, A. V., Akl, S. G., and Martin, P. (2010). *Adaptive cryptographic access control*, volume 48. Springer Science & Business Media. 115

Keßler, C. and McKenzie, G. (2018). A geoprivacy manifesto. *Transactions in GIS*, 22(1):3–19. 20, 237

Khelifi, H., Luo, S., Nour, B., Moungla, H., and Ahmed, S. H. (2018). Reputation-based blockchain for secure NDN caching in vehicular networks. In *Proc. of Conference on Standards for Communications and Networking (CSCN)*. IEEE. 40

Khudhur, N. and Fujita, S. (2019). Siva-the ipfs search engine. In *2019 Seventh International Symposium on Computing and Networking (CANDAR)*, pages 150–156. IEEE. 95

Kirrane, S., Villata, S., and d'Aquin, M. (2018). Privacy, security and policies: A review of problems and solutions with semantic web technologies. *Semantic Web*, 9(2):153–161. 170, 171

Kokott, J. and Sobotta, C. (2013). The distinction between privacy and data protection in the jurisprudence of the cjeu and the ecthr. *International Data Privacy Law*, 3(4):222–228. 26

Kondova, G. and Erbguth, J. (2020). Self-sovereign identity on public blockchains and the gdpr. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 342–345. 30, 41, 69, 214

Kongruangkit, S., Xia, Y., Xu, X., and Paik, H.-y. (2021). A case for connecting solid and blockchains: Enforcement of transparent access rights in personal data stores. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–5. IEEE. 173

Koscina, M., Manset, D., Negri, C., and Perez, O. (2019). Enabling trust in healthcare data exchange with a federated blockchain-based architecture. In *IEEE/WIC/ACM International Conference on Web Intelligence-Companion Volume*, pages 231–237. 40, 117, 173

Kudumakis, P., Wilmering, T., Sandler, M., Rodríguez-Doncel, V., Boch, L., and Delgado, J. (2020). The challenge: from mpeg intellectual property rights ontologies to smart contracts and blockchains [standards in a nutshell]. *IEEE Signal Processing Magazine*, 37(2):89–95. 167, 171, 175, 189

Kwon, Y., Liu, J., Kim, M., Song, D., and Kim, Y. (2019). Impossibility of full decentralization in permissionless blockchains. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 110–123. 34

La Piana, F., Leurini, A., Tropea, D., and Zichichi, M. (2021). Hypercube dht implementation ans vehicular scenario. DOI: `10.5281/zenodo.5810396`. 106

Laufer, R. S. and Wolfe, M. (1977). Privacy as a concept and a social issue: A multidimensional developmental theory. *Journal of social Issues*, 33(3):22–42. 29

Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., and Zhao, J. (2013). Prov-o: The prov ontology. *W3C recommendation*, 30. 173

Li, H., Yu, L., and He, W. (2019). The impact of gdpr on global technology development. 25

Li, L., Liu, J., Cheng, L., Qiu, S., Wang, W., Zhang, X., and Zhang, Z. (2018). Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2204–2220. 65

Longley, D., Allen, C., Sporny, M., Sabadello, M., and Reed, D. (2021). Decentralized identifiers (DIDs) v1.0. Candidate recommendation, W3C. https://www.w3.org/TR/2021/CRD-did-core-20210529/. 210

Longley, D., Noble, G., Burnett, D., Zundel, B., and Sporny, M. (2019). Verifiable credentials data model 1.0. W3C recommendation, W3C. https://www.w3.org/TR/2019/REC-vc-data-model-20191119/. 210, 216

Lopez, D. and Farooq, B. (2020). A multi-layered blockchain framework for smart mobility data-markets. *Transportation Research Part C: Emerging Technologies*, 111:588–615. 30, 67

Lopez, P. G., Montresor, A., and Datta, A. (2019). Please, do not decentralize the internet with (permissionless) blockchains! In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1901–1911. IEEE. 30, 31, 35, 37, 38

Lundkvist, C., Heck, R., Torstensson, J., Mitton, Z., and Sena, M. (2017). Uport: A platform for self-sovereign identity. *URL: https://whitepaper. uport. me/uPort_ whitepaper_DRAFT20170221. pdf*. 128, 214

Lyons, T., Courcelas, L., and Timsit, K. (2018). Blockchain and the gdpr. In *The European Union Blockchain Observatory and Forum*. 43, 44, 46, 48, 69, 70, 77, 78, 133, 137

Ma, C. Y., Yau, D. K., Yip, N. K., and Rao, N. S. (2010). Privacy vulnerability of published anonymous mobility traces. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 185–196. 19

Maesa, D. D. F., Mori, P., and Ricci, L. (2019). A blockchain based approach for the definition of auditable access control systems. *Computers & Security*, 84:93–119. 6, 63, 64, 115, 116

Mahindrakar, A. and Joshi, K. P. (2020). Automating gdpr compliance using policy integrated blockchain. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 86–93. IEEE. 173

Markus, A. F., Kors, J. A., and Rijnbeek, P. R. (2020). The role of explainability in creating trustworthy artificial intelligence for health care: a comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, page 103655. 212

Mazzoni, M., Corradi, A., and Di Nicola, V. (2022). Performance evaluation of permissioned blockchains for financial applications: The consensys quorum case study. *Blockchain: Research and applications*, 3(1):100026. 146, 154

Meessen, P., Venema, M., Sonnino, A., and Bano, S. (2018). D3.8 decentralised models for data and identity management: Blockchain and abc mvps. *Decode H2020, Decode consortium, DECODE Project, Tech. Rep. H2020-ICT-2016-1*. 119

Mehrnezhad, M. (2020). A cross-platform evaluation of privacy notices and tracking practices. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 97–106. IEEE. 27

Mellet, K. and Beauvisage, T. (2020). Cookie monsters. anatomy of a digital market infrastructure. *Consumption Markets & Culture*, 23(2):110–129. 27

Metamask (2021). Metamask website. *https://metamask.io/*. Online. Accessed 2021. 233

Mockus, M. (2017). *Open government data licensing framework: an informal ontology for supporting mashup*. PhD thesis, Mykolas Romeris University. 234, 237

Molina, F., Betarte, G., and Luna, C. (2020). A blockchain based and gdpr-compliant design of a system for digital education certificates. *arXiv preprint arXiv:2010.12980*. 69, 173

Montresor, A. and Jelasity, M. (2009). Peersim: A scalable p2p simulator. In *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, pages 99–100. IEEE. 102

Mougayar, W. (2016). *The business blockchain: promise, practice, and application of the next Internet technology*. John Wiley & Sons. 65

Naik, N. and Jenkins, P. (2020). uport open-source identity management system: An assessment of self-sovereign identity and user-centric data platform built on blockchain. In *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–7. IEEE. 42

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. 31, 32, 46, 64, 65, 121, 127

Naz, M., Al-zahrani, F. A., Khalid, R., Javaid, N., Qamar, A. M., Afzal, M. K., and Shafiq, M. (2019). A secure data sharing platform using blockchain and interplanetary file system. *Sustainability*, 11(24):7054. 40, 67

Norberg, P. A., Horne, D. R., and Horne, D. A. (2007). The privacy paradox: Personal information disclosure intentions versus behaviors. *Journal of consumer affairs*, 41(1):100–126. 29

Nunez, D. (2018). Umbral: A threshold proxy re-encryption scheme. 132

Olteanu, A.-M., Huguenin, K., Shokri, R., and Hubaux, J.-P. (2014). Quantifying the effect of co-location information on location privacy. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 184–203. Springer. 20

Onik, M. M. H., Kim, C.-S., Lee, N.-Y., and Yang, J. (2019). Privacy-aware blockchain for personal data sharing and tracking. *Open Computer Science*, 9(1):80–91. 40, 47, 68, 118

OpenEthereum (2020). OperEthereum Secret Store. 146

Ortega, V., Bouchmal, F., and Monserrat, J. F. (2018). Trusted 5G vehicular networks: Blockchains and content-centric networking. *IEEE Vehicular Technology Magazine*, 13(2). 40

Özyilmaz, K. R., Doğan, M., and Yurdakul, A. (2018). IDMoB: IoT data marketplace on blockchain. In *Proc. of the Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE. 40

Palm, E. (2017). Implications and impact of blockchain transaction pruning. 46, 138

Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., and Robaldo, L. (2018a). Pronto: Privacy ontology for legal reasoning. In *International Conference on Electronic Government and the Information Systems Perspective*, pages 139–152. Springer. 170

Palmirani, M., Sperberg, R., Vergottini, G., and Vitali, F. (2018b). Akoma ntoso version 1.0 part 1: Xml vocabulary. OASIS standard. 226, 227

Palmirani, M. and Vitali, F. (2011). Akoma-ntoso for legal documents. In *Legislative XML for the semantic Web*, pages 75–100. Springer. 212, 226, 227

Pandit, H. J., Debruyne, C., O'Sullivan, D., and Lewis, D. (2019a). Gconsent-a consent ontology based on the gdpr. In *European Semantic Web Conference*, pages 270–282. Springer. 170

Pandit, H. J., O'Sullivan, D., and Lewis, D. (2018). An ontology design pattern for describing personal data in privacy policies. In *WOP at ISWC*, pages 29–39. 170

Pandit, H. J., Polleres, A., Bos, B., Brennan, R., Bruegger, B., Ekaputra, F. J., Fernández, J. D., Hamed, R. G., Kiesling, E., Lizar, M., et al. (2019b). Creating a vocabulary for data privacy. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 714–730. Springer. 167, 184

Pangrazio, L. and Selwyn, N. (2019). 'personal data literacies': A critical literacies approach to enhancing understandings of personal digital data. *New Media & Society*, 21(2):419–437. 26

Pariser, E. (2011). *The filter bubble: What the Internet is hiding from you*. Penguin UK. 3, 21

Partridge, K. and Price, B. (2009). Enhancing mobile recommender systems with activity inference. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 307–318. Springer. 19

Pellegrini, T., Schönhofer, A., Kirrane, S., Steyskal, S., Fensel, A., Panasiuk, O., Mireles-Chavez, V., Thurner, T., Dörfler, M., and Polleres, A. (2018). A genealogy and classification of rights expression languages-preliminary results. In *Data Protection/LegalTech-Proceedings of the 21st International Legal Informatics Symposium IRIS*, pages 243–250. 170

Phithakkitnukoon, S., Horanont, T., Di Lorenzo, G., Shibasaki, R., and Ratti, C. (2010). Activity-aware map: Identifying human daily activity pattern using mobile phone data. In *International Workshop on Human Behavior Understanding*, pages 14–25. Springer. 19

Politou, E., Alepis, E., Patsakis, C., Casino, F., and Alazab, M. (2020). Delegated content erasure in IPFS. *Future Generation Computer Systems*, 112:956–964. 6, 47, 48, 63, 79

Politou, E., Casino, F., Alepis, E., and Patsakis, C. (2019). Blockchain Mutability: Challenges and Proposed Solutions. *IEEE Transactions on Emerging Topics in Computing*, PP(99):1–1. 46, 65, 79, 137

Pontes, T., Magno, G., Vasconcelos, M., Gupta, A., Almeida, J., Kumaraguru, P., and Almeida, V. (2012). Beware of what you share: Inferring home location in social networks. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 571–578. IEEE. 19

Popov, S. (2016). The Tangle. 64, 65

Preukschat, A. and Reed, D. (2021). *Self-Sovereign Identity: Decentralized digital identity and verifiable credentials*. Simon and Schuster. 210, 213, 214, 221

Qian, J., Li, X.-Y., Zhang, C., and Chen, L. (2016). De-anonymizing social networks and inferring private attributes using knowledge graphs. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE. 19

Ramachandran, G. S., Radhakrishnan, R., and Krishnamachari, B. (2018). Towards a decentralized data marketplace for smart cities. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–8. 67

Ramachandran, M., Chowdhury, N., Third, A., Jan, Z., Valentine, C., and Domingue, J. (2020). A framework for handling internet of things data with confidentiality and blockchain support. 173

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. (2001). A scalable content-addressable network. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172. 95

Rieger, A., Guggenmos, F., Lockl, J., Fridgen, G., and Urbach, N. (2019a). Building a blockchain application that complies with the eu general data protection regulation. *MIS Quarterly Executive*, 18(4):263–279. 48, 69

Rieger, A., Guggenmos, F., Lockl, J., Fridgen, G., and Urbach, N. (2019b). Building a Blockchain Application that Complies with the EU General Data Protection Regulation. *MIS Quarterly Executive*, 18(4):263–279. 69, 80, 137

Rinckes, D. and Bunge, P. (2015). Open location code: An open source standard for addresses independent of building numbers and street names. *Github. com. https://github. com/google/open-locationcode/blob/master/docs/olc definition. adoc (accessed Dic 2021)*. 106

Robaldo, L., Bartolini, C., and Lenzini, G. (2020). The dapreco knowledge base: representing the gdpr in legalruleml. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5688–5697. 170

Rodríguez, E., Delgado, J., Boch, L., and Rodríguez-Doncel, V. (2014). Media contract formalization using a standardized contract expression language. *IEEE multimedia*, 22(2):64–74. 175

Rodriguez-Doncel, V. and Delgado, J. (2009). A media value chain ontology for mpeg-21. *IEEE MultiMedia*, 16(4):44–51. 174

Rodríguez-Doncel, V., Delgado, J., Llorente, S., Rodríguez, E., and Boch, L. (2016). Overview of the mpeg-21 media contract ontology. *Semantic Web*, 7(3):311–332. 170, 175

Rodríguez, E., Delgado, J., Boch, L., and Rodríguez-Doncel, V. (2015). Media contract formalization using a standardized contract expression language. *IEEE MultiMedia*, 22(2):64–74. 175

Roio, D. (2019). The zencode whitepaper: Secure crypto language. 40, 226

Rossi, A. and Palmirani, M. (2020). What's in an icon? promises and pitfalls of data protection iconography. *Data Protection and Privacy: Data Protection and Democracy*, pages 59–92. 28

Rouhani, S. and Deters, R. (2019). Blockchain based access control systems: State of the art and challenges. In *IEEE/WIC/ACM International Conference on Web Intelligence*, WI '19, page 423–428, New York, NY, USA. Association for Computing Machinery. 116, 119

Sadilek, A., Kautz, H., and Bigham, J. P. (2012). Finding your friends and following them to where you are. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 723–732. 19

Samarati, P. and Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, technical report, SRI International. 200, 202

Sambra, A. V., Mansour, E., Hawke, S., Zereba, M., Greco, N., Ghanem, A., Zagidulin, D., Aboulnaga, A., and Berners-Lee, T. (2016). Solid: a platform for decentralized social applications based on linked data. Technical report, Technical report, MIT CSAIL & Qatar Computing Research Institute. 41, 42, 172

Santos, J., Santos, N., and Dias, D. (2019). Dclaims: A censorship resistant web annotations system using ipfs and ethereum. *arXiv preprint arXiv:1912.03388*. 93

Schneier, B. (2019). Blockchain and trust. `https://www.schneier.com/blog/archives/2019/02/blockchain_and_.html`. 32, 33

Schneier, B. (2022). On the dangers of cryptocurrencies and the uselessness of blockchain. `https://www.schneier.com/blog/archives/2022/06/on-the-dangers-of-cryptocurrencies-and-the-uselessness-of-blockchain.html`. 32

Serena, L., D'Angelo, G., and Ferretti, S. (2020). Implications of dissemination strategies on the security of distributed ledgers. In *Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pages 65–70. 31

Serena, L., Zichichi, M., D'Angelo, G., and Ferretti, S. (2021). Simulation of dissemination strategies on temporal networks. In *Proc. of the 2021 Annual Modeling and Simulation Conference (ANNSIM)*, pages 1–12. Society for Modeling and Simulation International (SCS). 154

Shafagh, H., Burkhalter, L., Duquennoy, S., Hithnawi, A., and Ratnasamy, S. (2018). Droplet: Decentralized authorization for iot data streams. *arXiv preprint arXiv:1806.02057*. 40, 67, 75, 118

Shafagh, H., Burkhalter, L., Hithnawi, A., and Duquennoy, S. (2017). Towards blockchain-based auditable storage and sharing of IoT data. In *Proc. of the Cloud Computing Security Workshop*. ACM. 40

Shahid, A. R., Pissinou, N., Njilla, L., Alemany, S., Imteaj, A., Makki, K., and Aguilar, E. (2019). Quantifying location privacy in permissioned blockchain-based internet of things (iot). In *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '19, page 116–125, New York, NY, USA. Association for Computing Machinery. 65

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613. 116, 126

Shokri, R., Theodorakopoulos, G., Le Boudec, J.-Y., and Hubaux, J.-P. (2011). Quantifying location privacy. In *2011 IEEE symposium on security and privacy*, pages 247–262. IEEE. 29

Singh, A., Click, K., Parizi, R. M., Zhang, Q., Dehghantanha, A., and Choo, K.-K. R. (2020). Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications*, 149:102471. 64, 126, 133, 134

Sovrano, F. and Vitali, F. (2021). From philosophy to interfaces: an explanatory method and a tool inspired by achinstein's theory of explanation. In *26th International Conference on Intelligent User Interfaces*, pages 81–91. 212

Sovrin Foundation (2020). Innovation meets compliance: Data privacy regulation and distributed ledger technology. Technical report, The Sovrin Foundation. 42, 43, 44, 46, 48, 49, 69, 78, 128, 137, 214

Tang, C. (2005). Ecdkg: A distributed key generation protocol based on elliptic curve discrete logarithm. *sE· CURECOMM*, pages 353–364. 152

The Graph (2020). The graph protocol. 95

Thorstensson, J. (2021). Ceramic improvement proposal 20: Cip-94 nft did method specification. `https://github.com/ceramicnetwork/CIP/blob/main/CIPs/CIP-94/CIP-94.md`. 222

Toyoda, K., Machi, K., Ohtake, Y., and Zhang, A. N. (2020). Function-level bottleneck analysis of private proof-of-authority ethereum blockchain. *IEEE Access*, 8:141611–141621. 127

Trail of Bits (2022). Are blockchains decentralized? Technical report, Trail of Bits. 33

Trakman, L., Walters, R., and Zeller, B. (2019). Is privacy and personal data set to become the new intellectual property? *IIC-International Review of Intellectual Property and Competition Law*, 50(8):937–970. 184

Truong, N. B., Sun, K., Lee, G. M., and Guo, Y. (2020). Gdpr-compliant personal data management: A blockchain-based solution. *IEEE Transactions on Information Forensics and Security*, 15:1746–1761. 118, 173

Ulbricht, M.-R. and Pallas, F. (2018). Yappl-a lightweight privacy preference language for legally sufficient and automated consent provision in iot scenarios. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 329–344. Springer. 167

Unterweger, A., Taheri-Boshrooyeh, S., Eibl, G., Knirsch, F., Küpçü, A., and Engel, D. (2018). Understanding game-based privacy proofs for energy consumption aggregation protocols. *IEEE Transactions on Smart Grid*, 10(5):5514–5523. 138

Utz, C., Degeling, M., Fahl, S., Schaub, F., and Holz, T. (2019). (un) informed consent: Studying gdpr consent notices in the field. In *Proceedings of the 2019 acm sigsac conference on computer and communications security*, pages 973–990. 27

Van Ooijen, I. and Vrabec, H. U. (2019). Does the gdpr enhance consumers' control over personal data? an analysis from a behavioural perspective. *Journal of consumer policy*, 42(1):91–107. 28, 29

Vitali, F., Palmirani, M., and Parisse, V. (2019). Akoma ntoso naming convention version 1.0. OASIS standard. 226, 228

Vitali, F., Palmirani, M., Sperberg, R., and Parisse, V. (2018). Akoma ntoso version 1.0. part 2: Specifications. OASIS standard. 226, 227

Vogelsteller, F. and Buterin, V. (2015). Erc-20 token standard. *Ethereum Foundation (Stiftung Ethereum), Zug, Switzerland*. 169

Vorick, D. and Champine, L. (2014). Sia: Simple decentralized storage. *Nebulous Inc*. 64, 67

Waldo, J. (2019). A hitchhiker's guide to the blockchain universe. *Communications of the ACM*, 62(3):38–42. 32

Wang, F. and De Filippi, P. (2020). Self-sovereign identity in a globalized world: Credentials-based identity systems as a driver for economic inclusion. *Frontiers in Blockchain*, 2:28. 214

Wang, S., Zhang, Y., and Zhang, Y. (2018). A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *Ieee Access*, 6:38437–38450. 47, 118, 119

Wang, X., De Martini, T., Wragg, B., Paramasivam, M., and Barlas, C. (2005). The mpeg-21 rights expression language and rights data dictionary. *IEEE Transactions on Multimedia*, 7(3):408–417. 171

Waters, B. (2011). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *International Workshop on Public Key Cryptography*, pages 53–70. Springer. 119

Westin, A. F. (1967). *Privacy and freedom*. Atheneum. 18, 26

Wikipedia community (2022). Decentralised system. `ipfs://zdj7WX5pBeuj18FDbNqxv55msheeEwFnmcRExintmC2men7ZS/wiki/Decentralised_system`. 5, 31

World Data Exchange Company (2022). digi.me. 42

Xu, H., He, Q., Li, X., Jiang, B., and Qin, K. (2020). BDSS-FA: A Blockchain-Based Data Security Sharing Platform With Fine-Grained Access Control. *IEEE Access*, 8:87552–87561. 118, 119

Xu, H., Luo, X. R., Carroll, J. M., and Rosson, M. B. (2011). The personalization privacy paradox: An exploratory study of decision making process for location-aware marketing. *Decision support systems*, 51(1):42–52. 29

Yamaguchi, Y., Amagasa, T., Kitagawa, H., and Ikawa, Y. (2014). Online user location inference exploiting spatiotemporal correlations in social streams. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1139–1148. 20

Yan, Z., Gan, G., and Riad, K. (2017). Bc-pds: protecting privacy and self-sovereignty through blockchains for openpds. In *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pages 138–144. IEEE. 68, 117, 118

Ye, C., Li, G., Cai, H., Gu, Y., and Fukuda, A. (2018). Analysis of security in blockchain: Case study in 51%-attack detecting. In *2018 5th International conference on dependable systems and their applications (DSA)*, pages 15–24. IEEE. 34

Zemler, F. and Westner, M. (2019). Blockchain and GDPR: Application Scenarios and Compliance Requirements. *2019 Portland International Conference on Management of Engineering and Technology (PICMET)*, 00:1–8. 69

Zhang, Y., He, D., and Choo, K.-K. R. (2018). Bads: Blockchain-based architecture for data sharing with abs and cp-abe in iot. *Wireless Communications and Mobile Computing*, 2018. 118, 119

Zhu, L., Xiao, C., and Gong, X. (2020). Keyword search in decentralized storage systems. *Electronics*, 9(12). 96

Zichichi, M. (2019). Data and scripts for IOTA vehicular scenario. DOI: `10.5281/zenodo.3552198`. 81

Zichichi, M. (2020). Data and scripts for IPFS and Sia vehicular scenario. DOI: `10.5281/zenodo.4572578`. 81

Zichichi, M. (2021a). Intelligible Decentralized Identity implementation. `github.com/miker83z/desp3d-intelligible-identity`. 231

Zichichi, M. (2021b). Intelligible Verifiable Certificate implementation. `github.com/miker83z/desp3d-intelligible-certificate`. 231

Zichichi, M. (2021c). Privacy Policy Generator from Smart Contracts. `github.com/miker83z/desp3d-policy-mco-generator`. 197

Zichichi, M. (2021d). Privacy Policy Parser for Smart Contracts. `github.com/miker83z/desp3d-policy-mco-parser`. 197

Zichichi, M. (2021e). Rust implementation of the umbral threshold proxy re-encryption scheme. DOI: `10.5281/zenodo.6548260`. 146

Zichichi, M. (2022a). Client tools to interact with Intelligible suite and Privacy Policy software. `github.com/miker83z/desp3d-client-tools`. 197, 231

Zichichi, M. (2022b). Github Repository including all software. `https://github.com/stars/miker83z/lists/phd-thesis`. 14

Zichichi, M. (2022c). Implementation of a dao that governs the exchange of data with an aggregator. DOI: `10.5281/zenodo.6548262`. 145, 149

Zichichi, M. (2022d). Intelligible packages based on DID and VC. DOI: `10.5281/zenodo.7132777`. 229, 231

Zichichi, M. (2022e). Privacy-policy dlt manager: Manage policies based on mco and dpv ontologies. DOI: `10.5281/zenodo.7132775`. 197

Zichichi, M. (2022f). Privacy Policy Smart Contract Templates. `github.com/miker83z/desp3d-policy-web3-templates`. 197

Zichichi, M. and Sparber, J. (2021). Personal data decentralized access control tests. DOI: `10.5281/zenodo.4572552`. 149

Ziegler, S., Evequoz, E., and Huamani, A. M. P. (2019). The impact of the european general data protection regulation (gdpr) on future data business models: Toward a new paradigm and business opportunities. In *Digital business models*, pages 201–226. Springer. 25

Zuboff, S. (2019). *The age of surveillance capitalism: The fight for a human future at the new frontier of power*. Profile books. 20, 26

Zyskind, G., Nathan, O., et al. (2015). Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184. IEEE. 68, 118