UNIVERSITÀ DEGLI STUDI DI URBINO CARLO BO

Department of Pure and Applied Sciences

Ph.D. Programme in Research Methods in Science and Technology

Cycle XXXV

PhD Thesis

DECENTRALIZING THE INTERNET OF MEDICAL THINGS: THE INTERPLANETARY HEALTH LAYER

Disciplinary Scientific Area ING/INF-05 - INF/01

Thesis written with the financial contribution of Regione Marche
Progetto di Dottorato Innovativo a Caratterizzazione Industriale

Supervisor:
Prof. Emanuele Lattanzi

Co-Supervisor:
Prof. Sandro Fioretti

Candidate:
Dr. Gioele Bigini

Academic Year 2021-2022

Amore e Dovere.
Love and Duty.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 About this Work

This document is the formal description of the work carried out during the Research Methods in Science and Technology doctoral path at the University of Urbino, in collaboration with the Marche Polytechnic University and the company DIGIT srl. This thesis is part of the research project funded by Regione Marche with Decreto del Dirigente della Posizione di Funzione (DDPF) n. 1189, whose initial goal was to develop an application for monitoring postural stability in individuals enhanced lately with the support of sharing sensitive health data through digital health devices.

1.1.1 Context

Nowadays, patient health data can be stored and processed digitally, quickly allowing remote access to healthcare professionals and institutions. Sharing information would significantly improve healthcare and medical research, as health data may be critical to essential discoveries. However, to exploit the full potential of these valuable digital assets, data needs to be managed with appropriate mechanisms and tools.

Medical records are often generated, collected and owned by healthcare providers in the context of patient care. Telemedicine will race to enable individuals to take autonomously through self-monitoring with the Internet of Medical Things (IoMT) devices. These devices have the great potential to generate millions of additional health data records over conventional collection methods in healthcare. However, they put new privacy and security issues in the spotlight.

Over time, data regulations have become increasingly stringent regarding protecting individuals' data. This has placed at the centre the attempt to shift from centralized to decentralized solutions that do not rely on a central authority, increasing awareness of the importance and value of one's data.

1.1.2 Motivation

The current state of patient health data management poses a significant challenge to the healthcare industry, where data is often distributed across multiple repositories and managed by various healthcare institutions and professionals. The result is keeping information isolated that makes it difficult to share critical data that could benefit patients, private entities, and researchers. Without a collaborative approach to data sharing, patients lose significant potential for treatment and care that could improve their health outcomes.

However, with proper management, the potential for data sharing and collaboration can be realized. Data controllers must comply with legal frameworks that govern data protection and privacy, such as the European General Data Protection Regulation (GDPR) [27], which imposes strict rules on the collection, storage, and use of personal data. As a result, access to patient data is often very restrictive, creating an environment of unfair competition where research institutions are under pressure to keep their data and procedures secret to avoid data leaks.

Despite these challenges, there is growing awareness of the value of patient data, which is driving the industry in a positive direction towards a more decentralized context. Decentralized data management allows for greater collaboration and sharing of data while ensuring compliance with legal and regulatory frameworks. This approach enables discoveries and the development of new services that can improve patient outcomes and offer new opportunities for researchers and private entities.

The management of patient health data in the current context is complex and challenging, but with proper management and a collaborative approach, the potential benefits of data sharing can be realized. Decentralized technologies and growing awareness of the value of patient data are paving the way for new discoveries and services that can improve healthcare outcomes and drive innovation in the industry.

1.1.3 Problem

As the healthcare industry rapidly embraces digital transformation, there is growing interest in how Internet of Medical Things (IoMT) devices can revolutionize the provision of medicine and monitoring. However, there is currently a lack of concrete evidence demonstrating how these devices can radically transform healthcare delivery, and more work is needed to fully understand how these systems can efficiently exchange information when they involve end-users [18].

To address these issues and create a system that enables the sharing of health information while maintaining its privacy and integrity, a multilayered data management system is needed. This system should include various levels of authentication, authorization, and verification, as well as traceability and detection mechanisms to ensure the security of health information.

Furthermore, it is essential to raise awareness about these processing procedures and place users at the center of the system while promoting stakeholder collaboration. By creating a decentralized traceability system and implementing incentive mechanisms, healthcare institutions can maximize the use of health data and initiate new research projects. Additionally, users can become custodians of their data and have the ability to create clinical histories that can be passed down from generation to generation.

To fully realize the potential benefits of IoMT devices and digital health solutions, it is important to continue developing and refining these systems. Through a collaborative approach and ongoing innovation, we can create a healthcare system that empowers patients, healthcare providers, and researchers alike, leading to better health outcomes for all. Awareness of processing procedures is also necessary to put users at the system's centre. Through such a decentralized traceability system and incentive mechanisms, healthcare institutions can be free to make the most of health data and start new research projects [20].

1.1.4 Objectives

This work aims to help decentralize the Internet of Medical Things (IoMT) by leveraging the properties of Distributed Ledger Technology (DLT) to support the process of sharing and tracing sensitive health data. This includes the development of a mobile application for collecting health data and introducing enabling technologies for the sharing of sensitive information, intending to make the user the data owner and controller, enabling them to share data in non-trusted contexts.

The work seeks to explore the space of improving health data sharing by formulating research questions, such as the ability to track data and determine their provenance without a central authority, enabling users to manage their data according to regulations, and increasing awareness about health assets and data availability through social networks. Through the implementation of a prototype application called Balance and the InterPlanetary Health Layer (IPHL), the work aims to solve the issues identified in the current system of sharing health research data and increase cooperation among entities involved.

The aims are to investigate and enhance the sharing of health data, as outlined by the following research questions:

- Can we provide a mobile application for collecting health data to be used as a use case in the Internet of Medical Things ecosystem?

- Can we enable the ability to track data and determine their provenance without a central authority?

- Can we enable users to manage their data according to the regulations by introducing decentralized technologies?

- Can we increase users' awareness about their precious health assets and the availability of health data by introducing social networks?

Through this work, we hope to stimulate new research and put the problem of health research data sharing under the spotlight, leading to a paradigm shift in the management of an individual's sensitive data and the possibilities that such a shift may bring in the investigation of treatments for new diseases in the future.

1.1.5 Contributions

This work provides a use case application in the IoMT ecosystem and a DLT-based solution for sharing and tracing data. The IoMT application is called Balance and can assess an individual's postural stability.

The developed solution should help the different entities in the system to be aware of each other's data processing steps. A decentralized system provides this awareness because multiple entities with different interests are involved, and the need for trust must be avoided. Ultimately, we expect to increase cooperation in the system further, creating a virtuous circle that improves the ecosystem's health.

Therefore, we have researched various solutions within the scientific community that address or explore the issue of traceability, such as platforms and projects. The broad applicability of the context, due to the breadth of the medical ecosystem and the lack of solutions related to health data sharing with support for traceability of data transformations, demonstrates the innovative aspect of our approach.

Other contributions concern an analysis of the solutions in the space or hypothesized by other researchers to understand the advantages and disadvantages of each one. Based on this analysis, we determined the best compromise by taking advantage of the problems of some of these aspects alone. Then, we analyzed the current frameworks available for developing DLT-based applications to select the one that best fits the goal we want to achieve.

The result of the study allowed us to present an approach that takes advantage of the best compromise. The implementation is provided within this paper through the related use case. Finally, we provide a final vision of what such an InterPlanetary-scale implementation might represent tomorrow.

In this work, we matched the academic research with its implementation counterpart. For this reason, we developed several components:

- the mobile application;
- the backend of the application;
- the infrastructure in which the sensitive data reside;
- the security measures taken;
- the infrastructure for decentralized data sharing.

We envision that such a development can positively contribute to interested stakeholders and users, contributing to new medical advancements.

1.1.6 Document Structure

Below we describe the structure of the document.

Chapter 2 is oriented toward the Internet of Medical Things and its investigation. In this chapter, we analyze the limitations of devices in the digital health space and their lack of sharing abilities. First, it describes current solutions to data tracking problems, including those involving medical data sharing and proposed alternatives in the field. The description of each solution is preceded by an introduction briefly explaining the solution's context, its main objectives, and the problems addressed. After describing the solution, a summary is presented, which reviews the solution, presents its main advantages and disadvantages, and briefly analyzes how well it fits our problem. In the end, this chapter compares the closest solutions to our addressed issues, describing their advantages and disadvantages and the trade-offs between each solution.

Chapter 3 introduces the IoMT use case proposed in this thesis, Balance, the application created to monitor an individual's posture. It describes the problem of stability, traditional analysis in the medical field is the digital alternative that uses smartphone sensors for stability assessment, then its implementation.

Chapter 4 introduces the InterPlanetary Health Layer and the decentralized version of Balance. Following the work, the implementation available for download and open source follows. It introduces the opportunities represented by cryptography in healthcare solutions and how it may be helpful in the development of the sharing layer. The final solution implements a cryptographic system through which information can be easily shared in a decentralized context.

Chapter 5 introduces use cases focusing on social networks with the dual purpose of providing insight on system maintenance and data availability. Indeed, a social network on top of a decentralized network could increase its maintenance, data availability over time and enable additional interesting scenarios.

Chapter 6 shows the experimental results performed and the performances of the single components developed within this thesis project.

Chapter 7 contains the final summary, contributions, the future vision, the questions to be explored and the conclusion.

1.2 History of the Internet

1.2.1 The Internet of Information

After its initial conception, the internet rapidly evolved into a client-server architecture. The idea was that the clients could establish a connection to a server, which must always be online and listening for requests. Since the server is responsible for fulfilling all client requests and handling all security and logic, they are called service providers. Placing trust in the server providers means the architecture has the disadvantage of centralizing the decision-making power and business logic implementation. This translates into individuals (the owner or an attacker breaking into the machines) with the potential to reach the user's data [17].

These concerns raised with client-server architecture have prompted individuals to seek alternatives, such as peer-to-peer [106]. However, the architecture had an unhappy debut, earning a bad reputation linked to illegal activities such as piracy since the beginning. Luckily, the reputation has slowly improved, thanks to the gaming industry, when applied to multiplayer games and content distribution networks.

But, over time, peer-to-peer networks continued to have one specific issue: there was no way to trust peers in the network. Since the entities running the peers are unknown, there is no guarantee that they will refrain from manipulating data and business logic. For this reason, the entities managing private servers have been incentivized to be honest because they were able to establish businesses. This is why client-server architecture prevailed for years until major incidents around data breaches went out [57]. From these assumptions, decentralized ecosystems were born.

1.2.2 The Internet of Value

The resolution of the trust problem quickly led to the structuring of a new web composed of decentralized networks. Bitcoin was the first decentralized network to provide a proper solution to the double spending problem of value moved between participants in an untrusted network [82].

This technology was later called blockchain, a constantly growing list of blocks containing all the transactions between peers. The decision for the legitimate chain is made according to the “most extended” chain rule. In order to create a new block in the chain, the users need to use computational power and solve a cryptographic challenge. The system can provide byzantine fault tolerance as long as more than 50% of the resource used for consensus of the system is in the hands of honest users. In order to provide this byzantine fault tolerance, Satoshi Nakamoto proposed an algorithm called proof-of-work to achieve consensus over which ledger is the legitimate one and which changes to accept. If the entire network can agree on which ledger to trust, there will be no double spending problem since honest users all agree on the same balance for every user in the network.

1.3 Blockchain

The term blockchain refers to a technology discussed for the first time over 20 years ago and is often confused with Bitcoin, which represents the first successful attempt to apply the technology.

From a technical perspective, all the elements for its implementation have been known for years. However, it was increasingly challenging to implement, which is why today's blockchain is so interesting to apply. Its name has been attributed due to its "chain of blocks" structure. The distribution of information is guaranteed in a decentralized manner, therefore, in the absence of a central entity, avoiding any tampering or minimizing it to the point that makes it a negligible possibility.

The birth of the concept comes from Stuart Haber and W. Scott Stornetta in 1991 through their paper "How to time stamp a digital document" from which practically all ideas are collected [49]. Blockchain is a growing list of data structures, called blocks, connected and secured through encryption.

A blockchain block can potentially contain any information in addition to that intended as mandatory. For example, to form a chain of blocks, it is necessary to know the previous and the next element in the chain. A Hash gives the link between a block and its previous one. The choice is not random at all and wants to find correspondence with fingerprints; that is, it is an identification method without leading to the characteristics of its bearer.

The chain originates from a single block called "Genesis Block". The content of the blocks can be anything, depending on the purpose, and it can be readable or not, depending on access to the blockchain. This is why we classify blockchains as "Permissioned" and "Permissionless" blockchains, in which a central entity manages the access to the chain. This fact also impacts decentralization and will be later discussed as well as the concept of "Public" and "Private" blockchains.

Implementations of any kind exist, exploiting several advantages. It is convenient to mention that blockchains find their roots in concepts such as Hash Encryption, Immutable Ledger, Distributed P2P Networks, Mining, and Consensus Protocol. Nonetheless, these properties are generally only adopted as a whole by some proposed solutions.

1.3.1 The Security of Blockchain

Between blocks in a blockchain, cryptographic proof links each block to the previous one, thus providing trust over the immutability of the ledger. If an attacker wanted to change a block (and so maliciously corrupt the data), they would have to reconstruct the whole chain. Because the cryptographic challenge is hard to solve and grants a reward, the process of solving it, and therefore the process of creating new blocks, is called mining.

A blockchain consensus algorithm is a basis for the blockchain's security, supporting the system's immutability and tamper resistance. The algorithm must consist of a challenge that is hard to solve but easy to verify if the solution is correct. This makes creating new blocks very difficult because the challenge needs to be solved, and since the blocks are linked to each other cryptographically, if someone wanted to change a block, they would have to re-do the proof of the entire chain that comes after it.

The blockchain consensus algorithms always require some resources to be used for the proof mechanism. This resource works as proof that most people are using the resource for a specific chain; therefore, we should assume that most of the network is looking at that specific chain as being the valid one. In that sense, we have a consensus. In the case of Bitcoin, it is computational power [7].

Still, the latter features do not mitigate the attack of altering the contents of the chain. The attacker could re-compute some of the chain and broadcast it to the network as a valid chain. The longest chain rule is the algorithm that allows honest users to achieve consensus over which chain to trust. The rule says that the chain to trust is always the longest, with the highest proof of work on it. This way, if an attacker wanted to cheat the consensus mechanism, he would have to control more than 50%

of the network's computational power in order to be able to produce blocks faster than the rest of the network. This makes the byzantine system fault-tolerant as long as more than 50% of the computational power is in the hands of honest users.

In the following sections, we better describe the most famous consensus algorithms.

1.3.2 Proof of Work Consensus Algorithm

Proof of Work (PoW) is a consensus algorithm introduced by Nakamoto that uses computational power as a resource to provide immutability and tamper resistance to the blockchain [11]. This mechanism uses cryptographic hash functions to create a unique representation of each block, the hash. Every block is linked to the previous one through a cryptographic challenge involving the use of the previous block's hash. So, the cryptographic challenge always depends on the previous block, and the content is the transactions created by the network. Therefore, the possibility of using pre-computed (already solved) cryptographic challenges does not work because the network naturally chooses which cryptographic challenge will be the next one by submitting transactions.

The nodes that are mining blocks are often called miners. Since the miners perform hashing, the total computational power used to secure the network is commonly called the network hash rate. The hash rate tends to increase with time due to the evolution of computer hardware, although the network's security remains the same. The challenge's difficulty is solving changes based on the network hash rate through a process called difficulty adjustment to ensure that the reward created remains the same regardless of the network hash rate. Therefore, it is preferable to measure the network's security as the energy being used for the mining process. This is dependent on the profitability of the mining process, which is dependent on the reward's value. The reward's value depends on the adoption of the blockchain network and the value people give to it. In the case of digital currencies, this can be measured using the currency's market capitalization.

A possible attack requires the attacker to have more than 50% of the computational power and, therefore, is commonly called a 51% attack. These attacks may have different intentions. There are possible DoS to the network using the consensus algorithm, but we will not go into much detail about these. Instead, we will focus more on the attacks that can deceive the network, causing it to trust false information.

The first type of attack is to change the blockchain's contents and re-compute the cryptographic challenges that link the blocks together to convince the network that the new content is the correct one. This attack is hard to reproduce since re-computing the cryptographic challenges from the block that the attacker intends to change is difficult. Normally, the attacker concentrates its computational power starting from the point of the last block, creating a hard fork chain that is a chain that grows parallel to the one being secured by honest miners. The steps to reproduce the attack usually involve:

1. The attacker starts mining on a parallel (malicious) chain without broadcasting the blocks to the network. Eventually, the malicious chain will grow bigger than the legitimate chain because the attacker has more than 50% of the hash rate.

2. The attacker then broadcasts the malicious chain to the network.

3. The network follows the longest chain rule and, therefore, trusts the malicious chain. The malicious chain overrides the legitimate chain, which is thrown away.

4. The malicious chain is now the main chain.

This attack is commonly attempted in cryptocurrency blockchain projects to spend double the coins. Therefore, what makes the networks secure is having the majority of the available computational power allocated to secure the network [7].

1.3.3 Proof of Stake Consensus Algorithm

Proof of Stake (PoS) is a consensus algorithm that uses the resource exchanged in blockchain transactions to provide proof of most of the network [22]. If the blockchain is used for money, the resource is the currency. The consensus algorithm was introduced in 2011 as an alternative to proof of work, introduced by Nakamoto in 2008. In this type of validation system, the blocks are linked through a hashing process over a limited search space instead of an unlimited search space like in proof of work. This enables the algorithm to be more energy efficient than proof of work. The users working for the consensus of the network are often called validators. The validators place some value at stake that can be lost in case they are caught voting for a block that is not valid.

The proof of stake mechanism uses the resource exchanged in the transactions to secure the blockchain. Therefore, the attack requires the attacker to gather more than 50% of the resource. This is supposed to be more challenging to achieve than with the proof of work system because the attacker would need to buy a considerable amount of the resource that is exchanged in the transactions. This would increase its value and make it even more difficult to buy such a huge amount due to the supply and demand rule. However, attacks are still possible on proof of stake consensus algorithms.

Following the same logic as proof of work, the networks are more secure as validators' stake increases. This comes from the incentive, which comes from the asset's value. In conclusion, the value it has for people makes a PoS blockchain network secure, regardless of the consensus algorithm. That value increases the value of the validation process reward, therefore attracting more consensus resources to be used to secure the network.

1.3.4 Public and Private Blockchains

Public blockchains are built to avoid control by any central entity on the network. This means that if peers in the network trust the technology, the blockchain constitutes a distributed network sharing a cryptographic secure, immutable ledger accessible to anyone. The ability to add blocks to the chain is guaranteed by a consensus protocol, a mechanism defined for the specific blockchain through which the participants converge to reach consensus. This is the best way to exploit blockchain in all its capabilities, and they are often referred to as "Permissionless" since there is free access to block content across the network.

Private blockchains are restricted to providers that determine various levels of access. This kind of implementation sacrifices decentralization for restricted control over the blockchain itself. It can be helpful in those cases where there is a need to have only some actors participating in adding blocks to the network for various reasons. It can be both "Permissioned" or "Permissionless". Anyway, in the case of a permissioned environment, the technology differs from the original vision of blockchain technology since the blocks are not freely accessible. These implementations have a small group of actors who can access it, and since the participants are very limited, the blockchain could not need a decentralized consensus protocol.

1.4 On-Chain and Off-Chain Transactions

Off-chain transactions are those transactions occurring on the blockchain which move the value outside of the blockchain. This is possible through several behaviours, such as swapping existing wallets' private keys or using a third-party or coupon-based interlocutor. This can lead to no fees, immediate settlement and complete anonymity without recording anything on-chain. Moreover, this kind of transaction can always be reversed if no operation has been done on-chain. In contrast, an off-chain transaction takes value outside of the blockchain. It can be executed using multiple methods. First, there can be a transfer agreement between transacting parties. For example, coupon-based payment mechanisms are based on off-chain methods: a participant purchases coupons in exchange for the crypto-tokens and gives the code to another party to redeem them [84].

On-chain transactions refer to those transactions which occur on the blockchain. The usual interpretation of cryptocurrency transfer so when the transaction is put in a block and validated by miners. Depending upon the network protocol, it becomes irreversible once a transaction gains enough confirmations from network participants. To reverse it, it would mean being able to tamper the ledger. On-chain transactions are supposed to occur in pseudo real-time because new blocks are broadcasted and added to the blockchain. This broadcasting need makes the transactions not occur instantly since the information needs to reach the whole network and a proper consensus. They also come at a cost, as miners ask for a fee for their transaction confirmation on the blockchain in the shortest possible time. The fee somehow determines the response's rapidity since miners could look for the highest fees.

1.4.1 Distributed File Storage

A Decentralized File Storage (DFS) offers an alternative way to store files to the traditional client-server models, i.e. where a domain name is provided and is then translated to an IP address. A DFS comprises a network of peer nodes that have their storage and follow the same protocol for content storage and retrieval. In Content-Based Addressing, contents are directly queried through the network rather than establishing a connection with a server. In order to know which DFS node in the network owns the requested contents, it is possible to rely on a distributed hash table in charge of mapping the contents, i.e. files and directories, to the addresses of the peers owning such data. DFS follows this approach and offers higher data availability and resilience using data replication.

An example is the InterPlanetary File System (IPFS), a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files [15]. The IPFS was born by looking at the data-sharing platforms of the past: Napster, for example, were an extensive file distribution system supporting over a million users. However, the applications were not designed as infrastructure to be built upon, i.e. as a general file system that offers global, low-latency, and decentralized distribution.

Moreover, industries were initially interested in such systems because the network was slower and moving files across two endpoints was relatively tricky. Instead, with the advent of wide bandwidths, HTTP did its job greatly. Anyway, several challenges could appear again as:

- Hosting and distributing content;
- Versioning and linking of massive datasets;
- Computing on large data across organizations;
- High-volume on-demand real-time media streams.

These challenges will be a reality in the future, and IPFS provides a high throughput content-addressed block storage model solution with no single point of failure where nodes do not need to trust each other.

1.4.2 Solid Pods and Linked Data

Solid is a project led by Sir Tim Berners-Lee, one of the inventors of the World Wide Web, developed collaboratively at the Massachusetts Institute of Technology (MIT). The project aims to make a paradigm shift from the current centralized one resulting in true data ownership as well as improved privacy [99].

The project aims to make a paradigm shift from the current centralized data management, resulting in the improvement of interoperability in data formats and data-sharing protocols. Solid is based on the Linked Data Platform (LDP)[1] W3C Recommendation and other standards, such as the Hypertext Transfer Protocol (HTTP), the Resource Description Framework (RDF) or the Web Access Control (WAC) specifications and it is a completely decentralized ecosystem which allows users' to control rather than be controlled by other entities.

Linked Data uses the RDF and HTTP to publish structured data on the web, linking data from different data sources and creating a global linked data cloud. Linked Data technologies and, more broadly, Semantic Web technologies have been enthusiastically adopted in the health

[1] https://www.w3.org/TR/ldp/

domain, and there is plenty of biomedical ontologies, datasets and other useful resources ready to be exploited. And even if these technologies are oriented to publish data on the web, they can also be used in private environments and still exploited, being Solid the best example.

The principles of Linked Data were first outlined by Tim Berners-Lee and provide a broad guide on which data publishers have started to realize the Web of Data [16]. The Web of Data can be accessed with Linked Data browsers, just as the traditional Web of Documents can be accessed with HTML browsers. However, instead of following links between HTML pages, Linked Data browsers allow users to navigate between different data sources by following RDF links. In this way, the Users can start with one data source and then move through a potentially infinite Web of data sources linked by RDF references. Just as the web of traditional documents can be explored by following hyperlinks, the web of data can be explored by following RDF references. Working on crawled data, search engines can provide sophisticated query functionality similar to that provided by conventional relational databases.

Since the results of queries are structured data and not simple links to HTML pages, they can be immediately processed, thus enabling a new class of applications based on the 'Web of Data'. The glue of the Web of Data is RDF links. An RDF link simply indicates that a piece of data has some relationship to another piece of data. The Web of Data is nowadays very wide, with datasets in different domains interconnected by millions of RDF links.

Solid is a protocol that builds upon this Web of Data. Instead of using a centralized system with a hub-and-spoke distribution paradigm, a decentralized peer-to-peer network is implemented in a way that adds more control and performance than traditional peer-to-peer networks such as BitTorrent. The other goals are that the system is easy to use, fast and allows developers to easily create applications.

The main objective of Solid is to allow information to be discovered and shared, thanks to the high interoperability of Linked Data. A user stores its personal data in 'Pods' (personal online datastores) or 'forms' or 'repositories' hosted wherever the user wishes. Its users can store their personal data in Pods hosted wherever they wish, either through a Pod provider or self-hosting.

Several applications could be implemented according to the Solid-related technical specifications [2], to allow interoperability between applications and the data being used/generated. Those may read resources on the Pods or even create new resources to be stored there if the user has given permission to the application.

1.5 Cryptographic Schemes

Access Control Systems (ACS) aims to regulate access to system resources by enforcing permissions based on a set of system policies to determine who can access information. Centralized ACS rely on a single authority to access the data and, therefore, carry the risk of a single point of failure and the loss of privacy [58]. DLTs solve the single point of failure by providing the means to implement decentralized ACS. Different approaches are: (1) Discretionary ACS, which enables the management of data stored outside of the DLT through the access control policy stored on the ledger [122]; (2) Mandatory ACS, which constrains the ability of a subject to access data through smart contracts [119]; (3) Role-based ACS, allows to achieve authentication based on user roles [28]; (4) Attribute-based ACS, grants or denies user requests based on user's attributes, object and environment conditions [74].

1.5.1 Proxy Re-Encryption

The Proxy Re-Encryption (PRE) offers a scalable protocol where it is not necessary to know the recipient of data in advance [9]. It is useful when communication between an arbitrary number of data owners and consumers is dynamic. PRE is a type of public-key encryption where an untrusted proxy entity transforms a ciphertext c, encrypted with a public key pk_1, into a ciphertext decryptable with a private key sk_2, without learning anything about the underlying plaintext. This is possible using a

[2] https://solidproject.org/TR/

re-encryption key rk_{1-2} generated by the data owner who has the key pair (pk_1, sk_1) and that divulges (to the proxy) the authorization of access to the plaintext to a data consumer holding the keypair (pk_2, sk_2).

1.5.2 Threshold Scheme

A (t, n)-threshold scheme can be employed to share a secret among a set of n participants, allowing the secret to be reconstructed using any subset of t (with $t \leq n$) or more shares, but no subset of less than t. In a network where more than one server keeps secret shares, a mutual consensus can be reached when t nodes provide the shares to a secret recipient, enabling the secret to be known. This can be used to provide data protection to a user sharing a secret since none of the servers can obtain the whole secret without the help of other $t - 1$ servers.

1.6 Health Data

1.6.1 Health Data in the Context of Big Data

Big Data refers to large subject data sets with a complex structure that is difficult to store and analyze to find patterns and correlations that are typically not immediate. Part of these data is subject data that are becoming increasingly valuable because of the benefits derived from their analysis. Data mining is the process of discovering interesting and valuable structures in large data sets.

The attention created by this process has also attracted the attention of regulators, leading data regulations to be increasingly stringent in protecting people's data. Moreover, the consequence of this has raised awareness among individuals of the importance and value of their data, encouraging them to protect it and exploit its value. This, along with the emergence of solutions for decentralized data sharing, is raising awareness among subjects of the importance and value of their data.

The Health Data growing as Big Data is increasing in popularity [12]. The need for sharing health data among multiple parties has become evident in several applications. The use of subjects' data to make crucial decisions and improve people's lives is growing interest in the community. Research on large health data sets provides significant opportunities for improving health systems and individual care.

The widespread adoption of research using electronic health records (EHRs) is pushing the collection of sensitive clinical data. The evolution of technologies like the internet makes remote access to data an almost instant process. In order to take full advantage of the value of the subjects' medical data, there is a constant need for the proper mechanisms and technologies to ensure this data availability. Currently, medical records are spread over multiple repositories, making access to this data very challenging, compromising individual health care and health research.

The attention from research projects and regulators has raised subjects' awareness of the value of their health data and the importance of protecting it more than any personal data. Although essential for public health, patient care and clinical research, this sharing process raises privacy concerns because health data is susceptible from social and economic points of view. These concerns are raising the attention of the regulators of the health field.

This tightening in regulation and the awareness of subjects to protect their health data have led to an increased demand for mechanisms to grant subjects privacy when sharing health data.

1.6.2 GDPR Regulation in Europe

The most important law in the EU Data Protection regulatory framework is the General Data Protection Regulation - GDPR, which harmonizes the rules on the protection of EU citizens' data [27]. However, the GDPR only defines high-level requirements and user rights. How the GDPR is interpreted depends on national Data Protection Authorities and official bodies, such as the European Data Protection Board and ENISA. There are also other laws i.e. ePrivacy Regulation, or the new Cookie law envisioned for 2019.

According to the GDPR, Health Data are "all data pertaining to the health status of a data subject". As such, they are considered a special category of Personal Data. The definition provided by the GDPR is further explained by the European Data Protection Board (formerly the Art. 29 Working party), the EU body with advisory status on data protection matters. This identifies situations in which personal data will be considered Health Data. Examples of Health Data can be heart rate (ECG), weight tracking, blood pressure, healthcare payments, step counts, heartbeat tracking, diseases and many others. Although the definition provided by the GDPR may seem straightforward, the presence of some "grey areas" makes data categorization difficult. Therefore, it is essential to define the type of data you will collect: different data bring about different legal challenges.

GDPR and EU data protection laws identify different roles when handling data:

- Data Subjects: individual users to whom the data belongs.
- Data Controllers: the entity responsible for data collection and management. For example, when delivering a service directly to consumers (e.g. a fitness/disease tracking app). If you deliver your service to a hospital, and then the hospital delivers your services to its users, you are not (usually) nominated as Data Controller.
- Data Processors: These entities help deliver a service. Chino.io, for example, is a Data Processor that provides a set of services. As suggested before, the act of providing a service to a hospital, and then the hospital delivers the service to its users, usually means being a Data Processor (instead of Data Controller).

Assigning roles is the first step in identifying the requirements to be satisfied and implemented within systems.

Chapter 2

Exploring Data Management in the IoMT

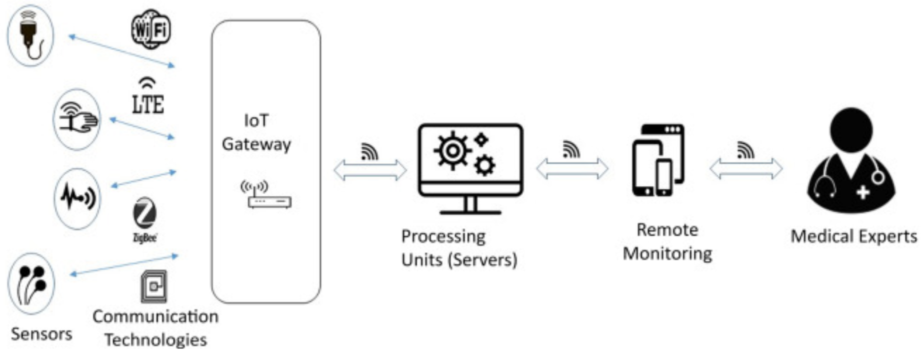


Figure 2.1: The Internet of Medical Things

Over the past 30 years, the technology sector moved so quickly that many consolidated products and services have been replaced to provide better solutions in a disruptive innovation process. Computer science and engineering forced many industries to innovate to survive, resulting in new products and services and new professions and improvements. A rising sector is the Digital Health field, represented by the combination of computer science and healthcare to empower professionals and increase the well-being of people with innovative systems. The healthcare sector enormously benefited from the introduction of computer science, but much of the work still needs to be done, such as enabling the sharing of information and transitioning to decentralized architectures.

In this flourishing landscape also comprising the Internet of Things sector, the narrower sphere of the Internet of Medical Things can be considered at the early stages of its potential development. The word “Medical” emphasize the specificity of the implementations in the IoT space.

But, sharing information is not a critical issue to address in several applications. If the information processed by the devices does not constitute sensitive information, it is easier to find a way to share them, i.e. with techniques like anonymization. Nevertheless, these techniques' drawbacks are represented by the reduced amount of intrinsic information, which could reduce data exploitation. In a perspective in which medical information can lead to research assisted by emerging data analysis tools such as machine learning, it is essential to aim at the complete integrity of the data respecting privacy through transparent data management.

People using Internet of Medical Things solutions are involved in the process of collecting health data. However, they would not be fully rewarded for their activities except through performances coming from the usage of a service, even though their contribution is also scientific. For example, every time an individual reaches a doctor, he contributes to the practical knowledge of the doctor himself. So, imagining using a mobile Internet of Medical Things application is still valid: using the application, it is implicitly possible to contribute to something useful for the scientific sector, which means giving free contributions at no cost. Consequently, data from users represent a resource and should have the possibility for them to choose what to give free, to hold or sell in the process.

The interest in the field brought researchers to employ multiple data-sharing technologies. A technology that promises to achieve this goal is Distributed Ledger Technology (DLT) and its three fundamental properties such as decentralization, immutability and transparency. The potential of the technology is to overcome the barriers represented by privacy and security for the health sector in which the sharing of information without the explicit consent of the individual constitutes a substantial violation of a person's rights.

2.1 Related Works

An increasing amount of work dealing with DLT and IoT can be retrieved within the current scientific literature, explicitly pointing out opportunities for overcoming the challenges posed by security and privacy in the healthcare sector. But a lack of a consistent amount of papers in the specific field of the IoMT confirms that the field appears to be only partially investigated, leaving room for more attention, research, and studies.

In conducting our research, we started by gathering relevant information and identifying key review articles in the field. When studying the Internet of Medical Things (IoMT), a good starting point has been to select those providing a comprehensive overview of the field. This has been used as a guide to help identify the most important areas of research and development in the field, as well as other contributors and their related works. After that, we identified relevant primary research studies, which can then be analyzed and synthesized to understand the current state of IoMT research better. This process involves carefully reading and analyzing each study's methods, results, and conclusions to identify key findings, limitations, and areas for further research. Through these investigations, we subsequently obtained our requirements, which will be mentioned throughout all the work.

In several works, researchers focus on addressing specific problems related to blockchain limitations or giving informative and procedural roadmaps on how to start a healthcare project on blockchain. The articles from Pilkington [94] and Borovska [21] addressed the expanding segment of the Internet of Medical Things by providing insights on how these devices could contribute to big data, potentially resulting in the development of new medical solutions through the application of machine learning techniques. They examined blockchain technology as a medium for healthcare data management in general, taking into account the shortcomings of private and centralized organizations and analyzing blockchain's transformative role in managing electronic health records. The intersection of big data analytics and precision medicine can be advantageous for detecting future diseases. According to Mackey et al. [72] and Agbo et al. [3], the role of blockchains in facilitating data management, provenance, and security has the potential to transform healthcare. For instance, the use of blockchain to ensure the privacy of electronic health records or to facilitate the credentials and licensing of medical professionals. In addition, they presented several examples of blockchain-based solutions for healthcare application scenarios, which typically need more prototype implementations. As a result, they highlighted the current state of development of blockchain applications for healthcare, concluding that more research is required to improve and evaluate the impact of the adoption of this technology.

We classified papers based on the macro areas they belong to Challenges and Implementations as shown in Table 2.1.

2.1.1 Addressing Issues in the IoMT Space

Data Management in the healthcare industry is crucial due to security and privacy issues. From the IoMT perspective, this goal could be even more difficult, as today mobile devices are generally more valuable to hackers than other devices. Several researchers, such as Chukwu and Garg [25], focused on discoveries in the field of privacy, security, cost, and performance, highlighting present issues, frameworks and implementations.

The articles by Khezr et al. [60] and Banerjee et al. [13] discuss the data management issues in the Internet of Medical Things devices. Specifically, Banerjee et al. [13] examines the issue of tracking datasets on the blockchain as a means of data sharing. This is especially important because they avoid sharing information directly on the blockchain. It is difficult to conceive of a blockchain solution that stores data on the blocks because the technology needs to scale efficiently.

Information processing and sharing are just some of the obstacles to overcome. Privacy and security concerns transcend data management

Challenges	**Works**
Privacy and Security	Nanayakkara et al. [83] Neshenko et al. [86] Seliem and Elgazzar [101]
Data Management	Banerjee et al. [13]
Frameworks for Blockchain-Based IoMT	Fernández-Caramés and Fraga-Lamas [42] Al-Turjman et al. [6] Pavithran et al. [93] Chukwu and Garg [25]
Implementation Attempts	**Works**
Scalability	Mazlan et al. [79]
Data Management and Interoperability	Zhang et al. [114] Saha et al. [98]
Healthcare Sector	Hussien et al. [53] Hölbl et al. [52] Zubaydi et al. [121] Khezr et al. [60]
Industrial Sector	Al-Megren et al. [5] Ahram et al. [4]

Table 2.1: Challenges and Implementations in the IoMT

and blockchain technology. The review by Nanayakkara et al. [83] examines a variety of healthcare-based applications in the field of IoMT, examining the threats and risks associated with the field. The same IoMT devices, for instance, could be tampered with, putting an individual's information at risk. In this paper, the authors discuss how IoMT devices could be categorized according to the risks associated with them. They consider the Middleware Threats, Application Layer Threats, and Business Layer Threats posed by the sensors that comprise medical devices.

Other researchers provided a strategy for approaching a new blockchain based Internet of Medical Things project. Fernández-Caramés and Fraga-Lamas [42] and Al-Turjman et al. [6] focus on the IoT context, attempting to establish a framework for identifying the components and design elements of a new application in order to develop it. Taking into account the IoT's general architecture, they view the blockchain as a medium for cloud applications. Pavithran et al. [93] also includes application development strategies. The authors centred their attention on the construction of such applications and the identification of the ecosystem's most important factors and components. They are simulating two distinct types of blockchain implementation and discussing the advantages of using device-to-device architectures as opposed to gateway-based implementations in terms of throughput.

2.1.2 Implementation Attempts in the IoMT Space

Numerous researchers have considered the variety of available applications to comprehend better and define future directions and obstacles.

Mazlan et al. [79] addresses the scalability issues that blockchain poses in general, in all contexts, not just the IoMT context. They suggest that in a number of instances, the scalability issue could be mitigated in two main ways: storage optimization and blockchain redesign.

Data Management and Interoperability are important aspects of IoMT and the healthcare industry. The review from Zhang et al. [114] summarizes the existing blockchain-based systems and applications, classifying them by traceability and data security protection and attempting to comprehend industry development opportunities and challenges.

The work from Saha et al. [98] examined the capacity of blockchain systems to support data integrity, dependability, and the capacity to address cloud security issues. Specifically, they investigated the current state-of-the-art blockchain-based medical healthcare systems and discussed a variety of works in the field.

Regarding healthcare use cases, Hussien et al. [53], Hölbl et al. [52], Zubaydi et al. [121], Khezr et al. [60] conducted their research by analyzing the use of blockchain in healthcare applications and organizing them into a taxonomy. Their work sheds light on the growing number of studies pertaining to the adoption of blockchain technology, including data sharing and security concerns between healthcare providers, by highlighting its potential to revolutionize the healthcare industry.

Last but not least, Al-Megren et al. [5] examined the Internet of Things, healthcare, supply chain management, and government sectors. They discovered the increasing maturity, benefits, and challenges of blockchain technology, highlighting the need for further research in all sectors at the time. For each industry, they described the use cases where blockchain solutions are attempted to be implemented.

2.2 DLT-based Implementations to Address Privacy and Security

From a general perspective, the IoT infrastructure is made up of several devices connected to the Internet able to communicate with each other, i.e. smartphones are the most widely diffused personal devices, a building block of the IoT ecosystem. However, any electronic device that can interface with and communicates with other peers in the network could be part of the IoT network, such as home automation systems or voice recognizers.

Communicating over the Internet poses essential security and privacy issues. For example, a voice recognizer needs more interaction with the user and listening to improve its learning. At the same time, a home automation system could be hacked by a malicious user getting access to important functions of the smart home solution. Therefore, the context in which the devices operate is fundamental for understanding the rules regarding privacy and security. For the specific subset of the Internet of Medical Things, the sensitivity of the user's data and the devices' vulnerabilities represent serious problems.

Sometimes the devices need to exchange information on the network rather than communicate. This makes the device a perfect target to hit. The solution to avoid information leakage is avoiding solving the problem: all the processing is moved onboard the device, avoiding data transmission on the network or transferring information only once masked or anonymized, definitely impacting data quality and integrity. In other words, from one side, processing data on the device limits the risk of information leakage by malicious users at the cost of performance, while the anonymization techniques prevent identification but losing part of the data. This implicates extensive data stripping and largely excludes data linkage and update, sometimes essential activities for the Internet of Medical Things [80].

The available integration of Internet of Medical Things architecture and DLTs are a few. Nevertheless, the great news is that IoMT growth estimates are of about 140 billion dollars by 2026 [96]. In Table 2.2 a classification of those trying to focus on the kind of solutions and their degree of decentralization is shown.

As the classification shows, building fully decentralized solutions seems to need to be investigated.

2.2.1 Data Management and Interoperability

The data sharing issue is of high interest to the medical field. The possibility of freely sharing sensitive information between professionals and health institutions would allow a great leap forward from the point of view of research in the medical field, taking advantage of the most advanced machine learning techniques that computer science is offering.

Area	Type	Work
Data Management and Interoperability	Mixed	Jiang et al. [59]
	Mixed	Xu et al. [112]
	Centralised	Wang et al. [111]
	Centralised	Dey et al. [31]
	Centralised	Azbeg et al. [10]
	Centralised	Nguyen et al. [87]
Data Crowdsourcing	Decentralised	Fernández-Caramés et al. [43]
	Centralised	Rupasinghe et al. [97]

Table 2.2: Implementations Decentralization

There are several ways of sharing data on the DLTs: storing data in their blocks (not feasible due to the blockchain trilemma) or using them as a medium of data provenance, i.e. storing the positions of data in the blocks. In the latter case, data is never moved within the network; instead, it is accessed knowing its position during the time.

Nguyen et al. [87], Dey et al. [31], Azbeg et al. [10] and Nguyen et al. [87] focus on the safe transmission of healthcare data. They specifically focus on cloud-based IoMT devices used for monitoring disorders finding in the blockchain a safe system for data sharing between devices through the help of smart contracts.

A step through interoperability has been made by Jiang et al. [59] and Xu et al. [112] through a solution more prone to decentralization. Both tried to reach a significant decentralization by using a combination of different blockchains to achieve different scopes.

2.2.2 Data Crowdsourcing

Some researchers focus on the hypothesis that the blockchain can actually be a new way of doing crowdsourcing with monetization. Fernández-Caramés et al. [43] and Rupasinghe et al. [97] build a decentralized solution based on smart contracts for achieving this goal.

2.3 Achieving Self-Sovereign Data Management

In the research works appear strong the need to reach a user-centric approach where the user has complete control over her/his data. Several attempts have been made to solve this problem, but they still need to offer a final solution that goes forward in this direction. What we just highlighted can be further clarified with the considerations that follow.

Decentralization of IoMT architectures could lead to user-centricity. Being user-centric means the user has complete control over his data. IoMT systems generally rely on a centralized entity through which they give a service, which happens even in several proposed DLT implementations.

A correct user-centric implementation with DLTs solution should give the participant the ability to no longer rely upon a central provider to take care of his data. This goal should enable worldwide interoperability too.

2.3.1 Explored Implementations

Jiang et al. [59] propose a platform named BlocHIE based on blockchain for data sharing between individuals employing two Blockchains, namely EMR-Chain for medical institutions and PHD-Chain for individuals, both able to submit and share healthcare data. They handle healthcare data through the combination of off-chain storage and on-chain transactions. The off-chain storage is achieved by storing the data in the distributed databases of the hospitals, while on-chain verification is achieved by including the hash value of each medical record in the transaction.

Because medical institutions usually submit very privacy-sensitive data as medical reports and treatments (because of healthcare professionals) while individuals are more prone to submit a considerable amount of data (because of data generated by IoMT devices), such kind of approach provides from one side a centralized solution where institutions are able to keep control of user data and to the other side a decentralized solution for data provenance. The whole system could then be considered as a mixed solution between centralization and complete decentralization but still system-centric.

Xu et al. [112] propose Healthchain, a large-scale health data management scheme based on blockchain where users have full control of their data as well as access policies. The system uses two blockchains, namely Userchain, a public blockchain used to publish users' data, and Doccchain, a private blockchain of healthcare institutions used to publish doctors' diagnoses. For the researchers, this scheme should ensure the design goals of supporting large-scale IoT devices, reaching a high efficiency, and creating a real-time online diagnosis system, which could preserve privacy, ensure accountability and, finally, manage permissions.

It is composed of five entities: the IoT Devices; the User Nodes, able to manage one or more IoT devices aggregating, encrypting and sending data to the storage node; the Doctor Nodes, which are doctors or companies providing healthcare services; the Accounting Node: a specific node maintained by the consortium to verify whether the transactions from doctor nodes are correct and valid; Storage Nodes, IPFS-based systems maintained by the consortium that collaboratively store complete encrypted users' IoT data and encrypted doctors' diagnoses in a distributed manner.

Basically, IoT devices send health data to the User Nodes that encrypt the data forwarding them to an IPFS storage that takes care of the transaction to the Userchain. The Doctor Node is then able to give real-time online diagnoses readable by patients reading the Docchain.

Wang et al. [111] proposes a blockchain-based eHealthcare system using Hyperledger Fabric interoperating with wireless body area networks (WBAN), which employs WBAN to network patient devices and blockchain technology as the data management system. Participants in Hyperledger Fabric's private, permission-based network have mutual trust. Patients, physicians, healthcare institutions, and suppliers compose the system's actors. The proposed workflow is as follows: the patients transmit the sensor data collected via the WBAN to the centralized devices. The devices await instructions from the centralized device, which will then generate the final record of data to be submitted to the blockchain in order to update the patients' physical data. In this architecture, data are effectively protected, as only healthcare professionals have access to patient records. In any case, Hyperledger Fabric cannot be considered a blockchain because it lacks one of the essential blockchain characteristics: decentralized consensus. Hyperledger Fabric does not require any consensus mechanism, making it difficult to determine whether the ledger has been tampered.

Dey et al. [31] propose a blockchain-based model in which a sensor collects real-time data on a patient's medical condition and stores it in the blockchain for use with smart contracts at a later date. The solution utilizes IPFS for off-chain storage, and a smart contract connects the sensor to the blockchain. This enables IoMT devices to discover one another and begin exchanging data off-chain or with the platform. The model lacks a mining solution and therefore does not permit sensors to add new blocks to the blockchain (that are considered low power). In this configuration, the model could continue to be system-centric, permissioned, and non-decentralized.

Azbeg et al. [10] proposes a platform architecture for diabetes self-management based on a permissioned Blockchain. According to the researchers, integrating blockchain technology with low-power devices, such as those used for diabetes monitoring, is challenging. This objective was attained by registering each new device in the blockchain by its owner, who could grant access permission. Therefore, the system consists of medical devices, the blockchain, and medical institutions (that maintain the blockchain). The connection to the blockchain is established via a gateway (a smartphone) that can encrypt and route data to an IPFS database that authorized physicians and healthcare teams can access. The healthcare institutions are the network's full nodes; they store data pointers, validate transactions, and generate new blocks, similar to other centralized solutions.

The authors of Nguyen et al. [87] propose a system for sharing datasets within an IoMT infrastructure comprised of mobile devices and cloud computing. Their plan is to prioritize the integrity of the downloadable datasets and make them available for sharing. They created an access control mechanism utilizing smart contracts and delegated the maintenance of the datasets by the repositories to a central hub while distributing and storing information such as the address, ownership, and sharing policies on the blockchain. The blockchain is public and contains no sensitive information. Furthermore, this attempt is intriguing because the owner of the dataset can remove the data at any time, rendering them inaccessible. In this manner, the blockchain may contain several blocks with “empty links” (stored as transactions).

Fernández-Caramés et al. [43] proposed a system for remotely monitoring patients and alerting them of potential dangers. It collects data from smartphones and transmits it to a remote cloud or to distributed fog computing nodes. The system deploys a decentralized storage system that receives, processes, and stores the collected data and offers cryptocurrency as an incentive for participation in order to facilitate the exchange of data between healthcare parties. The architecture consists of a user interface that provides access to the stored data, a decentralized storage system that replicates the collected data and distributes it automatically across multiple nodes, and a distributed ledger that employs smart contracts to reward participation. This approach is intriguing and utilizes blockchain for data crowdsourcing.

Rupasinghe et al. [97] propose a conceptual blockchain-based fall prediction model using smart contracts and the FHIR (Fast Healthcare Interoperability Resources) standard protocol. They identify four roles: person under care, primary care provider (or long-term provider), secondary care provider (or short-term provider) and temporary caregiver. Each of these entities maintains its own electronic health record management systems and can be considered as data sources for the final prediction model. The architecture is based on a permissioned and private blockchain that leverages smart contracts for accessibility, creating different access levels based on each user category.

Chapter 3

IoMT Application Use Case: Balance

Balance is an innovative digital health application that represents a scientific and engineering contribution to the field of postural stability monitoring. It enable the analysis of human stability through the sensors embedded in a smartphone.

The development of Balance involves the integration of various technologies and disciplines, such as mobile computing, signal processing, machine learning, and human physiology. The algorithm employed in the application enables the processing of accelerometer and gyroscope data to determine postural stability metrics. These metrics are then analyzed to determine the individual's level of balance and stability, allowing for early detection of balance disorders and other health issues.

3.1 Postural Stability Foundations

Postural control refers to maintaining, achieving, or restoring that steady position during any static or dynamic posture or activity. By analyzing the position and dynamics of the barycentre or the projection of the barycentre on a plane parallel to the ground, posturographic (or stabilometric) analysis makes it possible to determine a patient's stability in an upright position. Stabilometric analysis does not assess actual balance but rather the patient's capacity to maintain an upright, balanced position.

When the object's projected *Center of Mass* (COM) hits the supporting surface, it is said to be in mechanical equilibrium. As more force is needed to disrupt this situation, stability rises. The human body likewise follows these rules since it is erect and has a relatively high COM compared to the little support foundation provided by the feet. The human body has the innate ability to use muscle activity to offset the effects of gravity through postural control, but for an inanimate item, a similar scenario would result in displacement or falling.

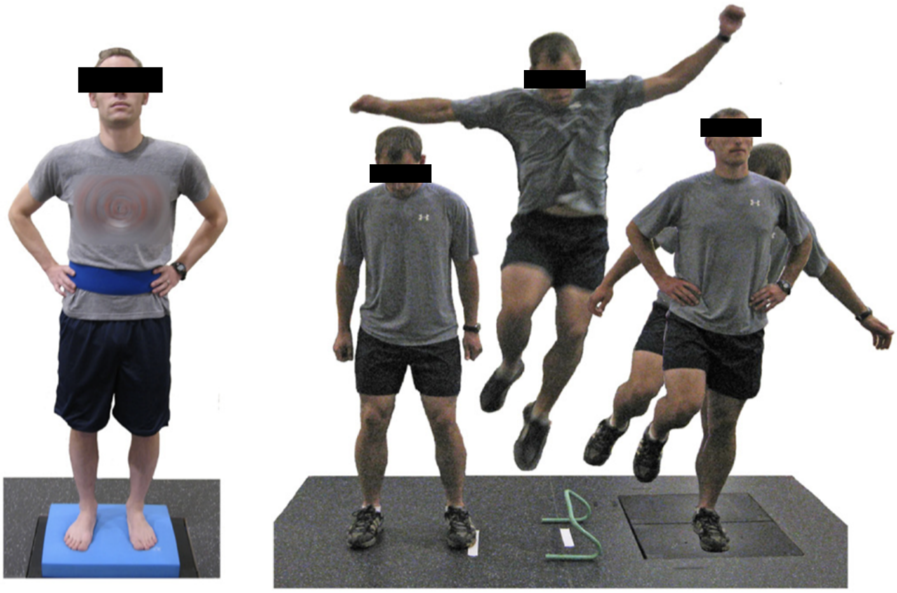
The methods used for posturographic analysis can usually be grouped into two categories: *static* or *dynamic*, as shown in Figure 3.1.

- In *Static* evaluations, the patient is placed standing on a flat horizontal measuring surface (often a stabilometric platform) with eyes either open or closed, without any external perturbation.
- In *Dynamic* evaluations, the posture is perturbed by external stimuli unpredictable by the patient to assess his ability to resume the initial posture.

These strategies investigate distinct parts of the human posture control system and permit to gather information independently of one another. In the *static* position, except for the plantar skin receptors, the majority of the human sensory system is active below a threshold limit (and can thus be considered at rest), whereas, in the *dynamic* position, all receptors are active beyond the threshold limit. Moreover, when *static*, the only source of postural instability is internal, thus the body can anticipate and rectify disruptions, whereas, in the *dynamic* condition, the disturbance is external and unanticipated.

3.1.1 The Force Platform

Internal forces, defined by the movement of muscles, and external forces, exchanged by the body with its environment, cause the body to move.


[redacted] [redacted] [redacted] [redacted]

(a) Static Evaluation (b) Dynamic Evaluation

Figure 3.1: Postural Stability Evaluation

The most popular stabilometric assessment device is the force platform shown in Figure 3.1a. The force platform can detect the external forces applied to force transducers by detecting the deformation they induce. Because the pressure is distributed on the foot's bearing surface, the point of force application is referred to as the *Center of Pressure* (COP).

During the test, the participant is put in a neutral standing stance in the center of the platform with arms at the sides. The most common postural acquisition protocols are:

- The monopodal test: the patient performs the test while standing on one leg.

- The Romberg test: the patient performs the test with both eyes open and closed to determine the influence of the visual system on posture.

- The cervical interference test: the patient performs the test while holding the head erect and while keeping the head flexed to assess cervical influences on posture.

3.1.2 The Single Inverted Pendulum Model

Mathematical modeling of the posture control system has yet to be considered a solved problem, particularly concerning its coordination, control principles, and associated motor commands. The main modeling assumption of state-of-the-art scientific literature relies on the representation of human body in standing balance as an inverted pendulum system.

The primary quantities used to derive the model are illustrated in Figure 3.2, which for the purpose of simplicity simply takes into account the anteroposterior orientation of a person standing still. In this framework, sway movements represent the back and forth oscillations of the pendulum as the effect of two opposing forces:

1. The gravity force, destabilizing the system;
2. The stabilizing effect of ankle muscles.

The system made up of the ankle joint, feet, and the rest of the body is modeled by the single inverted pendulum positioned around the ankle.

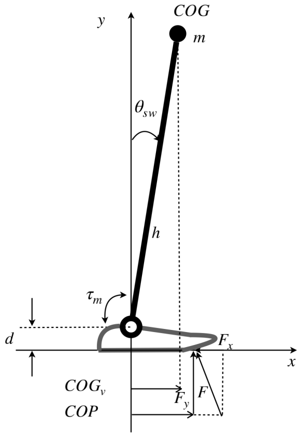


Figure 3.2: Modelling the Single Inverse Pendulum Problem

Specifically, the motor torque owing to the muscles operating around the ankle counterbalances the momentum of the applied ground reaction force F at the *COP*. According to the traditional Newton-Euler mechanics equations, the following equation describes the system's dynamics [33]:

$$\frac{\mathrm{d}^2 COGv}{\mathrm{d}t^2} \approx \frac{mgh}{I}(COGv - COP) \tag{3.1}$$

where *COGv* is the projection of the *Center of Gravity*, *COP* the *Center of Pressure*, h is the distance between the ankle and the barycenter and I is the moment of inertia of the body around the ankle joint.

From equation 3.1, the transfer function of the dynamical system with input *COP* and output *COGv* can be written as:

$$\frac{COG_v(\omega)}{COP(\omega)} = \frac{\omega_0^2}{\omega^2 + \omega_0^2} \tag{3.2}$$

In equation 3.2, ω represents the angular frequency and ω_0 the natural angular frequency of the inverted pendulum, shown in Equation 3.3. It follows that, as frequency grows, the output of the system progressively decreases, similarly to low-pass filters.

$$\omega_0 = \sqrt[2]{\frac{mgh}{I}} \tag{3.3}$$

Equation 3.2 also permits the derivation of *COGv* from *COP*; given the natural angular frequency of the pendulum, the time series of the *COGv* can be estimated by computing the inverse discrete Fourier transform of the product between the transfer function and the Fourier transform of the *COP*. We can use this method to estimate the center of gravity once we have recorded the *COP* and compare it with the center of gravity resulting from the measurements taken by the smartphone onboard accelerometers because we compare the statokinesigrams obtained by using a force platform with those collected by means of a smartphone. Several studies indicate that the *COP* signal (also known as a statokinesigram) represents a force, whereas the *COG* signal is related to the swing of an inverted pendulum and thus represents a movement [14, 78]. Consequently, the *COP* trajectory is a time series directly representing the forces generated by stabilizing muscles; the *COG* can be viewed as a variable controlled by the *COP*. The low-pass filter behavior resulting from modeling the human body balance as an inverted pendulum explains the relationship between these two quantities, where the frequency bandwidth of the *COP* signal is significantly greater than that of the *COG* signal, which oscillates with the majority of its components below 1 Hz.

The *COG* projection can be thought of as a filtered version of the *COP* in the frequency domain, with a cutoff frequency of roughly 0.4 Hz for a typical person. The trajectory of the *COP* in two dimensions-AP and ML-is typically the raw data acquired for posturographic analysis.

On a smartphone, we process and analyze data obtained from accelerometers, with the goal of measuring the *COG* sway (the trajectory along the AP and ML axis).

3.2 Evaluation through Smartphones

In what follows we describe how the foundations of postural stability were translated into a digital device.

3.2.1 Data Acquisition Sensors

The majority of people own smartphones for personal or professional use. To provide users with an increasing number of features, these devices make extensive use of sensors. Popular sensors include the proximity sensor, which is used to turn off the screen when the user brings

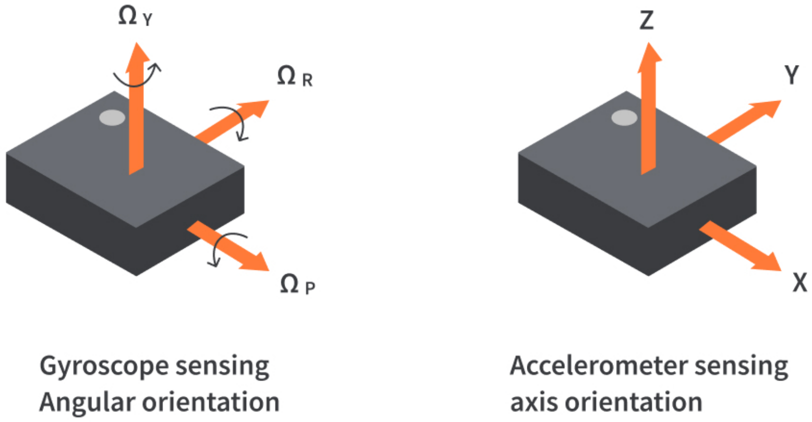


Figure 3.3: Gyroscope and Accelerometer Orientations

the phone close to his or her ear during a call; the brightness sensor, which adjusts the screen's brightness based on the light conditions present; and the increasingly popular fingerprint reader, which is replacing passwords to unlock the device or validate online transactions (such as electronic payments). Some smartphone manufacturers are integrating health-monitoring sensors such as a thermometer, heart-rate monitor, humidity sensor, and pedometer (to measure the number of steps taken by the user accurately).

The most common sensors, however, are the accelerometer, gyroscope, and GPS because they enable tracking systems and automatic screen rotation by allowing the device to know its location and direction. .

The accelerometer detect devices' linear acceleration, or acceleration along an axis. Conceptually, an accelerometer behaves like a damped mass attached to a spring: when the sensor is affected by acceleration, the mass moves by inertia, compressing the spring. The capacitive or piezoelectric components of these sensors can convert the motion of the mass into an electrical signal that measuring instruments can interpret. As a result of the widespread use of accelerometers in civilian applications, new types of sensors capable of making measurements in the most diverse ways have been developed; many sensors work in plane, that is, they are designed to be sensitive on only two Cartesian axes; therefore, to create a triaxial sensor (sensitive to all three axes), two sensors are combined, one perpendicular to the other.

On the other hand, gyroscopes use the Coriolis Effect in place of acceleration to measure angular velocity, or how quickly the body is turning. The gyroscope is a rotating device composed of a circular disk mounted on a system that allows the axis of rotation to move freely. In accordance with the law of conservation of angular momentum, the orientation of the disk's axis remains parallel and resists any attempt to change it when the disk is in rotation. Gyroscopes do not report the current angle; instead, they report the speed at which the object is turning. The motion along the axis that accelerometers and gyroscopes can detect is seen in the Figure 3.3.

Currently, gyroscopes have a wide range of applications, including automatic guidance systems for aircraft, missiles, and submarines, stedicams used to stabilize movie cameras, and inertial guidance systems for satellites. All rapidly rotating devices, such as flywheels and computer hard drives, exhibit the gyroscopic effect, which must be accounted for during design. Other examples of devices are the Inertial Measurement Units, or IMUs, typically contain accelerometers and gyroscopes. It is a set of sensors, including temperature sensors, magnetometers, accelerometers, and gyroscopes. Using a technology known as MEMS, or micro-electromechanical system, all these sensors are implemented on a microscopic level.

3.2.2 Data Processing Workflow

Since the raw data collected for posturographic analysis is the trajectory of the *COP* in two dimensions (AP and ML), processing and analyzing data collected from smartphone-integrated accelerometers can allow one to directly estimate the sway of the center of gravity (i.e., its trajectory along the AP and ML axes as shown in Figure 3.4).

Figures 3.5 depict the data processing applied to smartphone data (a) and force platform data (b) to obtain comparable signals.

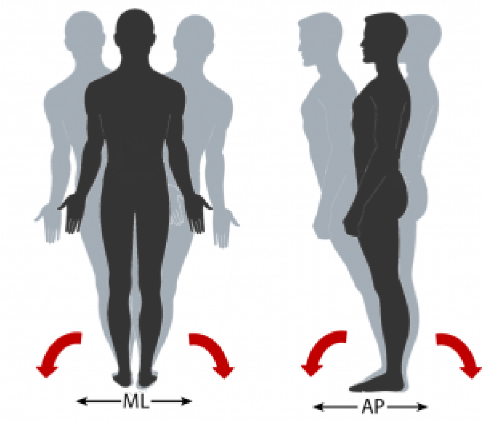
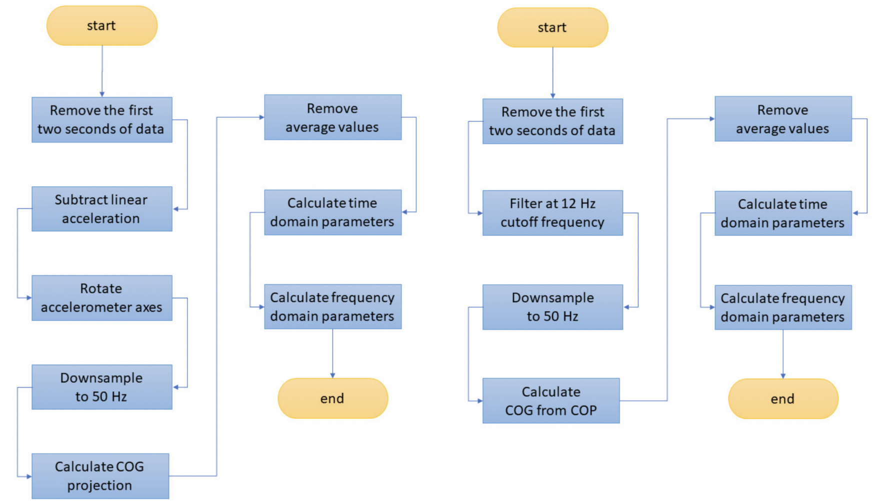


Figure 3.4: Body Oscillations Performing Static Tests

A force platform (Figure 3.5b) records the *COP* components along the AP and ML axes directly. After removing the pre-settling time (2 seconds), these two components are filtered with a Butterworth low-pass filter of 2nd order with a cutoff frequency of 12.0 Hz and downsampled at a frequency of 50 Hz. The *COP* is then applied to the inverted pendulum model to estimate the *COGv* trajectory. Also, in this instance, any possible baseline drifts are eliminated by subtracting the average values, followed by the calculation of the time and frequency domain features.

In a smartphone (Figure 3.5a) the components are recorded differently. As a pre-settling time, two seconds are subtracted from the beginning of the record for smartphone and force platform data. As proposed by Van Hees et al. [110], smartphone data are filtered with a 4th-order Butterworth low-pass filter with a cutoff frequency of 1.0 Hz to isolate gravitational acceleration. Then, a tilt axis correction is applied to the gravitational components by rotating the smartphone's accelerometer reference axes until the average value of each gravitational component, namely gx, gy, and gz, corresponds to perfect vertical positioning.

To maximize the average value of the gravitational components along the vertical axis y, a rotation is applied relative to the origin of the reference axes. The tilt axes correction is required to compensate for the smartphone's possible misalignment with respect to the body axes. This


start
Remove the first two seconds of data
Subtract linear acceleration
Rotate accelerometer axes
Downsample to 50 Hz
Calculate COG projection
Remove average values
Calculate time domain parameters
Calculate frequency domain parameters
end
start
Remove the first two seconds of data
Filter at 12 Hz cutoff frequency
Downsample to 50 Hz
Calculate COG from COP
Remove average values
Calculate time domain parameters
Calculate frequency domain parameters
end

(a) Smartphone Data (b) Force Platform Data

Figure 3.5: Data Processing Workflow

correction is necessary because in self-diagnosis applications, the user may lack the necessary skills to correctly orient the smartphone, which could compromise the accuracy of the analysis.

The subsequent processing step involves downsampling the recorded data to 50 Hz in order to conform to the typical working conditions of a stabilometric analysis. After that, the smartphone data are prepared for processing. The final steps eliminate any possible baseline drifts by subtracting the average values from the AP and ML components, followed by the calculation of the time and frequency domain features as described in the literature [14, 75].

3.3 Data Collected and Protocol

Typically, two types of charts are generated during a stability check:

- The Statokinesigram, also called the sway-path, depicts the shift in COP in the x, y plane

- The Stabilogram shows the change in COP over time.

Table 3.1: Data Collected through the Smartphone

Symbol	**Dimension**
Time Domain Features	
SWP	mm/s
SWA	mm^2/s
$DIST$	mm/s
$STD_{AP,ML}$	mm/s
R	mm/s
AR	$adimens.$
$FP_{AP,ML}$	Hz
$FM_{AP,ML}$	Hz
$F80_{AP,ML}$	Hz
Structural Features	
NP	$unit$
MT	s
ST	s
MD	mm
SD	mm
MP	s
SP	s
Gyroscopic Features	
$GR_{x,y,z}$	$degrees/s$
$GM_{x,y,z}$	$degrees/s$
$GM_{x,y,z}$	$degrees/s$
$GV_{x,y,z}$	$degrees/s$
$GK_{x,y,z}$	$adimens.$
$GS_{x,y,z}$	$adimens.$

The COP is expressed as a vector in two dimensions: Antero-Posterior (AP) and Medial-Lateral (ML).

Several valuable parameters can be extracted from the Statokinesigram. These parameters can be divided into two categories: *global parameters* and *structural parameters.* The former examines the sway pattern as a whole, whereas the latter breaks down the trajectories into smaller chunks and extracts useful data from them.

In order to develop a smartphone application capable of determining posture, the COP data are derived from the accelerometer values in the device, focusing the attention on the features to be computed as follows:

- Features in the time domain
- Features in the frequency domain
- Structural features
- Gyroscopic features

Time Domain Features Features in the time domain contain all the parameters derived from the study of sway-path behavior over time:

- *Sway Path*: length of the COP trajectory over time
- *Mean Distance*: mean distance from the center of the COP
- *Standard Deviation of the Displacement*: standard deviation of the total displacement of the COP
- *Range*: maximum distance between two points of the COP.

Frequency Domain Features Features in the frequency domain are derived from the frequency of the sway path. Their main purpose is to have an estimate of the sway-path energy and how it is concentrated in the various frequencies:

- *Frequency at 80%*: frequency band that contains 80% of the frequency in the AP and ML spectrum
- *Mean Frequency*: average of the frequency in the AP and ML spectra
- *Frequency Peak*: the peaks of the frequency in the AP and ML spectra

Structural Domain Features Structural features study the sway density curve (SDC) defined as the curve, time-independent, that for each time instant counts the number of consecutive samples of the statokinesigram within a circle of a given radius. The indicators are as follows:

- *Number of Peaks*: average number of peaks in the SDC
- *Mean Time*: mean of the time distance between two peaks in the SDC
- *Standard Time*: standard deviation of mean time
- *Mean Peak*: mean duration of peaks in the SDC
- *Standard Peak*: standard deviation of mean peaks
- *Mean Distance*: mean spatial distance between two peaks of the SDC
- *Standard Distance*: standard deviation of the mean distance

Gyroscopic Features Unlike the previous features, gyroscopic features study different parameters starting from the gyroscope data. They are summarised as follows:

- Mean: mean value of the gyroscope signal in the x, y, z axes
- Range: range of the gyroscope signal in the x, y, z axes
- Variance: variance of the gyroscope signal in the x, y, z axes
- Kurtosis: kurtosis index of the gyroscope signal in the x, y, z axes
- Skewness: skewness index of the gyroscope signal in the x, y, z axes

3.3.1 Anonymized anamnestic data

The application is designed in such a way as to guarantee complete anonymity to the user; because of regulations around health data, the application process everything privately within the user's device and interact with an external database only through anonymized data.

Although completely anonymous, the user is asked for some personal information: the main one is his height; it is used by the algorithm that extracts *COGvs* starting from the accelerometer and is essential for correct posture estimation. The remaining nonmandatory personal data are as follows:

- Age

- Gender
- Weight
- Postural Problems
- Presence of Postural Problems in the Family
- Assumptions of Drugs that may Interfere with Posture
- Traumas
- Visual Defects
- Auditory Defects

3.4 Balance - Postural Stability ©

Balance - Postural Stability ©, or simply Balance, is a smartphone application developed by researchers at the University of Urbino to measure postural stability. Thanks to the signals generated by the accelerometer and gyroscope, which are present on smartphones, it is possible to capture the dynamics of a subject's balance, obtain valuable information about the health status of the postural stability.

While Balance is a useful tool for monitoring postural stability, there are some challenges related to using mobile devices to collect data. One of the primary challenges is that mobile devices are not always held in a consistent manner, which can affect the accuracy of the data collected. Additionally, the imprecision of the accelerometers and gyroscopes in mobile devices can sometimes result in inaccurate readings.

Despite these challenges, the accuracy of Balance is considered to be adequate for most purposes. The app has been extensively tested and validated against other methods for evaluating postural stability, and the results have been consistently positive [68].

3.4.1 Development Framework

Balance has been developed in Flutter. Flutter is a Google-developed open-source framework for creating native iOS, Android, web, and desktop applications from a single Dart-based codebase.

The fundamental component of a Flutter application is the Flutter Engine. The engine manages the Dart virtual machine's life cycle, interfaces with the native SDKs of the platforms, and provides support for low-level rendering using the Skia graphics library, also developed by Google. The Flutter Engine's ability to perform a "hot reload" of the application, in which code changes are immediately published without the need for a complete reboot or state change, dramatically reduces waiting time during development and is a feature that is highly regarded.

Differentiating Flutter from other mobile development tools (Kotlin, Swift) is the reactive paradigm (or Reactive programming), which is based on data streams and change propagation. Flutter employs a declarative approach for graphics, with the main idea being that the UI is a function of the state, i.e., the user interface is reconstructed with each state change by applying certain Widget-defined functions. A Widget is responsible for describing the view's appearance based on its current configuration and state; when the state changes, the widget recreates the user interface with the new data. The widget can be viewed as the smallest element of programming in Flutter; in Flutter, everything is a widget, and applications are constructed by composing simpler widgets, one with the other, to produce a Widget Tree that describes the entire application.

3.4.2 Reading Sensors

The most important part of the application is the code related to reading the sensors to perform the evaluations. Flutter allows for the creation of the so called plugins, allowing retailed solution for both iOS and Android. The code written in the platform's development language (Kotlin/Java for Android, Swift/Objective-C for iOS) is native. This allows for greater performance and the use of platform-specific features.

Balance required to read the sensor results for an a priori configurable period of time with the ability to cancel it before the end and, most importantly, that the entire obtained sequence has a sampling rate of approximately 100Hz.

With these requirements in mind, we considered to use the plugin *sensors*, a native Flutter plugin designed to receive all sensors data. However, tests conducted at the time of development revealed that the plugin was incapable of producing sequences with sample rates greater than 8-10Hz, well below the target, so we customized the plugin to solve the issue.

Customizing plugin means creating the entry point of the dart-side code to a native flutter object that acts as a conduit for messages to and from native platforms.

3.4.3 Features Calculation

Once the data is collected, it is processed within the smartphone. The feature calculation algorithm consists of 3 steps:

1. The raw input is converted into lists for the axes of the accelerometer and gyroscope;

2. The *COGvAP* and *COGvML* values are calculated from the accelerometer data. Internally all the data are represented as matrices and then extracted into two lists at the end;

3. Features are computed using *COGv* and gyroscope values.

The entire algorithm is embedded within Flutter's native compute function, which allows the entire code to be executed inside a Flutter Isolate. Dart is a single-thread language, which means that instructions are executed one at a time; an Event Loop is used to achieve asynchrony. In case the is the need to perform very slow or heavy operations, the Flutter Isolate allows obtaining a different Event Loop in which to execute the de facto code in a parallel manner to the previous one. Isolates, as the name implies, are isolated memory spaces with which can exchange data only by messages.

3.5 Balance Centralized Data Management Architecture

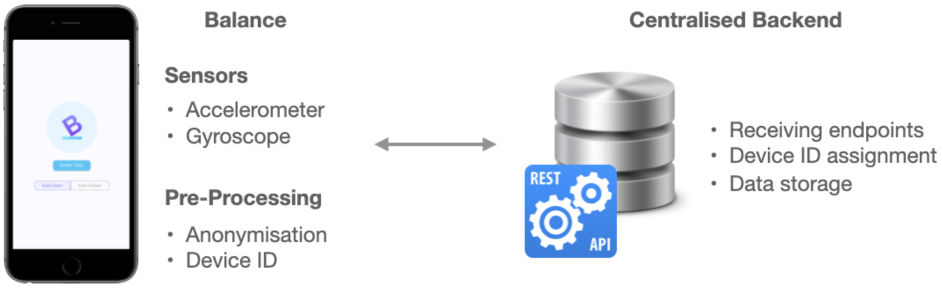


Figure 3.6: Balance Infrastructure Overview

Balance consists of a mobile application and a backend that resides in a centralized location, as shown in Figure 3.6.

The application allows for repeatable tests performed through the smartphone in a few seconds, and the appropriate mode (eyes closed or open) can be chosen on the home screen. Specifically, the analysis generates the *Sway Path* (SWP), which represents the projection of the displacement of the *Center of Gravity* (COG) over the floor. Usually it is also represented by its components in the antero=posterior (AP) and medial-lateral (ML) directions w.r.t. the body position.

Starting from the *SWP*, the Balance application automatically extracts global and structural parameters: the former belongs to methods whose aim is to estimate the overall size and features of the sway patterns, while the latter is based on the decomposition of sway trajectories into sub-units that can potentially be related by their role w.r.t. the underlying motor control processes [68]. The complete source code of the application can be found at:
https://github.com/ComputerScienceUniUrb/balance-mobile.

The smartphone application performs all the pre-processing onboard to comply with privacy regulations. For this reason, the user receives a unique token ID not associated with him or his device to refer to his data correctly. Moreover, as part of the required data, the user needs to provide some personal information such as height, age, gender, and the other anamenstic data such as any postural, hearing, vision problems, and some information about personal habits.

3.5.1 Backend Infrastructure

In app development, a critical, make-or-break stage is pushing to production or making an app production-ready. Specific configurations need to be done to ensure there are no breakages, such as security breaches, or exposing sensitive configurations.

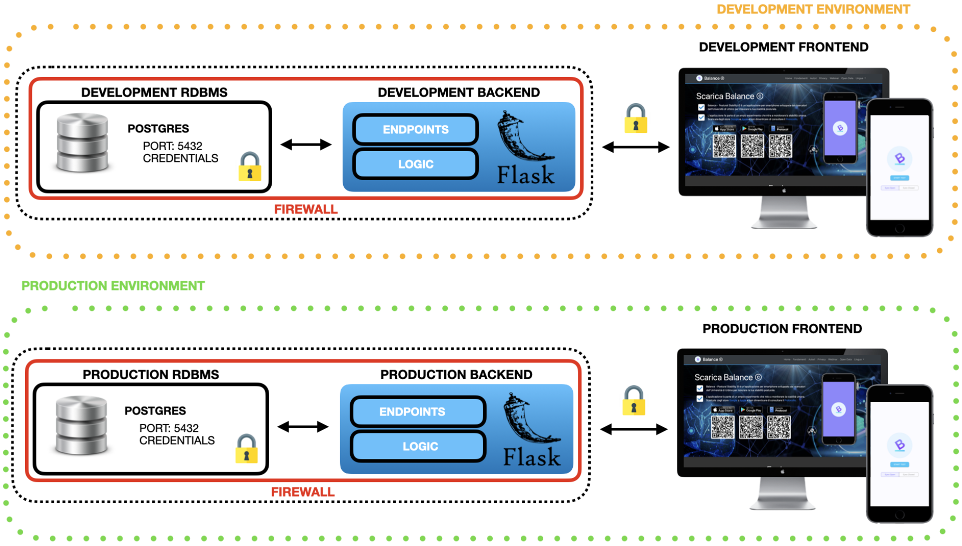


Figure 3.7: Balance Centralized Backend Infrastructure

The infrastructure of the project is divided into two main parts: *development* and *production*. The development environment is dedicated to building the application, while the production environment contains the stable version of the project.

Both the infrastructures are hosted in servers secured with a firewall and run the backend along with the database, which is able to communicate with the smartphone app, and the database and offers the web interface for data deletion later explained in this section. The service is written in Python because of the large number of libraries available and its versatility, including libraries for data manipulation of main importance for the project. The ability to manage data easily with Python eliminated useless parsing processes and improved data processing phases. The communication channels are secured with SSL and authentication mechanisms to strengthen protection.

The backend server includes all the components exposing services to users. Although the user cannot see any of the code, servers, networks, or databases, all of these elements work together in the backend design to determine the correct information to provide. To serve in production, we used the Flask microframework combined with the Django framework. This guarantee a stable and reliable web server gateway interface (WSGI) for receiving HTTP requests and a proxy server that can also act as a load balancer in the event the Django app receives heavy HTTP traffic.

The backend application is served through Docker, making it portable on all the machines running it. Using Docker containers, one for the Flask app and another for Nginx web server shipped together with Docker-compose, allowed for the quick setup of the application and disaster recovery.

3.5.2 Tech Stack

The technologies used for backend development are described in the following:

- **NGINX**: Open source software called NGINX serves as a web server, reverse proxy, cache, and video streaming. High performance and versatility are combined with its straightforward structure: NGINX is compatible with Unix, Linux, macOS, Solaris, and Windows and swiftly delivers static material without using the system's resources, allowing them to be used for other tasks.

- **gUnicorn**: Gunicorn is an application server that takes care of networking, asynchronization, and vertical scalability, specifically meant to help keep python applications alive. It is able to handle increased server load by having multiple workers or threads, and it is robust with failover.

- **Flask Microframework**: Flask is a micro-framework written in Python, based on the Werkzeug WSGI tool and with the Jinja2 template engine. It is distributed under a free BSD license. Flask is called referred as a micro-framework because it has a simple but extensible core. There is no abstraction layer for the database, form validation, or any other component to provide standard functionality for which third-party libraries already exist. At any rate, Flask supports extensions that can add functionality to an application as if they were implemented by Flask itself. For example, there are extensions for form validation, file upload management, various authentication technologies, and more.

- **PostgresSQL**: is a complete object-oriented DBMS released under a free license (BSD License style). Often abbreviated as Postgres, although this is an old name for the same project, it is a real alternative to both other free products such as MySQL, Firebird SQL, and MaxDB and closed-code products such as Oracle, IBM Informix or DB2 and offers unique features that place it at the forefront of the database industry in some respects.

3.5.3 API Endpoints

Even though the web had achieved great success in the ease of interaction between users and services, there were still significant challenges in the interaction inside dispersed systems talking over the network prior to the introduction of Web services. The World Wide Web Consortium (W3C) defines a Web Service as a “software system designed to allow interoperability between various machines on the same network or in a distributed environment”.

The idea of an API (Application Programming Interface) is comparable to that of a Web Service. However, instead of allowing for straightforward communication between two machines connected by a network, a Web Service serves as an interface for other types of applications. An API offers an abstraction that makes it straightforward to use by shielding the programmer from understanding how “what lays beneath” functions. One or more endpoints that are accessible to request and response messages make up these APIs.

Each APIs provide its endpoints. They must be static entities that specify the location of a specific resource so that it may be queried. They are typically reachable through URIs that receive HTTP queries. The endpoints developed for Balance's backend operation are as follows:

- **POST /sway**: allows connected devices to store raw data originating from the calculation of the mobile application
- **POST /measurement**: allows connected devices to store the measurement data originating from the processing of raw data collected by the smartphone sensors
- **POST /system**: the endpoint collects information on connected devices for development purposes
- **POST /signup**: the endpoint is used in the very first instance to issue the unique token through which the user can link back to his health data

3.5.4 Security of Stored Data

Data is collected and stored according to the GDPR non-confidentiality rules. The actions taken are described below.

- **Authentication Token**: Authentication is the act of confirming the claims made by or about the subject are true and authentic. It serves vital functions within any organization: securing access to corporate networks, protecting the identities of users, and ensuring that the user is really whom he is pretending to be. The information authentication can pose special problems, especially man-in-the-middle (MITM) attacks. Most cryptographic protocols include some form of endpoint authentication specifically to prevent MITM attacks. For instance, transport layer security (TLS) and its predecessor, secure sockets layer (SSL), are cryptographic protocols that provide security for communications over networks such as the Internet. In this work we use the Bearer token authentication (also called token authentication). Bearer token authentication (also known as token authentication) is an HTTP authentication system that uses security tokens known as bearer tokens. The bearer token is a string of ciphertext that is often created by the server in response to an authentication request. When making requests to protected resources, the client must include this token in the Authorization header.

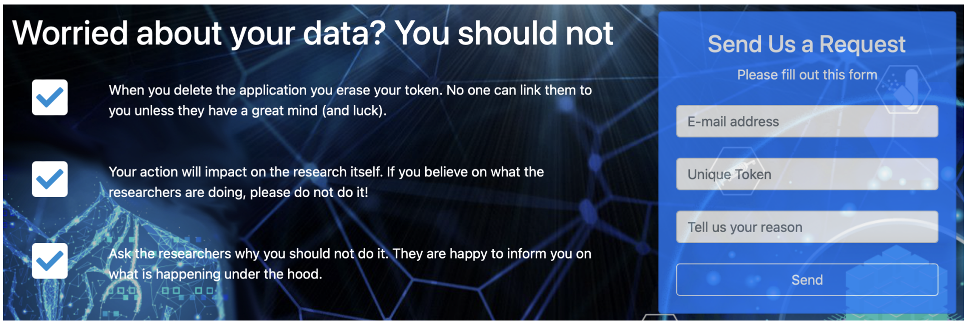
- **Encrypted SSL Channel**: Data encryption is an efficient means of preventing unauthorized access to sensitive data. Its solutions protect and maintain ownership of data throughout its lifecycle-from the data center to the endpoint. Encryption is useful to avoid exposure to breaches such as packet sniffing and theft of storage devices. Healthcare organizations or providers must ensure that the encryption scheme is efficient, easy to use by both patients and healthcare professionals, and easily extensible to include new electronic health records.

- **Data Anonimization**: Masking replaces sensitive data elements with an unidentifiable value. It is not an encryption technique, so the original value cannot be returned from the masked value. It uses a strategy of de-identifying data sets or masking personal identifiers such as name, social security number and suppressing or generalizing quasi-identifiers like date of birth and zip codes. Thus, data masking is one of the most popular approaches to live data anonymization.

3.5.5 Data Deletion Utility

Organizations are required by the General Data Protection Regulation to comply with requests from individuals to delete personal data, with the exception of the following circumstances.

The personal information that a business or organization is in possession of is needed to exercise the right to free expression, and it must



Worried about your data? You should not

☑ When you delete the application you erase your token. No one can link them to you unless they have a great mind (and luck).

☑ Your action will impact on the research itself. If you believe on what the researchers are doing, please do not do it!

☑ Ask the researchers why you should not do it. They are happy to inform you on what is happening under the hood.

Send Us a Request

Please fill out this form

E-mail address

Unique Token

Tell us your reason

Figure 3.8: Data Deletion Utility at Balance Mobile Website

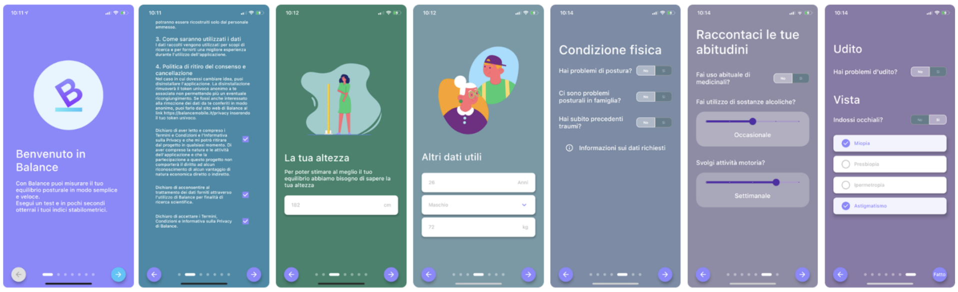
be kept for legal and public interest reasons (i.e. public health, scientific, statistical, or historical research purposes). If a business or organization processes data improperly, it must erase it. Data gathered about a person when they were still minors needs to be removed. The same rules apply to a request from a person whose personal information obtained when he was still a minor.

Regarding the "right to be forgotten" online, organizations are required to take reasonable steps (such as technical ones) to alert other websites that a specific person has asked for their personal information to be erased.

Additionally, data that have been appropriately anonymized may be kept. For Balance, we built a web interface through which the user can request deletion at any time, coupled with the centralized backend. The web platform allows users to view their rights, where their data is stored, and are given the ability to delete their data effortlessly on demand. The interface is shown in Figure 3.8.

3.6 Smartphone Application Demo

Onboarding Upon initial launch, the user is directed to an onboarding screen consisting of multiple pages. After welcoming the user to the application, he is prompted to enter his medical history using separate screens for each category. The historical data are categorized as follows:


10:11

Benvenuto in Balance

Con Balance puoi misurare il tuo equilibrio posturale in modo semplice e veloce.
Esegui un test e in pochi secondi otterrai i tuoi indici stabilometrici.

10:11

potranno essere ricostruiti solo dal personale ammesso.

3. Come saranno utilizzati i dati

I dati raccolti vengono utilizzati per scopi di ricerca e per fornirti una migliore esperienza durante l'utilizzo dell'applicazione.

4. Politica di ritiro del consenso e cancellazione

Nel caso in cui dovessi cambiare idea, puoi disinstallare l'applicazione. La disinstallazione rimuoverà il token univoco anonimo a te associato non permettendo più un eventuale ricongiungimento. Se fossi anche interessato alla rimozione dei dati da te conferiti in modo anonimo, puoi farlo dal sito web di Balance al link https://balancemobile.it/privacy inserendo il tuo token univoco.

Dichiaro di aver letto e compreso i Termini e Condizioni e l'Informativa sulla Privacy e che mi potrò ritirare dal progetto in qualsiasi momento. Di aver compreso la natura e le attività dell'applicazione e che la partecipazione a questo progetto non comporterà il diritto ad alcun riconoscimento di alcun vantaggio di natura economica diretto o indiretto. ☑

Dichiaro di acconsentire al trattamento dei dati forniti attraverso l'utilizzo di Balance per finalità di ricerca scientifica. ☑

Dichiaro di accettare i Termini, Condizioni e Informativa sulla Privacy di Balance. ☑

10:12

La tua altezza

Per poter stimare al meglio il tuo equilibrio abbiamo bisogno di sapere la tua altezza

182 cm

10:12

Altri dati utili

26 Anni

Maschio

72 kg

10:14

Condizione fisica

Hai problemi di postura? [✓ No] Sì

Ci sono problemi posturali in famiglia? [✓ No] Sì

Hai subito precedenti traumi? [✓ No] Sì

Informazioni sui dati richiesti

10:14

Raccontaci le tue abitudini

Fai uso abituale di medicinali? [✓ No] Sì

Fai utilizzo di sostanze alcoliche?

Occasionale

Svolgi attività motoria?

Settimanale

10:14

Udito

Hai problemi d'udito? [✓ No] Sì

Vista

Indossi occhiali? No [✓ Sì]

☑ Miopia

○ Presbiopia

○ Ipermetropia

☑ Astigmatismo

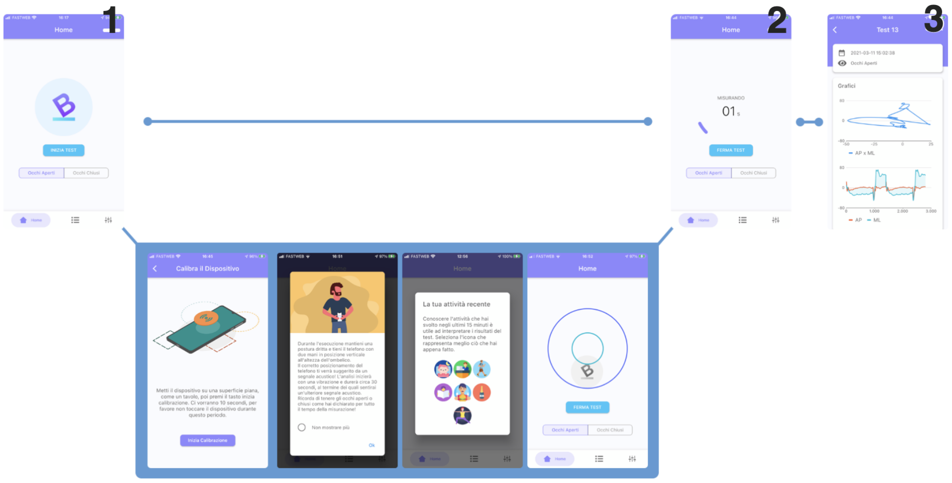
Figure 3.9: Balance Onboarding Screens Demo

- Personal information
- Habits details
- Physical condition
- Traumatic history
- Visual/Auditory defects

Measurements Page and Measurement Protocol After the initial launch, every time the user opens the application, he will be greeted with the *Measurement Page*; this is to provide quick access to the primary functionality, which is to create a new test. The user is required to calibrate his or her smartphone at least once on this screen in order to adjust the accuracy of the sensors and eliminate any errors in factory targeting or manufacturing defects. The 10-second calibration process begins when the “Start Calibration” button is pressed. To begin the measurement, the user must press the “Start Test” button, after which a 5-second timer will appear, which is helpful for getting into position before the actual sensor measurement begins. The process is described in Figure 3.10

In conventional systems, the subject is positioned in the middle of a force platform for postural acquisition in order to perform the necessary test. In this thesis the focus is on replicating the Romberg test with smartphones, which is a test in which the patient performs the analysis with both eyes open or closed, revealing the effect of the visual system on posture.

In a few seconds, the repeatable test is conducted through the smartphone using Balance. By selecting the appropriate mode on the home screen, the test can be taken with the eyes open or closed. The user is instructed to maintain a straight posture and holds the smartphone with two hands in a vertical position at the navel level while performing a test. Actual sensor measurement requires approximately 30 seconds. In order for the analysis to be meaningful, the test must be periodically repeated. Thus, the evolution of postural stability can be monitored by comparing historical indices to the most recent ones.



Metti il dispositivo su una superficie piana, come un tavolo, poi premi il tasto inizia calibrazione. Ci vorranno 10 secondi, per favore non toccare il dispositivo durante questo periodo.

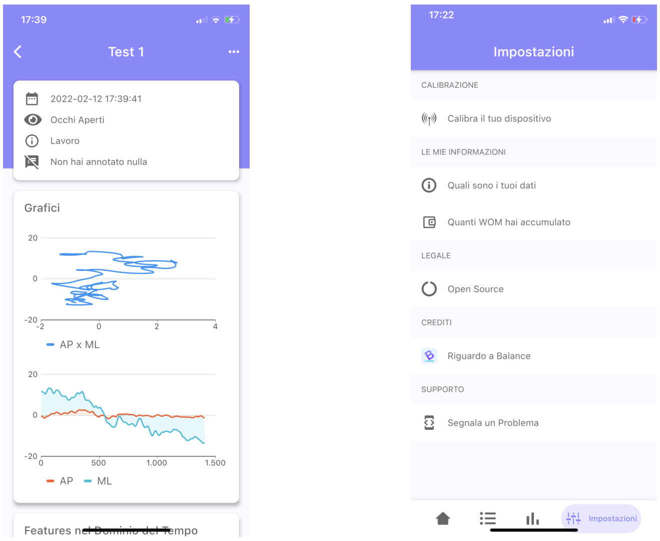
Durante l'esecuzione mantieni una postura dritta e tieni il telefono con due mani in posizione verticale all'altezza dell'ombelico.
Il corretto posizionamento del telefono ti verrà suggerito da un segnale acustico! L'analisi inizierà con una vibrazione e durerà circa 30 secondi, al termine dei quali sentirai un'ulteriore segnale acustico.
Ricorda di tenere gli occhi aperti o chiusi come hai dichiarato per tutto il tempo della misurazione!

Conoscere l'attività che hai svolto negli ultimi 15 minuti è utile ad interpretare i risultati del test. Seleziona l'icona che rappresenta meglio ciò che hai appena fatto.

Figure 3.10: Balance Measurement Workflow Demo

Results and Utilities The user can view the history of all performed tests on the appropriate page, where each test is listed in a chronologically ordered list. The user can view the outcomes of a specific test; the page is divided into the following sections: The first card contains general information about the test (date and whether it was performed with eyes open or closed); the second card contains the statokinesigram and stabilogram graphs; and the remaining cards contain the values of the characteristics seen in the preceding sections.

A settings page is displayed with the device calibration, a summary of personal data, details about the application's dependencies, and additional information about the application.



17:39

Test 1

2022-02-12 17:39:41

Occhi Aperti

Lavoro

Non hai annotato nulla

Grafici

Features nel Dominio del Tempo

17:22

Impostazioni

CALIBRAZIONE

Calibra il tuo dispositivo

LE MIE INFORMAZIONI

Quali sono i tuoi dati

Quanti WOM hai accumulato

LEGALE

Open Source

CREDITI

Riguardo a Balance

SUPPORTO

Segnala un Problema

(a) Result

(b) Settings

Figure 3.11: Balance Results and Settings Screens Demo

Chapter 4

The InterPlanetary Health Layer

Traditional health systems primarily rely on self-managed infrastructures. These centralized solutions are inaccessible to external stakeholders, lack transparency, and require periodic maintenance.

Collecting data through centralized entities affects accessibility and availability because the provider is solely responsible for regulatory compliance. When multiple healthcare institutions need to collaborate, they typically agree to share and outsource health data to cloud computing services for medical practices, i.e. Trojano et al. [108] describe disease registries as shared tools for collecting and analyzing personal and clinical health data. However, these solutions only promote the sharing of data in specific agreements.

The paradigm shift towards decentralization should improve the efficacy of healthcare institutions and increase the well-being of individuals. Distributed ledgers (DLTs) and storages (DFSs) are promising technologies that could lead to decentralized healthcare. Bitcoin [81], and Ethereum [22] are examples of DLTs, specifically public blockchains. Examples of distributed storages include the IPFS, a peer-to-peer (P2P) network for storing large amounts of information with decentralized fault tolerance, or Solid, a project led by Tim Berners-Lee at MIT that provides secure storage for data exchange between two endpoints on the internet [99].

The combination of these technologies could result in secure, decentralized data layers with global reach. This section describes the envisioned decentralized data layer called the InterPlanetary Health Layer (IPHL), which could be helpful to the global healthcare industry. The contributions of this chapter are summarized as follows:

- We provide a new shared, agnostic, and permissioned decentralized data layer with enhanced data availability. We use decentralized technologies for this purpose: a DFS layer as a medium of storage, an experimental DLT to provide smart contract functionality and tracing capabilities, smart contracts to manage access policies, and authentication mechanisms to manage user data;

- We implement the proposed architecture on a real-world use case represented by a traditional IoMT application connecting to the IPHL implementation;

- We provide experimental results of the work, demonstrating the feasibility of such an implementation;

- We propose the development of a social network on top of the IPHL for the dual purpose of increasing the availability of data and the accountability of individuals in maintaining the system.

4.1 Reasons behind the IPHL

The need for more transparency in how data is collected, stored, and utilized by various services and businesses is the primary cause of the subsequent interest in data ownership. Christl et al. [24] tried storing data in inaccessible and disconnected data lakes highlighting how this makes them inaccessible to the public for innovation. In this regard, little effort has been expended to simplify data management so that a person can comprehend and manage the risks associated with the exploitation of his private data.

Through personal devices, individuals can make significant contributions to science, particularly in the field of personalized medicine [88, 18]. Unfortunately, sharing information without the individual's explicit permission is a serious violation of their rights. Regulations such as the "General Data Protection Regulation" (GDPR) by the European Union help promote a pro-individual perspective. Specifically, these regulations impose a number of accountability measures on actors responsible for processing personal data and grant individuals rights. However, these do not always address the lack of transparency in the management of personal data and the technical ability to make personal data portable, i.e. De Hert et al. [30].

A vision of involving the individual in the flow of personal data could be realized through the creation of a user-centered framework for managing personal data. One of these is that individuals have control over their data assets and that businesses comply with the law. Data protection and security can be achieved by decoupling file storage, access control mechanisms, and application logic. This would pave the way for individuals' privacy needs and have a significant impact on the capabilities the healthcare industry could strive for, as well as a unique common data lake and market [39], which capitalizes on data interoperability for the social good [45, 116, 118].

With the emergence of the first proposals to use DLT-based systems outside of finance, i.e. digital currencies, some researchers such as Zyskind et al. [122] have already discovered a link between DLT and the sharing of personal data. In this context, the general approach is to store access control policies on DLTs in a secure manner so that the applicant can be made aware of his permissions to access personal data stored outside the DLT. Using a (t, n)-threshold scheme, Yan et al. [113] presents a Personal Data Store (PDS) that enables the collection, storage, and fine-grained access to their data. Their solution of sharing personal information in fragments was innovative but costly and non-GDPR compliant, i.e., the system stores personal data directly on the DLT, which is not GDPR compliant. Programming access control policies as smart contracts to manage control automatically in compliance with GDPR is an alternative approach [29].

Koscina et al. [64] enable the exchange of healthcare data via a distributed architecture, with a focus on the consent provided via smart contracts. In their system, users maintain a digital copy of their medical information in a personal data account that can be hosted by any cloud-based data management service. By interacting with smart contracts and choosing who and for what purpose, users can personalize their consent preferences, which are dynamic. Data transactions are always auditable and GDPR compliant. In addition, it is intriguing that Onik et al. [92] emphasized the breaking points between the GDPR and DLTs. They propose a model that stores personal data off-chain and tracks its life cycle via data processors and processors using a DLT.

Other researchers, such as Nguyen et al. [88], Madine et al. [73] introduce the use of mobile devices, such as smartphones, and reputation systems to encrypt and share data. In other examples, the emphasis is placed on the effective compliance of these systems and the possibility of providing an identity to participants anticipating a health digital identity system [91]. Other solutions provide an economic incentive for those who share their health data [43], as doing so contributes to a greater body of knowledge for personalized medicine.

So there is a clear need to achieve a shared architecture in which users can create and manage their own data, while complying with privacy regulations and enabling innovation.

4.2 State of the Art

In the IoMT, researchers attempt to envision systems where users can effectively store and own their data. The primary technology of choice for the DFS is generally IPFS or cloud storage, while the DLT could vary by the requirements required by researchers, from public blockchains to permissioned DLTs. The investigations often converge on topics such as:

- Overall system decentralization;
- Data authentication;

- Data scalability.

The first attempts in the field were mostly related to the healthcare institutional domain. Jiang et al. [59] pointed out how the direction of health information exchange should integrate blockchain-based systems. Along with them, several other researchers made the same assumption as Srivastava et al. [104], Seliem and Elgazzar [101], Uddin et al. [109], and Marangappanavar and Kiran [77]. The proposed visions focused on the possibility of connecting healthcare institutions to nimbly share EHR data, in a few cases including even medical mobile devices. Other visions focused on emergency management, such as the one proposed by Tantidham and Aung [107], or telemedicine by Kordestani et al. [63] and tracking of data flows occurring in the network as described by Nascimento Jr et al. [85]. Other works in the field only sometimes focus on long-lasting health data management but sometimes on short-lasting health data, like healthcare passports or clinical studies, as in the work of Omar et al. [91].

Slowly, some researchers identified how there is a need to overcome technological barriers. Several gave more relevance to scalability issues, such as Saweros and Song [100], Donawa et al. [32], Lücking et al. [71], and Cisneros et al. [26], privileging the usage of Directed Acyclic Graphs (DAGs). Other researchers such as Adlam and Haskins [2] and Fernandes et al. [41] gave more importance to the user's identification in the system and to the ability to set permissions on the ledger by using permissioned DLTs such as Hyperledger.

Moreover, researchers choose to decouple data from DLTs to avoid sensitive data being immutably stored on the ledger. Dwivedi et al. [34] proposed a solution by decoupling the data from the ledger and allowing users to have control but relying on centralized entities to receive an identity and register in the system. Similar works followed proposing solutions for EHR data management, such as the one proposed by Arul et al. [8], Garg et al. [46], Stamatellis et al. [105], Mani et al. [76].

So, researchers deepened several alternatives for linking institutions employing a DLT, trying to increase the scalability of these systems and the possibility of tracking data for provenance [73, 88, 112]. However, the field still has few concrete implementations available, and very few are related to the IoMT.

Kumar et al. [67][65, 66] move toward including data from the IoMT, trying to provide a solution for medical device storage and authentication that can prevent security and privacy obstacles in the IoMT. The solution, called MedHypChain, is built on Hyperledger Fabric and focuses on authenticating and authorizing patient data while protecting the dissemination of medical device information in the blockchain network. Medical devices in the IoMT generate data transmitted to the blockchain network. The IPFS cluster is responsible for facilitating patient synchronization and providing secure information storage, while smart contracts are used to achieve consensus. Another proposal come from Egala et al. [35] proposed an architecture for the IoMT with encryption methods that intend to provide privacy, security, traceability, low latency, low storage cost, and data availability. To achieve decentralized EHRs, they maintain a public ledger for medical records and critical events to provide traceability. Smart contracts are used to help medical professionals perform event-based automation tasks without human interference. Similarly, Abdellatif et al. [1] propose the MEdge-Chain, which includes several e-health entities whose role is to monitor, promote and maintain people's health. The blockchain architecture is a consortium-based multichannel architecture that enables secure access, processing, and sharing of medical data among different electronic health entities.

4.3 The InterPlanetary Health Layer

The IPHL ensures data accessibility, availability and involves users in the process of data sharing, leveraging the DLT and smart contracts to manage the information stored in the DFS efficiently.

In the system, data is never shared with remote users without permission and are subject to verifiable transactions and access mechanisms coded in smart contracts. The user can create a private network that we called *Halo Network* to increase the availability (discussed in Chapter 5) and sharing of data. Along with the architecture, we present the decentralized version of Balance that interacts with the network and demonstrates its practical application in a decentralized domain.

Our focus is making the user directly responsible for the network maintenance instead of relying solely on known entities for decentralization. In fact, we found how nearly all works provide potential IoMT solutions without regard for how they might involve the same user in the decentralization of the system, assuming that this is likely to be held by those who wish to create the network, such as central authorities.

This layer's implementation is based on a DLT network constructed using the experimental IBM Hyperledger framework. The implementation takes into account the current technological limitations of mobile devices, namely their computing and storage capacities, and serves as the foundation for our vision of secure decentralized data layers.

The InterPlanetary Health Layer is proposed as a decentralized, international, shared, agnostic, and data lake containing IoMT data. The IPHL should enable individuals to act as their own data administrators. Users, as data creators, are the ones who have control over data while we consider data consumers all the participants interested in retrieving data.

The requirements of the InterPlanetary Health Layer (IPHL) should be: *(i) Data Accessibility and Availability*: The IPHL must ensure that data is accessible and available to all interested parties; *(ii) User Involvement*: The IPHL must involve users in the process of data sharing; *(iii) Verifiable Transactions and Access Mechanisms*: The IPHL must use smart contracts to manage the information stored in the DFS efficiently, ensuring that data is subject to verifiable transactions and access mechanisms; *(iv) Permission-based Sharing*: The IPHL must never share data with remote users without permission; *(v) User Control*: The IPHL must enable individuals to act as their own data administrators, giving users control over data.

The general framework of the system is described in Figure 4.1:

- **IPHL**: individuals can exchange information without relying on a central authority to store the data. The network enables private and secure data-sharing communication channels and allows

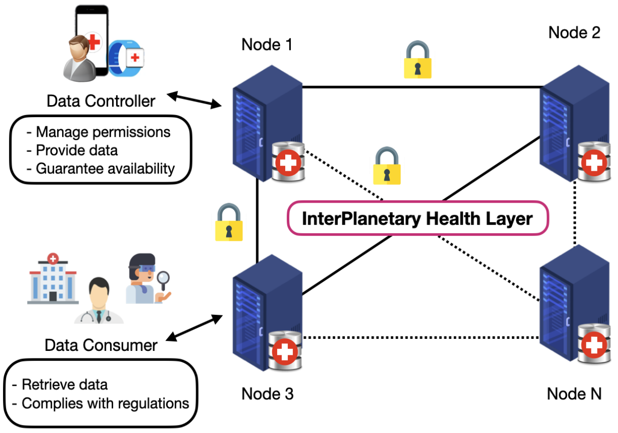


Figure 4.1: Conceptualizing the InterPlanetary Health Layer

individuals to manage data permissions. In addition, network participants should act as operators or maintain the network in the event of a failure. Participants in the network could be anyone interested in sharing IoMT data, including medical device owners, researchers, physicians, and other healthcare professionals.

- **The Data Provider**: users of IoMT devices, such as smartphones and smartwatches that function as medical devices, interact with the IPHL by supplying data and managing permissions. They can interact with nodes (possibly their own nodes) and manage the data made available by their applications. They can be citizens and ordinary individuals.

- **The Data Consumer**: remote users such as researchers, physicians, health research institutions, universities, and health professionals may be included. They may be interested in collecting data from IoMT devices and may interface with the IPHL network. They interact with nodes by activating data access mechanisms and kicking off the process of sharing.

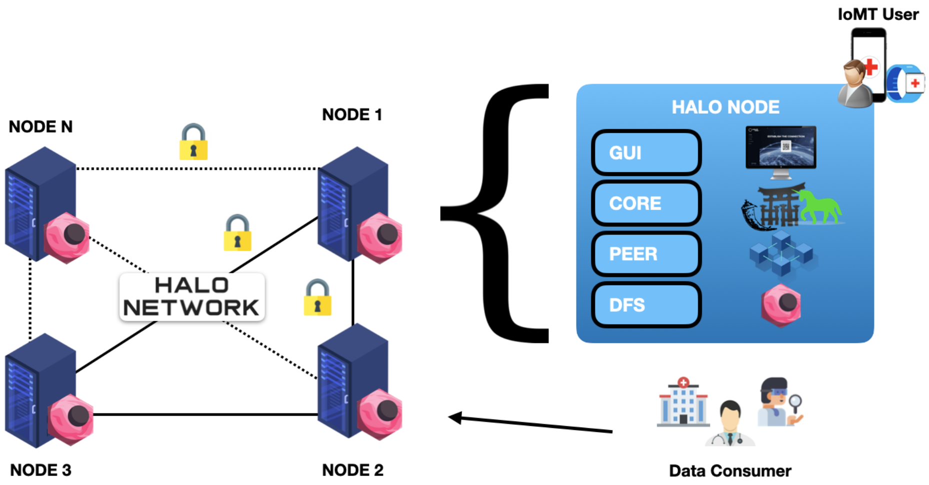


Figure 4.2: Halo Network Architecture

4.3.1 Implementation: the Halo Network

We called the proposed IPHL implementation the *Halo Network*, as described in Figure 4.2.

Two layers comprise the Halo network:

- The DLT Hyperledger Fabric (HLF), the experimental platform built by IBM and developed in collaboration with the Linux Foundation;
- The DFS called InterPlanetary FileSystem (IPFS). Since the DLTs expose immutable information, which is never advisable for personal data, even with private and permissioned ledgers, an available alternative approach is to store the data off-chain.

Following this link will lead to the network's open-source source code: *https://github.com/BigG-DSC/fabric-network*. In what follows we describe the *Halo Node*, key component of the architecture described in Figure 4.2 along with the implemented smart contract.

The Halo Node

In the following is a description of the *Halo Node* implementation, followed by a discussion of the Chaincode. The *Halo Node* acts as an access

point to users, allowing them to participate in the *Halo Network* and manage the personal data submitted by the IoMT devices. The source code can be found by following this link:
https://github.com/BigG-DSC/halo-node.

The node can be divided into four main layers as described in Figure 4.2:

- **Graphic User Interface**: the visualization panel consists of a web interface designed as a single web page application. It is accessible on the installed machine and was designed to allow the easy management of the node by the user. It is built with Jinja on top of the Flask framework, serviced by a Web Server Gateway Interface. It lets the user establish the connection and easily retrieve updates from the node. The event notification service is triggered every time the user accesses the visualization interface and tries to establish a connection with the node. Among the enabled functions, it allows transferring data to the node, accepting requests, and participating in the voting session.

- **Core**: it contains the node's logic, and it deals with the GUI and both the DFS and the Peer. Whenever interactions in the GUI occur, the Flask server dialogues with the corresponding underlying DLT and DFS applying the logic required. The module was specifically thought for managing current implementations and enabling future works acting as a middleware for the different underlining technologies. This ensures that the new layers can be easily added and that all operations pass through it.

- **Peer**: it allows interaction with the Halo Network. It is a Hyperledger Fabric peer communicating with the rest of the network performing invocations to Chaincodes and a NodeJS application to allow easy interaction with the peer. The Peer consists of holding the shared ledger, ensuring immutability and transparency, and making it possible to store the history of transactions in the network. It is built by implementing the peer provided by the HLF and an application written in NodeJS employing the Fabric SDK. The code helps to easily interact with the DLT when requests come from the Core. As soon as a request is received, the Core can dialogue with the peer responsible for the readings and writings on the ledger.

- **DFS**: IPFS is a distributed storage that does not organize data for querying. To solve the problem, we use OrbitDB: a decentralized database built on top of the IPFS, independent and secure, that allows data to be stored in a distributed manner with many replicas. As discussed by Shapiro et al. [102], the replicas are constantly synchronized among all available peers and rewritten according to Conflict-free Replicated Data Types. Thanks to OrbitDB, IPFS-based applications have a register to consult to handle the IPFS as a database but distributed. In addition, it allows participants to set read and write permissions reflected in the ability not to disclose data stored. It is built by implementing the OrbitDB library with an application written in NodeJS.

The Chaincode

A Chaincode is created above each node to allow users to vote with respect to the management of their data. A Chaincode is an implementation containing several smart contracts available on each peer. The stored data structure is a list of lists that represent the Access Control System constituted by the Polls and the list of votes from participants. Its functionality is described below:

- **CreatePoll**: this operation is dedicated to inserting a new vote when it is requested by an external entity, such as a physician. Following such a request, the receiving node notifies, through the DLT, that a new request has been received and creates an entry in the ledger so that all nodes can cast a vote.

- **Approve/Decline**: the operation allows voting on the ballot. The operation allows the ledger to be updated with the vote of the considered user. Each user has an identity within the ledger and can write his vote within the smart contract. Based on the policy chosen for vote validation, the node interested in closing the vote will wait for all votes to be received, and only then can it do so.

- **ClosePoll**: the closing operation and an update performed by a participant at the time he or she has the opportunity to do so. It consists of confirming the outcome of the vote and then granting or not granting permission for access to the data by an external user.

- **GetPoll/GetAllPolls**: allows the retrieval of information saved on the ledger. This can be done by requesting a specific voting id or by recalling all votes.

The source code of the Chaincode can be found at this location: *https://github.com/BigG-DSC/fabric-contracts*. A more in-depth investigation with respect to smart contract implementation is made later in Chapter 5.1.2.

4.3.2 Decentralising Balance with the Halo Network

We modified the source code of Balance to integrate it with the *Halo Network*. Interactions with the network are based on the possibility that third parties can communicate with the peer target. Once a request is received, the peer collects it and notifies the mobile device in its first interaction.

The mobile device interacts with its node, can send its data, and is updated on external requests to vote and receive information about network participants. Figure 4.3 shows the GUI that interacts with the *Halo Node* and, consequently, the *Halo Network*. The available functions are:

- **Halo Network**: the function allows the user to see the users added to their network and who participate in maintaining the data by contributing to its availability.

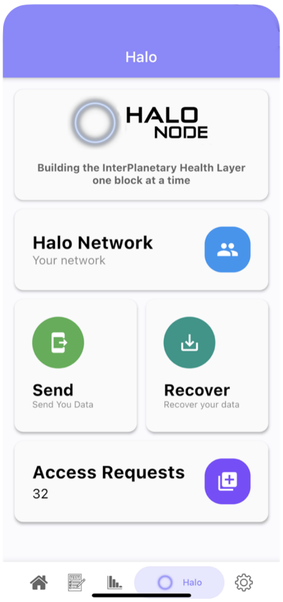


Figure 4.3: Halo Node GUI Integration on Balance

- **Send**: the function allows the user to transfer their data to their repository on IPFS through OrbitDB.

- **Recover**: the function allows the user to recover his data in the case he needs to, i.e. if the device is re-initialized.

- **Access Requests**: the function allows the user to keep track of all requests made by the network and vote on them while maintaining control over his data.

Finally, we can distinguish two primary workflows: the *Sharing Phase* and the *Storing/Retrieving Phase* are described below. The mock-up source code of the smartphone application can be found at the following location:
https://github.com/BigG-DSC/balance-decentralized.

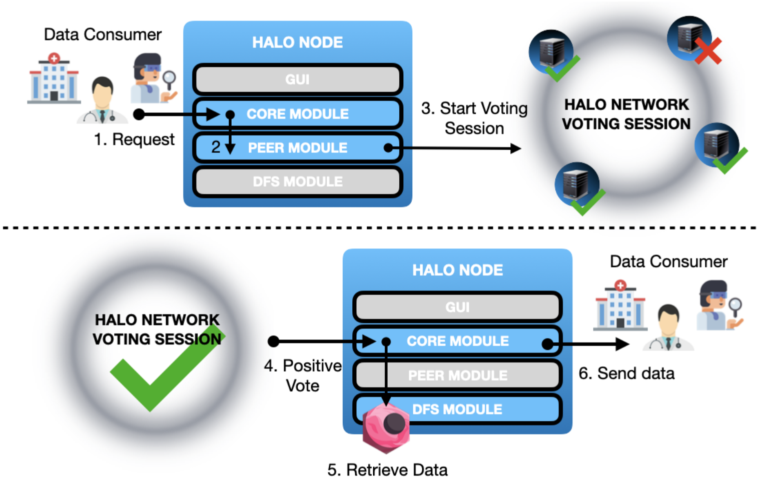


Figure 4.4: Halo Network Sharing Phase

The Sharing Phase

The *Sharing Phase* involves the interaction with the DLT and retrieving data from the distributed storage. It could be costly, including encryption techniques, and implies increasing response times. The workflow is described in Figure 4.4, and it involves the following steps:

1. The remote user makes the request;
2. The request is registered by the peer who received it on the DLT;
3. Voting begins on the authorization of the request;
4. When voting is closed, the remote user receives the requested information (if authorized), otherwise is declined.

The full sequence follows ten steps reported in the diagram shown in Figure 4.5:

1. A data consumer, i.e. a doctor, requests access to the receiving peer's data.
2. The node creates the poll by interacting with the DLT.
3. The DLT replies with and acknowledges.
4. The receiving node updates the mobile device regarding the request.

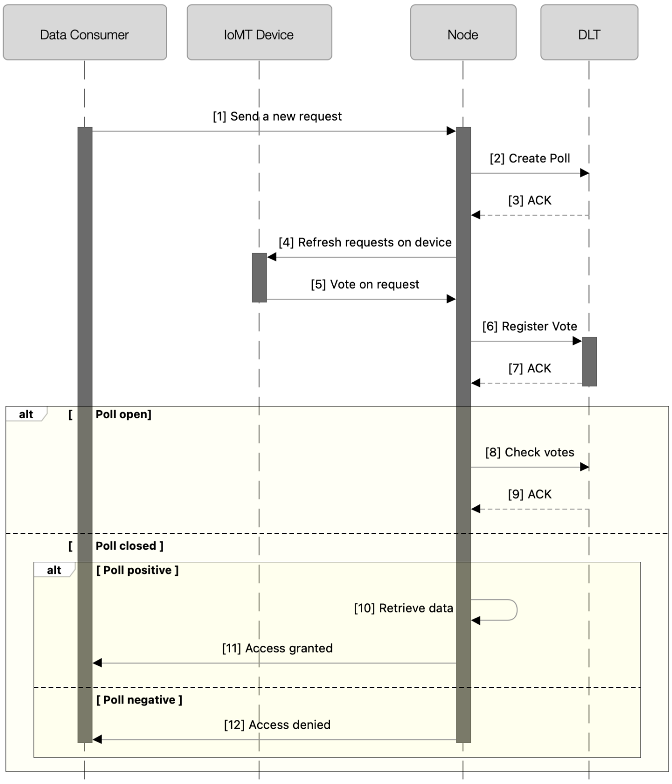


Figure 4.5: Halo Network Sharing Phase Sequence Diagram

5. The receiving user initiates the vote, sending its own to the node.
6. The node redirects the request to the DLT, recording the vote on the smart contract.
7. The DLT replies with and acknowledges.
8. After voting, the user keeps waiting for the votes of the remaining participants.
9. The DLT replies with and acknowledges.
10. When all the votes have been received, the poll is closed.
11. A positive outcome triggers the retrieval of the data, granting permissions to the remote user along with the requested data.
12. A negative outcome results in an access denied.

Halo Network Storing/Retrieving Phase

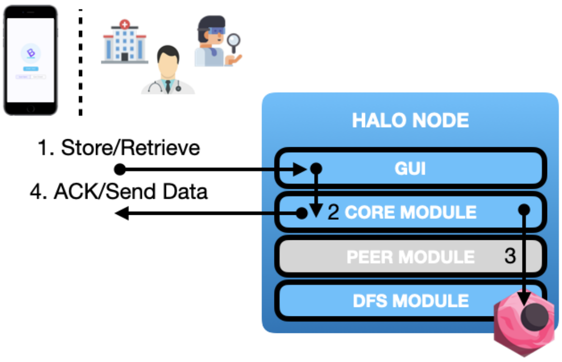


Figure 4.6: The Storing/Retrieving Phase

The *Storing/Retrieving Phase*, described in Figure 4.6, can both involve the interaction through the mobile medical device or a data consumer. Specifically, it is triggered when the IoMT application establishes the connection with the *Halo Node* to transfer newly collected data or a data consumer asks for data.

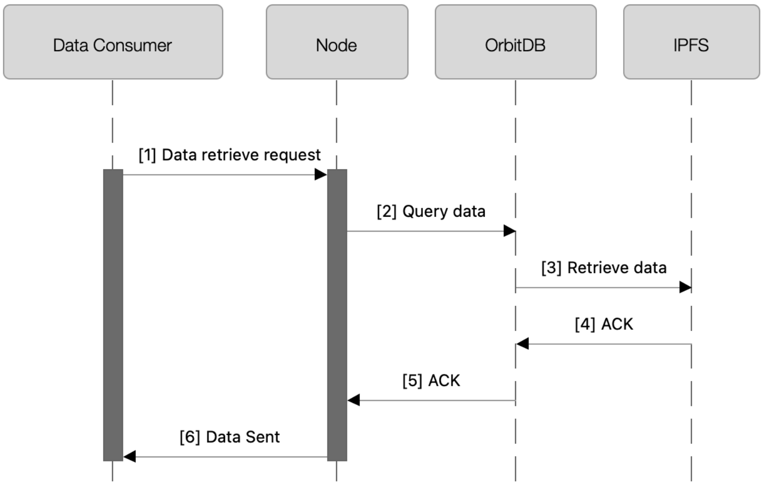


Figure 4.7: Halo Network Retrieve Phase Sequence Diagram

The sequence diagram describing the Retrieving Phase is highlighted in Figure 4.7. The Storing Phase is very similar and obtained by substituting the data consumer with the producer, sending the data to OrbitDB, storing it in the IPFS, and receiving the final acknowledgment as the last step 6. The Retrieving Phase considers the following six steps:

1. A request for storing data or retrieving is received, i.e. following a remote request or by the user himself attempting to store or retrieve data.
2. The Core sends the query to the decentralized database OrbitDB.
3. The data is retrieved from IPFS with the help of OrbitDB.
4. The IPFS acknowledges the operation to OrbitDB
5. The OrbitDB acknowledges the operation to the Core
6. The data is finally sent back to the requestor

4.4 Enabling Data Sharing

This section describes the decentralized access mechanism that conveys the practical method for individuals to safely share their data through the *Halo Network*.

The plan is to use cryptographic techniques to encrypt every piece of data stored in the DFS. The DLT network stores universal, immutable resource identifiers for data and provides smart contracts to ensure data integrity verifiability and manage Access Control Lists (ACLs) for each piece of data.

4.4.1 Embedding the Access Control Layer

The ACL is embedded in the *Halo Node*, as depicted in Figure 4.2. It address the specific case of health-related personal data and the architecture changes as follows:

- **The Halo Node** - the logic and APIs which a User Interface can exploit to interact with the other system components. The peers in the network verify and maintain the network's integrity by sharing a distributed ledger. This enables the storage of the entire history of transactions between peers and, consequently, requests made to the access mechanism. The nodes share the same access control keys and contribute to the verification process.

- **The Halo Network** - the underlying network of nodes that maintain the ledger that validates (through the untamperability property) data exchanged by the peers involved. At the heart of the architecture, the DLT network provides a peer-to-peer network with a distributed ledger that ensures the immutability and transparency of the records to be stored in the smart contract. The architecture enables interaction with a network that guarantees the immutability and transparency of the records that will be stored in the smart contract. It is essential to emphasize that this is a permissioned network in which consensus policies can be established. The decision is based on the fact that this network is also GDPR-compliant, as opposed to a public DLT that is transparent outside the participants and would allow external actors to view information. In general, as soon as a request is received, the core is able to communicate with the peer responsible for ledger readings and writings through the smart contract. Each time an operation is performed on the smart contract, it is reflected to the network peers that maintain the integrity of the distributed ledger.

- **The IPFS** - the component that deals with the actual storage of encrypted personal data. The DFS network stores and shares information, files, and directories in the form of content-identified objects (CID). This CID is generated when a hash function is applied to a piece of data, and it is also used to retrieve the data from the network. Once a piece of data has been published in the off-chain storage, i.e., the DFS, the returned CID can be used to retrieve it and verify its integrity. Thus, when a piece of data is initially uploaded into the system, it becomes a DFS object, which is then referenced asynchronously via its CID into a DLT. It would constitute the hash-pointing principle. If another network node attempts to share the same object, the CID will always be identical. Thus, in our system, encrypted health data are stored in a DFS and referenced in a DLT. Due to the fact that all data is encrypted at the User Interface/Core level, data security is maintained. Without introducing central trusted parties, this solution also improves performance and provides greater availability for data reads and writes [117].

- **The Access Control Layer** - a set of technologies and schemas that enable the access policies declaration (through smart contracts) and the actual data access (through keys distribution). The access control logic to share data is implemented via smart contracts. Through these, data access can be purchased or authorized by the owner. Therefore, only those users specified by the policies of a smart contract owned by the data subject are permitted to utilize the data. Due to the presence of smart contracts, there is no need for direct communication between the data owner and users interested in his data. In practice, each piece of data stored in the DFS is referenced in a specific smart contract by its CID or directory's CID. A straightforward policy would require the smart contract to maintain an Access Control List (ACL) that represents the rights to access one or more data sets. The remainder of this paper will focus on the implementation of such a policy. Once a user is permitted to access specific data, i.e., he is included in the corresponding ACL on a smart contract, he/she will also be permitted to obtain the key used to encrypt the data.

This ACL was introduced to preserve a set of principles:

(i) *Data Validation*: the integrity of data generated by (or on behalf) of users must be guaranteed and verified. To this end, the system takes full advantage of the untamperability property of DLTs.

(ii) *Traceability*: not only the integrity of personal data but also their life cycles must be guaranteed and verified. Also, in this case, the system takes advantage of DLTs and their smart contract features.

(iii) *Privacy-by-Design*: while we need to make it difficult to change or delete data, at the same time, it is needed to comply with regulations surrounding sensitive data, e.g., the General Data Protection Regulation (GDPR) [120]. The system requires the modification or deletion of data under certain circumstances as for the "right to be forgotten" [44]. This is one of the main breaking points between the DLTs and the GDPR that led to storing data off-chain.

(iv) *Data Protection*: cryptography plays a key role in the authenticity and integrity of the data and its treatment among all the agents in the data processing chain. For this reason, we refer to advanced cryptographic techniques [40] to verify the authenticity of data and to implement users' preferences in maintaining their privacy, i.e. authorized access.

In the following sections, we describe the applied cryptographic schemes.

4.4.2 Cryptographic Scheme

We provide a general overview of the cryptographic scheme without going into the details of the implementation in order to convey a clear understanding of the whole access control layer.

We refer to a hybrid cryptographic scheme making use of both asymmetric and symmetric keys. The general principle is that each piece of health data is encrypted using a symmetric "content" key k and then this key is encrypted using an asymmetric keypair (pk_{KEM}, sk_{KEM}). This consists of a Key Encapsulation Mechanism (KEM) [51] in which the key is encapsulated and the capsule is distributed.

Key Distribution Component

The presence of an off-chain key distribution component is necessary for two main reasons:

1. To free the owner of the data from the burden of managing the distribution of keys, which can be very costly in the case of fine-grained access;

2. To complement the public execution operations of smart contracts in the DLT, since it is not possible to independently release content keys or decrypt messages.

In the architecture, the DLT nodes are in charge of enforcing the access rights that are specified in the smart contracts ACLs. We take advantage of the high degree of trust that a DLT offers for the data written in the ledger, and therefore focus on the trust given to the entities that have to read this data and follow the correct policy. Indeed, DLT nodes rely on the ACLs to make so that the entitled data consumer can obtain the content key, and thus access the piece of data. In order to provide complete data protection to the data subject, only the entitled recipient of health data must obtain the key and not DLT nodes.

For this reason, we make use of a (t, n)-threshold scheme to share content keys among the network, and in particular, shares of the content key's capsule. When a data consumer with keypair (pk_c,sk_c) is entitled to access some data in a smart contract ACL, he requests the release of the associated capsule to some DLT nodes through a message signed with pk_c. Upon consumer request, the DLT nodes check if this one is entitled to through interaction with the smart contract. If this is the case, i.e. the data consumer is on the ACL, each DLT node starts the operation for releasing the part of the capsule that was shared with him previously by the data owner. Once the data consumer gets all the shares of the capsule, their reconstruction provides the key k needed to decrypt the desired data stored in the DFS.

We refer to a Threshold Proxy Re-Encryption (TPRE) scheme for the data capsule distribution. The capsule, initially obtained from the pk_{KEM} by the data owner, can be re-encrypted by each contacted DLT node using a re-encryption key $pk_{O \to C}$ generated by the owner. The re-encrypted capsule, then, can be decrypted using sk_c by the consumer to obtain the k_{DEM} needed to decrypt the data.

TPRE offers more guarantees rather than a simple PRE scheme that usually involves only one semi-trusted proxy node. One proxy node only can collude with the consumer to attack the data owner's private key. TPRE, instead, uses a (t, n)-threshold scheme to produce "re-encryption shares" in such a way that these can only be combined client-side by the data consumer and not by any $t - 1$ subset of proxies. We refer to the implementation of NuCypher [89, 36] for a decentralized scenario. PRE has the drawback of requiring the user to generate a re-encryption key $pk_{O \to C}$ for each new consumer. However, he has the option to stop producing new re-keys if some nodes are malicious.

Sharing Health Data

Table 4.1: Balance Health Data Summary

Health Data	
Sensitive Data	Measurement Data
Age	Stabilogram
Gender	Time Domain Features
Weight	Frequency Domain Features
Postural Problems	Structural Features

The proposed system for sharing health data enables the transaction of data between users and institutions while guaranteeing its provenance and immutability. Our architecture aims to prioritize decentralized authentication and authorization of health data for individuals. The DFS is responsible for facilitating data sharing and providing secure information storage, whereas smart contracts are used to reach consensus, thereby enabling secure access, processing, and sharing of medical data among various e-health entities.

Balance health data have been described in Chapter 3 and is generally composed of two main parts: general personal information and medical health records. Examples of personal information include age, gender, and weight, while medical health records depend on the topic, i.e. medications and treatments. Considering our platform for collecting postural stability data, we store the patient's sensitive personal data along with the results of the measurements he performs [19]. The summary of these data can be found in Table 4.1, for the complete description refer to Chapter 3.

Sharing Scenario

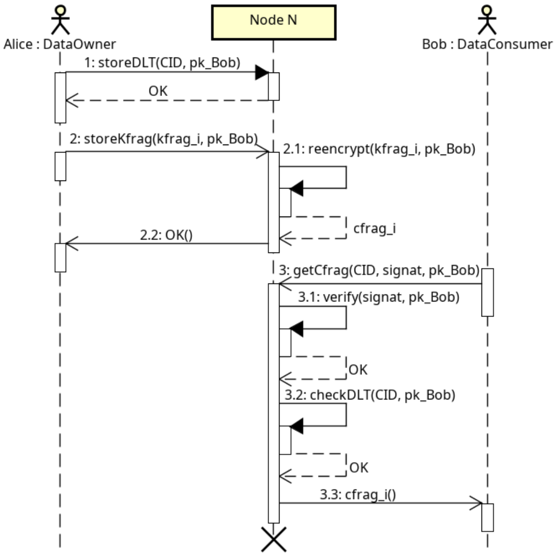

Alice : DataOwner
Node N
Bob : DataConsumer
1: storeDLT(CID, pk_Bob)
OK
2: storeKfrag(kfrag_i, pk_Bob)
2.1: reencrypt(kfrag_i, pk_Bob)
cfrag_i
2.2: OK()
3: getCfrag(CID, signat, pk_Bob)
3.1: verify(signat, pk_Bob)
OK
3.2: checkDLT(CID, pk_Bob)
OK
3.3: cfrag_i()

Figure 4.8: Halo Network Health Data Sharing Scenario

Let us now consider the following scenario shown in Figure 4.8: A system user, namely Alice, collects her data through her mobile device. We refer to her as the data owner. Another user of the system is her physiotherapist Bob, i.e., the data consumer. The idea is to share Alice's health data collected with Bob, which can use to provide a better medical evaluation for Alice.

1. Alice will first send her encrypted data to the DFS and contact the ACL to record all the necessary information for third-party authorization.
2. She distributes the keys to the DLT network to authorize her doctor. The DLT will retain the authorizations over time and serve as evidence of the transaction.

   2.1 The node re-encrypts the information shared by Alice, storing it locally.

   2.2 Provides communication that the process occurred correctly.

3. Once the process is successful, Bob is authorized and has the opportunity to look for Alice's health data. He will leverage the ACL in the opposite way of Alice, by recovering all the distributed parts from the DLT and decrypting the original health data.

   3.1 At this point Bob tries to get the necessary pieces from the nodes that store them.

   3.2 The node verifies that Bob's signature is valid.

   3.3 Through the shared DLT, it verifies that it is enabled to exchange the information.

Chapter 5

The Halo Network Use Cases

There are many use cases for the *Halo Network*. Potential uses of the data layer are listed below introducing two concrete actors in a real world scenario.

- **Exploiting data from IoMT**
  Alice is a medical doctor evaluating Bob's heart condition. Bob wears a FitBit. Bob grants Alice permission to access the activity tracker's records during a doctor's visit. Moreover, Bob issues general policies whereby his data is available in natural disasters, emergencies, or other cases of *force majeure*. Due to the availability of this sytem, Bob's data can be shared by personally applying his own rules without relying on third parties.

- **Proof of Authenticity**
  The insurance company Acme has made Alice an attractive insurance premium based on Alice's health habits, which can be guaranteed by the activity tracker's record. The IPHL provides the elements for Alice to provide proof of the authenticity of the activity records stored over time because of the access policies applied to allow a third party to access the data stored. Involved in a bilateral process, Alice consented to provide this proof.

- **Data Interoperability**

Due to administrative barriers, hospitals and health systems sometimes hold health records in isolated silos, and they do not have the possibility to bring the data together under one umbrella. Even if data flows, the possibility to make global changes does not exist, in the absence of an interoperability system, hospitals cannot make unique changes to individual patient records. The system could provide the ability to preserve user information and allows access by an external institution or individual who has been authorized to do so.

- **Data Altruism**
  Bob is a data altruist who wants to support medical research. He issues a policy whereby his activity tracker's data is also released to any research institution to produce new results in the field of medical research. He can set different policies with the given conditions to contribute to medical

- **Data Availability**
  Alice does not trust large institutions and prefers keeping her data on her mobile device. The first problem is that her smartphone is not always online, and the services using the policies she defined do not always work. However, Alice trusts her family and data lives replicated in a number of devices she trusts. Through the creation of a restricted-network, the system is fault-tolerant in the event of the shutdown of one of the nodes, and her data and her policies are available most of the time. The second problem appears when Alice's device is stolen or lost. Alice will not have a loss of data because it could be replicated in her household's and family members' devices. advancement.

5.1 Data Availability and Social Networks

Data availability in a decentralized network is complex, especially when this data is sensitive.

Li and Dabek [69] argue that, when implementing a distributed storage infrastructure in P2P systems, a node should choose its trusted node neighbors, i.e., the nodes with which it shares resources, based on existing social relationships, rather than randomly, e.g., their friends and colleagues. The system is called a F2F storage system, in which nodes are limited to sharing storage and network resources only with their friends. The authors argue that a friend-to-friend system incentivizes nodes to cooperate, resulting in a more stable system.

Based on this idea, they then proposed an online cooperative backup system called Friendstore, which allows users to back up data in trusted nodes (i.e., their friends and colleagues). Gracia-Tinedo et al. [47] showed that pure friend-to-friend storage systems have poor Quality-of-Service (QoS), mainly due to availability correlations, and proposed a hybrid architecture to combine it with cloud storage services. Their system uses erasure coding to replicate data and allows users to adjust the amount of redundancy based on the availability patterns exhibited by friends.

Liu et al. [70] presents a decentralized online social network designed to manage data without compromising user privacy, i.e., users' data are replicated to trusted servers controlled by friends.

However, today's work primarily focuses on designing intelligent data replication and storage policies. The approach proposed by Koll et al. [61] exchanges recommendations among socially related nodes to efficiently distribute replicas of a user's data among suitable nodes carefully selected in the OSN. In the approach developed by Olteanu and Pierre [90], preferences are given to nodes when it comes to selecting nodes for storing data (and their replicas) published by a user. The user's online friends have the highest priority. When all friends are offline, data is stored in nodes not part of the user's circle of friends.

Guidi et al. [48] use the Interplanetary File System (IPFS) to build a decentralized because of its decentralized nature for DOSN. In their work, they inspect whether IPFS is a good choice as data storage for Decentralized Social Applications.

Adding a social component should make the architecture more robust against failures because of an increased availability. The availability of a piece of data indicates its ability to be accessed at any time, at any place. However, decentralizing an infrastructure around individuals implies their commitment to ensuring that their data is always available. Therefore, guaranteeing such an approach means having to find strategies to guarantee it.

Social techniques and mechanisms can be vital in maintaining such an infrastructure. We mean social techniques designed to increase end-user involvement with the problem, such as the introduction of gamification techniques and social network activities.

This chapter contributes at giving a dual solution to decentralized data availability by exploiting the circle of trusted users discussed in the *Halo Network*. These users' could securely share stored information and delegate their information when needed, helping to improve data availability while ensuring privacy.

5.1.1 The Halo Network and Social Networks

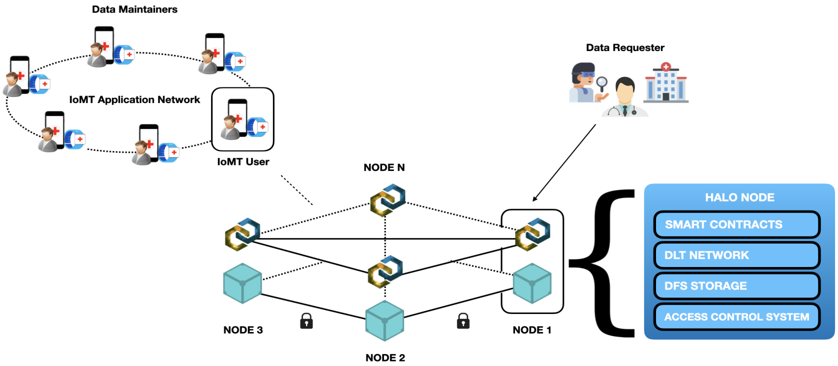


Figure 5.1: Decentralized Health Data Architecture for Data Availability

The IoMT, which enables remote monitoring, screening, and treatment of patients through telehealth, has been successfully adopted by caregivers, healthcare providers, and patients. IoMT-based smart devices and their applications are having a dizzying impact ubiquitously, particularly in the global pandemic state. The introduction of social networks could increase the possibility of ensuring the greater availability of data.

The Halo Network stores universal and immutable resource identifiers for data and provides smart contracts to ensure proper access control associated with each piece of data. We provide a mechanism geared toward increasing data availability in a decentralized context. In this context, the general approach is to securely store access control policies on DLTs so that the applicant can be made aware of his permissions to access his or her personal data stored outside the DLT [122, 118].

In the architecture shown in Figure 5.1, three main actors are identified that create such a social network:

- **IoMT User**: an IoMT user collects data through an IoMT device. An example of data used is health data, such as data related to one's postural condition. Then, through the IoMT application, the user creates a personal social network by adding other users whom we call data maintainers. IoMT users manage their node of the system described in this section and use the IoMT application to interact with the underlying components, i.e., to store data in the DFS and to manage the access policies through the DLT. Consider that this thesis introduces the system keeping even an eye on people with disabilities who are unable to manage their data.

- **Data Requester**: on the other hand, the data requester is generally a professional, researcher, or any entity that needs to take advantage of the data granted by the IoMT user and that needs to acquire permissions.

- **Data Maintainer**: To keep track of requests and to allow access to data, the IoMT user relies on its social network, i.e., a network of data maintainers who are none other than other IoMT users running a node. Based on the IoMT user policy, data can be exchanged or delegated to the data maintainers in such a way that data availability is ensured. It is not necessary for an IoMT user to worry about their node being online constantly because requests can be fulfilled by others in their trusted social network.

We focus on enabling users to replicate data, decide over it and involve them in storage and policy decisions in advance by employing a social network in a decentralized scenario.

The goal of this system is to provide a decentralized architecture to involve users in the decisions made concerning their data. A mechanism based on social networks consisting of a voting system has the dual purpose of representing an access mechanism to data and increasing data availability.

Users maintain their data, store it in their nodes, and create social networks to make joint decisions about the data, allowing the users in the network to replicate it and eventually delegate it. Through this mechanism, users of IoMT devices can directly own their personal data while ensuring availability through policies, i.e. delegation. A specific example for which the network could be relevant is in the event that a user becomes incapable, for some reason, of making decisions, such as in the case of an accident or sudden and unexpected disability.

Still, he might have delegated in advance his rights to trusted individuals in the social network. We will not dwell on the possible policies to identify delegates, or on devising proper multi-party decision-making schemes, since it is closely related to the specific use case. Indeed, we focus on the provision of a decentralized architecture fostering this kind of healthcare application. Thus, in this project, we will consider a naive data authorization scheme based on a voting system, i.e. the data owner and his delegates can vote to decide if a requester can get access to the data.

We describe the system architecture with the aid of Figure 5.1:

- **IoMT Application and Social Network** - end-users interact with an application for managing health data and providing social features for data availability.

- **The Halo Node** - the APIs which an IoMT Application can exploit to interact with the other system components.

- **The Halo Network** - the DLT, through smart contracts, is used to reach consensus on one's data, enabling secure access, processing, and sharing of medical data among different e-health entities.

- **Decentralized File Storage** - the DFS is responsible for facilitating data sharing and providing secure storage of information.
- **Access Control Layer** - the authorization mechanism coupled with approaches close to social networks enables the decentralization of users' health data and increases availability.
- **The Shared Voting System** - smart contracts that embed the authorization mechanism coupled with approaches close to social networks enables the decentralization of users' health data and increases availability.

5.1.2 Reasons behind the Shared Voting Mechanism

Smart contracts can be used to involve IoMT users in data management. In fact, a smart contract can enable data maintainers to accept or reject a result based on pre-established rules. This is fundamental to providing data availability and the possibility to continue the authorization service even when the IoMT user (the data owner) is offline. Each data maintainer, including the data owner, can vote through a smart contract whether or not to give data access authorization to a data requester. The idea is to implement a smart contract that provides a list of lists representing the social network constituted by the data maintainers and the list of their votes.

In conventional healthcare environments, health data are collected through personal mobile devices and generally stored in centralized locations. IoMT devices are thus forced to preprocess data on board or to hide information. The majority of health data are then hardly accessible or take the form of open datasets, of little use to interested stakeholders. Because of this, the traditional healthcare data management infrastructure is mostly self-managed or outsourced to third-party experts.

The infrastructure to protect data is an inaccessible infrastructure that integrates fine-grained encryption and access control techniques over a very specific domain. In this context, it is therefore difficult to make the best use of the information collected by IoMT devices and avoid raising additional privacy, security, and infrastructure cost issues [55].

Recently, however, DLT-based systems are proposing an overhaul of architectures by applying a different philosophy to data management, potentially including any data, such as those in the healthcare domain. We propose a use case that falls into this category. Our architecture provides a decentralized sharing of health data, ensuring that data can be transacted between institutions and individuals by storing provenance and immutability. However, these architectures being able to be fully decentralized, suffer from the problem of data availability, i.e. they may not guarantee stakeholders continuous access because the user providing data suffers from a disability or his node is offline. For this reason, we referred to an approach involving a social network that constitutes a network of trust and enables the user with the potential of delegation.

In general, the use case is the application of a voting mechanism in order to involve the user directly in the decisions made with respect to the data. This aspect has the dual purpose of representing an access mechanism and a social network. Individuals have their own data, save it in their own nodes, and create social networks to make joint decisions about the data. At the same time, the social network constitutes a network of trust, assuring remote users that they remain available in the case of various situations (e.g., if their own node is unable to respond). The authentication and authorization mechanism, coupled with approaches close to social networks, could enable better decentralization of individuals' health data and greater availability. DFS is responsible for facilitating data sharing and providing secure storage of information, while DLT, through smart contracts, is used to reach consensus on one's data, enabling secure access, processing, and sharing of medical data among different e-health entities, and increased data availability resulting from the introduction of social networks.

There are other specific cases for which such a network is crucial, for example, in the case of people with disabilities. Through this mechanism, these people can also own the IoMT devices and, at the same time, claim their rights through the automatic delegation made through the system and possibly imposing their own policy. In any case, although this represents a use case, we cannot investigate it in this paper for reasons of space and scope, wanting to focus on the issue of data availability.

Ensuring Data Availability Scenario

We consider a scenario where an IoMT user, Alice, collects her data through her smartphone and the application Balance. We refer to her as the data owner. Another system user is her physiotherapist Bob, i.e., the data requester.

Alice does not trust large institutions and prefers keeping her data on her smartphone or sharing it with her trusted network of individuals. But, her smartphone is not always online or could be lost, so the services using the policies she defined could only sometimes work. To address the issue, Alice trusts her family and allows them to replicate it.

This way, she avoids being a single point of failure through her household's and family members' devices, which we call data maintainers. However, Alice trusts her family, and data lives replicated in a number of devices she trusts.

Through the creation of a network of trust, the system is fault-tolerant in the event of the shutdown of one of the nodes, and data and policies are available most of the time. In this work, we consider Alice using a platform to collect health data related to her postural stability. With the platform, she is storing sensitive personal data along with the results of the measurements she performs. Her data comprises the ones shown in 4.1.

The IoMT application allows the user to see the users added to their network and who participates in maintaining the data by contributing to its availability, i.e., data maintainers.

The application also allows users to interact with their system node, thus enabling them to send or retrieve data from the DFS, be updated on external requests, vote on the smart contract in the DLT, and receive information on network participants.

At the same time, the user keeps track of all requests made by the network and votes on them keeping control over his data. These functions involve interaction with the DLT and retrieving data from the distributed storage. The full sequence for a standard access control process within her social network involves the following ten steps:

1. An incoming request from a data requester is forwarded to a data maintainer node.
2. The data maintainers create the record of the request, allowing all participants to vote.
3. The DLT replies with an acknowledgment.
4. The IoMT Users express their vote through their IoMT applications that are registered into the DLT.
5. The DLT replies with an acknowledgment.
6. After voting, the data maintainer checks if other maintainers have expressed their vote by following the specific policy related to the request.
7. A positive outcome grants permissions to the requester user along with requested data.
8. A negative outcome results in denied access.

5.2 Solid as an Alternative to IPFS

We investigated Solid to provide a different way of storing and sharing data through its Personal Online Datastores (Pods), instead of the IPFS. Solid is a system that was born to give users their data sovereignty. We investigate the ability of Solid to provide an interoperable way of sharing data and its distributed storage with the *Halo Network*. The nodes managing the DLT securely store all the health data sharing traces, i.e. access requests and consents, providing an immutable register of logs and the ability to execute smart contracts.

In the context of the IoMT, data remains in the hands of the individuals without the possibility of easily sharing it: the lack of data management technologies prevent unleashing the full potential of health data. We introduce the Open Digital Rights Language (ODRL) data access policies to allow data subjects to interact with IoMT devices and store and keep full control of their data stored in the PODs. The Open Digital Rights Language (ODRL) is a policy expression language that provides a flexible and interoperable information model, vocabulary, and encoding mechanisms for representing statements about the usage of content and services [54]. Since sharing data in the healthcare field is limited, the attempt should allow data never to be exchanged with remote users without consent and subjected to verifiable transactions thanks to the DLT and access mechanisms defined through Solid applications. The data requester, who will be defined as a consumer, will be able to access the data through policies described with the ODRL above the storage, while the DLT will take track of everything happening in the transactions.

Most works around Solid leverage the DFS as the off-chain solution for storing information. Specifically, some researchers focused on the possibility of introducing Solid and creating access layers above it to personalize the individual's data sharing such as Esteves et al. [37], proposed an access system following the ODRL policies. Policies are particularly convenient because they express the authorizations or prohibitions associated with the data stored in a given installation of Solid by means of the Data Privacy Vocabulary (DPV).

Other researchers, such as Ramachandran et al. [95] and Cai et al. [23], proposed a framework to store data generated by an IoT device in Solid with a Blockchain for validation purposes. Through an authentication mechanism, any third-party application can gain access to the IoT data in the Solid Pod and verify the authenticity of the data by cross-checking the hash of the data on the blockchain.

An adversary model that could challenge models created on the blockchain and Solid has also been described by Sharma [103]. In their proposed architecture, they consider that an adversary can force the private key to encrypt files. An adversary can also try to interpret sensitive data shared by the patient and the doctor through the IoT network, take possession of the server, or attack the web interface.

More works focus their attention on policy management in permissioned access. Kongruangkit et al. [62] believe that blockchain can provide a platform through which data access rights can be shared between users and service providers in a transparent and verifiable manner, especially when service providers need to enforce legitimate data access rights that may take precedence over those of users. The proposal provides for 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. These are useful when there is an interest in overriding user policies.

Ultimately, the use of Solid as distributed storage represents a novelty in the context and specifically in the Social Networks area.

5.2.1 Combining Solid with the IPHL

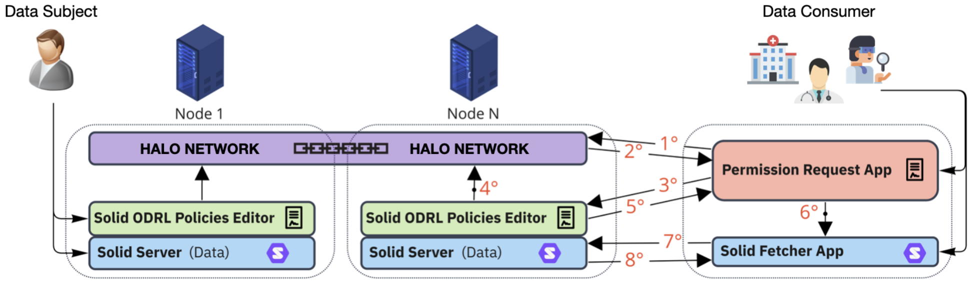


Figure 5.2: Combining Solid and the Halo Network - Architecture

We show in Figure 5.2 an overview of our proposed architecture that comprises four main components, represented with four different colors. The goal is to provide access to data through the use of Solid and the application of ODRL policies to guarantee access only to authorized parties. The main system actors are the data subject, the network nodes acting as data controllers, and the data consumer. In what follows we describe each component of the architecture:

- **Solid Server**: The first component we describe is the Solid Server[1], as it is the one that stores the health data. We assume that a network node manages a single Solid Server with one Pod related to a data subject. We will refer to resources as the files containing health data that can be accessed from the "outside" and are stored in a dedicated directory of the Pod. Each resource is identified by a Uniform Resource Identifier (URI), which reflects the containers under which the resource is actually stored, i.e., a resource A stored on the Solid would have a URI such as *https://my-solid-data-pod/resourceA*, while a resource B stored on a container A which in turn is stored under the Solid root level would have a URI as *https://my-solid-data-pod/containerA/resourceB*. Since we make use of Semantic Web technologies, it is possible to feasibly integrate the Resource Description Framework (RDF). RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. We also tag each resource with a specific personal data type, using the `rdf:type` predicate and the Data Privacy Vocabulary (DPV) personal data categories taxonomy[2], so that the specified ODRL policies, which are defined for personal data categories and not for resources, can actually be associated with the resources that contain the data to which the policy refers to. For the name of the resource (and thus the name of the file contained in the Pod) we make use of a specific protocol in order to keep the content unmodified for later audits. Specifically, instead of naming a file "normally", i.e., *"file1.txt"*, we make use of the file's content hash digest, i.e. *"QmdmQXB2m...DJ5MWcKMKxDu7RgQm"*. This is in line with the fact that if the content was specific at the time of an access request, an audit must verify that, subsequently, the file may have been altered. To generate the file's content hash digest, we make

[1] https://github.com/CommunitySolidServer/CommunitySolidServer

[2] http://www.w3id.org/dpv/dpv-pd

use of the InterPlanetary Linked Data (IPLD) technology ([56]), a set of standards and technologies leveraged to create universally addressable data structures. Encoding the file's content with IPLD standards makes it so that the result itself contains both the hash and data decoding information. The Pod in the Solid Server also contains some other files that cannot be accessed by consumers, such as the policy files.

- **SOPE application**: SOPE[3] is a Solid-compatible application that generates access control policies that are stored under the private container of a user's Pod. The generated policies are specified according to the ODRL Access Control profile[4] (OAC) [37] which aligns ODRL[5] – an RDF standard to specify policies over assets – with DPV[6] – a specification that contains taxonomies related with the privacy and data protection domain and specifies terms such as purposes for processing or legal basis. The policy files generated by the application are stored on the user's Pod and reflect the Pod owner's data-sharing preferences. Since they are stored in the private container of the Pod, by default, only the Pod owner has access and can modify or delete them. The policy file is the one used by the data subject to rule data access and cannot be accessed from the "outside". It is stored in a dedicated path of the subject's Solid Pod.

- **Permission Request App**: A Permission Request App can be any application used by the data consumer that also uses OAC to specify the type of data request that they want to make. However, to guarantee future auditability, the request policy should also be stored on the Pod and be accompanied by some data taken from the DLT related to the data subject whose access to resources is being requested, i.e., latest resources Merkle tree root and latest policy Merkle tree root. All these pieces of information are digitally signed

[3] https://github.com/besteves4/solid-sope

[4] https://w3id.org/oac/

[5] https://www.w3.org/TR/odrl-model/

[6] https://w3id.org/dpv

and stored safely by the consumer in appropriate storage (not part of this work).

- **The Halo Network**: The permissioned DLT is maintained by several network nodes to execute smart contracts and store hash digests or Merkle tree roots (as shown in Figure 5.3, top). Each time a data subject updates policies or resources, the DLT is updated. Each time a data consumer makes a request, the DLT is updated. However, the ones that have access to the ledger only see new hash digest information but no information on data consumers or the number of accesses or types of accesses. Nonetheless, the system, even in this configuration, guarantees a series of features in favor of complete transparency. This act of "logging" represents a guarantee for future audits, and, moreover, the DLT pones also the basis for other applications exploiting smart contracts. The reason to use a DLT is simple: if requests and consents are stored only in the data subject/controller side (i.e. in her/his Pod), the data consumer will not be protected in case of malicious behavior; on the other hand, if requests and consents are stored only in data consumer storage, then the data subject/controller has difficulties to prove the possible unlawful behavior of the consumer; finally, if both actors store two different copies of requests and consents, then difficulties once again arise to validate one or the other copy. The information stored in the DLT is organized in a series of smart contracts, one for each data subject, in which four key-value dictionaries store the logs. Each dictionary has a timestamp as a key and the root of a Merkle tree as a value. This enables keeping a history of the requests, consents, and modifications and possibly implementing some logic exploiting these logs directly through the smart contract.

To summarize Figure 5.2: (i) the network of nodes builds the Halo

Network. Each node maintains the ledger and executes the same protocol, i.e., the same consensus mechanism and the same smart contracts execution. (ii) Each node acts as a data controller maintaining and executing two implemented functions: the Solid ODRL Policies Editor (SOPE) app [38], and the Solid server. SOPE is used to communicate with the DLT, manage health data access policies and receive requests. (iii) The data consumer uses a Permission Request App to communicate with the node's Solid Permission App, triggering the DLT to request data access. Data consumers specify their data needs using a data request policy through the Permission Request App, and the interaction with the DLT verifies the data request and the policies set by the individual. (iv) A Solid server is maintained by each node to store health data on a Solid Pod.

5.2.2 Solid Policies

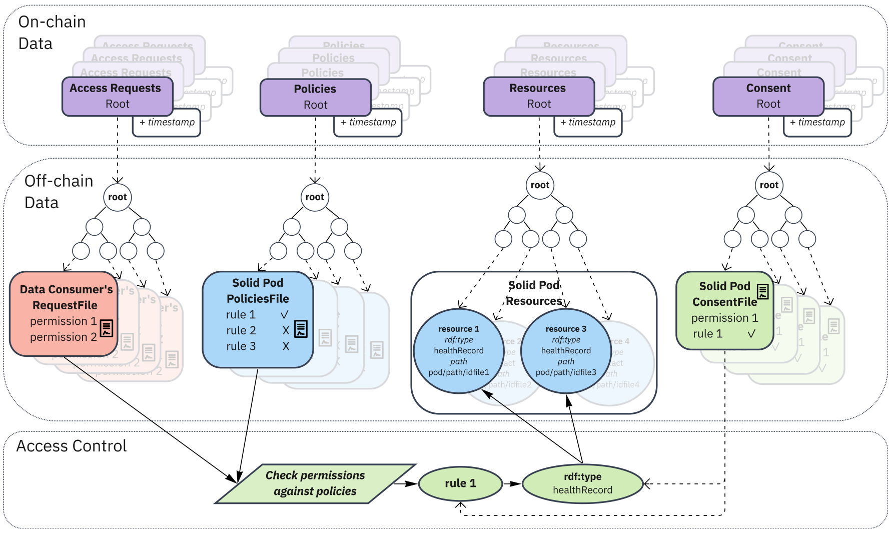


Figure 5.3: Access Control Model that uses the ODRL Access Control Profile to Provide Access to Solid Resources

Figure 5.3 (bottom) provides a diagram that explains how these policies are used for access control. A data access request is specified by the data consumer also, according to the OAC profile. This request is checked against the rules contained within the policy files added to the Pod by the user. The result of this operation is a set of rules that either permit or prohibit access to the resource (and can be an empty set). Each rule is associated with a certain type of personal data resource, i.e., rule 1 has *rdf:type healthRecord*. Thus, to the consumer, it is only returned the list of paths of resources that have an associated personal data type which is permitted by the rules set by the user. Moreover, a consent record containing information related to the matching operation should be created and stored by the node in the Solid Pod.

```
:example-4-4 a odrl:Policy ;
    odrl:profile oac: ;
    odrl:assignee :app-2-controller ;
    odrl:target oac:HealthRecord, oac:Prescription,
                                    oac:HealthHistory ;
    odrl:permission [
      odrl:action oac:Collect, oac:Copy ;
      odrl:constraint [
      odrl:leftOperand oac:Purpose ;
        odrl:operator odrl:isA ;
        odrl:rightOperand dpv:AcademicResearch ] ] ;
    odrl:permission [
      odrl:action oac:Anonymise, oac:MakeAvailable ;
      odrl:constraint [
        odrl:leftOperand oac:Recipient ;
        odrl:operator odrl:isA ;
        odrl:rightOperand dpv:ThirdParty ] ] .
```

Listing 5.1: ODRL policy example for sharing health records using RDF

Moreover, the Solid Permission App also manages a list of data structures that help to maintain an untamperable log of health data accesses (as explained in the following sub-sections). In Figure 5.3 (middle) Merkle trees are graphically shown to indicate the link between data handled by the Solid Permission App and the DLT ([50]). The roots of the Merkle trees or directly the digests are stored in the DLT:

- hash digest of each resource in a given time is used to create "resources Merkle tree" leaves;
- hash digest of each policy file in a given time is used to create "policy Merkle tree" leaves;
- hash digest of each request file received in a given time is used for "requests Merkle tree" leaves;
- hash digest of each consent file in a given time is used to create

"consent Merkle tree" leaves;

All the request and consent files are stored in two specific directories of the subject's Pod.

Storing/Updating/Removing Resources and Policies The data subject communicates directly with the Solid Permission App of a specific node, i.e., the data controller. In this case, the process of storing, updating, or removing a new resource and or a new policy (the policy file can be seen as equivalent to a resource) is always preceded by two steps:

1. updating the related "resources Merkle tree" or "policy Merkle tree", by creating or updating or removing a leaf;
2. storing the new Merkle tree root in the data subject's smart contract dedicated data structure.

5.2.3 Data Consumer Access Request

For the description of this part of the protocol, we will refer again to Figure 5.2, in particular to the arrows with red numbers.

1. The data consumer starts the access request by invoking the Permission Request App's main procedure; this procedure firstly checks the data that is currently stored in the smart contract related to the data subject whose access to data is being requested.
2. The smart contract returns: (i) the latest resources Merkle tree root, i.e., a tuple *(resRoot, timestamp1)*, and (ii) the latest policy Merkle tree root, i.e., a tuple *(polRoot, timestamp2).*
3. The data consumer creates a request using ODRL with a set of permissions; then the consumer sends a request payload to the nodes' Solid Permission App, which includes:
   - the request file written in ODRL;
   - the tuple *(resRoot, timestamp1)*;
   - the tuple *(polRoot, timestamp2)*;

- the signature of the above data.

4. The Solid Permission App executes a series of sub-processes:

   (a) it updates the requests Merkle tree adding the request payload, thus adding the payload's hash digest as the tree's leaf;

   (b) it invokes the data subject's smart contract providing the root of the updated requests Merkle tree and the two tuples of the request payload. The smart contract checks if, in its storage, the associated timestamps are associated with the same *resRoot* and *polRoot* provided by the consumer and that they are the latest ones (this validates the time of the request and provides a mean to validate the history of requests). If all checks, the new requests Merkle tree root is stored in the smart contract.

   (c) it checks the consumer's request file against the policy file in order to (possibly) provide access to data.

   (d) The rules that (possibly) match with the request are linked to a set of resource types (*"rdf:types"*).

5. The final output that is sent back to the Permission Request App is the list of resource identifiers that are associated with the types of rules matching with the request and their paths.

6. The Permission Request App forwards these paths to a Solid Fetcher App.

7. This Solid Fetcher App enables the data consumer to simply use the Solid API for accessing resources.

8. Thus, the consumer accesses the health data by getting the resources through their paths, directly from the subject's Solid Pod.

Chapter 6

Experimental Results

This section describes the evaluation of the IPHL architecture implemented in this work. We focused on testing the interaction with the DLT and DFS. The DFS is responsible for storing data, while the DLT is responsible for data management mechanisms. For each section, we show the performed experiments and the obtained results.

We summarize the experimental components in the following:

- The DLT Layer deployed is based on the Hyperledger Fabric Framework. It is a permissioned ledger where all the participant's identities are known and authenticated. The smart contract is implemented by exploiting the Fabric's Chaincode.

- The Access Control Layer includes a smart contract, a TPRE scheme and a key distribution mechanism. Chaincode allows the storage and retrieval of relevant information to the access mechanism in a shared and immutable way. Furthermore, the TPRE scheme and keys distribution are built using the Rust language and are based on the Umbral protocol [89].

- The DFS layer is based on the IPFS technology, which allows storing and accessing data on the IPFS network in a persistent but not permanent condition, which means that data stored on IPFS can eventually be deleted.

- The Halo Node exposes an API through which the users can interact with the system.

The tests and datasets can be found in [115].

6.1 On the Feasibility of the InterPlanetary Health Layer

In what follows, we describe the experiment performed to test the feasibility of the IPHL.

6.1.1 Experimental Setup

In order to test the DLT, we launched a Hyperledger Fabric network of four nodes in a real-use case scenario in various locations worldwide, including Europe, the United States, and China. The nodes have two cores, four gigabytes of RAM, fifty gigabytes of storage, and run Ubuntu 18.04 LTS. Since Fabric requires each participant to maintain their infrastructure, Docker Swarm can facilitate deployment through an overlay network. The fourth node is then responsible for synchronizing the ledger due to the deterministic nature of the Fabric consensus algorithm.

Regarding the DFS, the experiment was deployed on two VMs with the same specifications as the previous VMs used for Fabric. The *Read* and *Write* tests were conducted on a single machine, which we refer to as the *producer*, while the *Replication* tests were conducted on a second machine, which we refer to as the *consumer*. OrbitDB, which builds a decentralized database on top of IPFS, is accessible to both the producer and the consumer. In addition, the producer is the creator of the OrbitDB database, meaning that the consumer relies on its replica. Therefore, once it has established its own IPFS node, it does not bother to create a decentralized database; rather, it connects to the one created by the producer and shares a copy of its own database.

The experiments were conducted by interacting with the Halo Node via a NodeJS-developed remote client. Python-written scripts were used to determine the performance of the host machine (such as CPU load and network traffic).

6.1.2 Testing the Distributed Ledger System

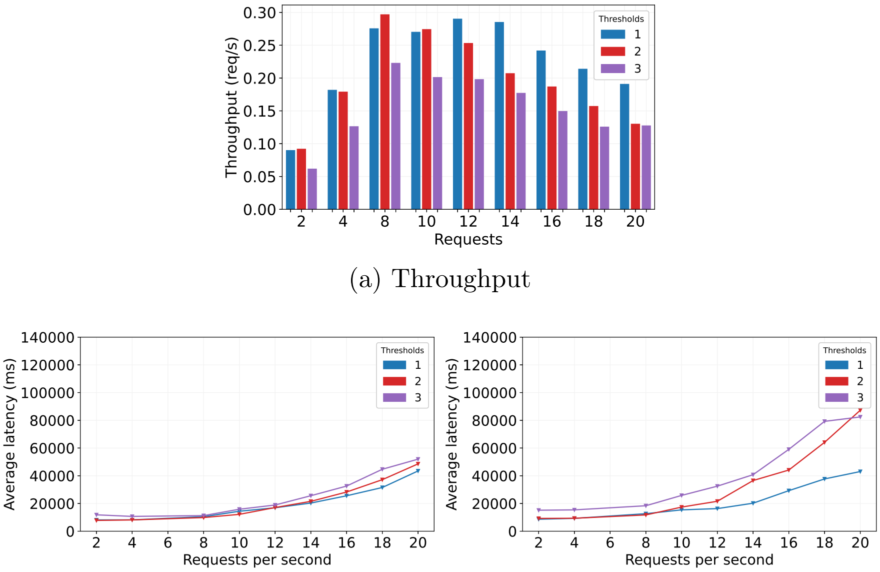
The workflow is composed of three primary operations accessible through the smart contract: (i) CreatePoll; (ii) Approve/Decline; (iii) ClosePoll. During the test, we simulate the delay of the real user in reacting to a new request to vote with a parameter randomly given by a Poisson Process with a mean $\lambda = 1000$ms. We collected information about the following parameters and metrics:

- **Fixed parameters**: the number of the maximum DLT nodes n was set to 3. For each test, the same requests were repeated five times. This means that we average the timings of the same tests.
- **Independent parameters**: the threshold t of the (t, n)-threshold scheme varies in the tests from 1 to 3, representing an increasing load on the DLT. A second parameter is the *number of requests per second* generated by the remote users, which varies from 2 to 20.
- **Dependent metrics**: the *request latency* which is the time between the submission of the request and its actual completion. Notice that this time is built on the first 7 contributions of the steps described in Chapter 4.

Results

Figure 6.1 latency depicts the throughput of the system, writing into DLT, and updating into DLT, with increasing requests per second and varying threshold values. A raised threshold allows network nodes to access the ledger simultaneously. In general, the results demonstrate a distinct relationship between the number of requests per second and the value of t.

At the top of Figure 6.1a the system's throughput increases as the number of requests per second. The graph depicts a performance peak



(a) Throughput

(b) Write on DLT

(c) Read on DLT

Figure 6.1: IPHL Feasibility: DLT Testing Results

when eight requests are sent to the node. Before this threshold, the system suffers from overuse, and after it, the overall performance declines. The chart provides a measure of scalability because the system becomes less efficient as the number of requests per second increases. Considering the user's interaction with the voting system, the obtained results indicate the system as a whole. As one might expect, as the number of users rises, the system's ability to respond efficiently diminishes as it waits for more votes. Despite the increase in waiting time, the system is still usable.

Regarding the operations *Write* (Figure 6.1b) and *Update* (Figure 6.1c), it is interesting to note how they ultimately differ. In the Update chart the difference in the spread of the threshold curves is most apparent. Indeed, it appears that the simultaneous interaction of multiple nodes (a greater value of t) would result in longer wait times on the ledger, likely due to the management of access conflicts. The increase in the number of concurrent nodes updating the same data on the distributed ledger verifies that the Fabric DLT employs lock-free optimistic concurrency with rollback for dirty reads/writes.

Unlike the *Update* operation, the *Write* operation does not appear to be affected by an increasing number of nodes. This is likely because nodes only insert new entries into the ledger, which do not generate conflicts.

In the best-case scenario, the DLT should establish approximately eight concurrent network connections, as we can assume that performance degrades beyond this threshold and we cannot guarantee low average latencies. In the worst-case scenario, the average latency could nearly double if there are 20 data consumers.

6.1.3 Testing the Distributed File System

The phases considered consist of the following operations:

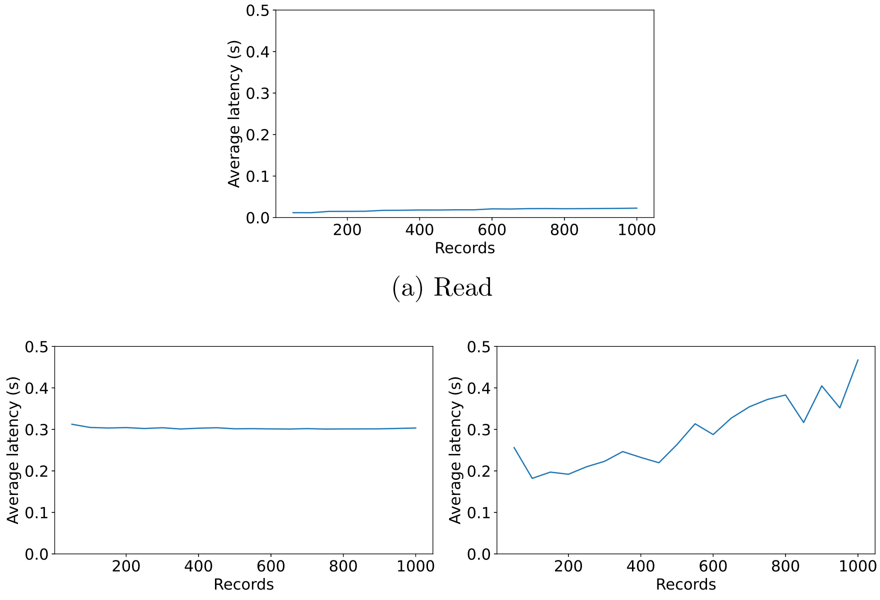
- **Read**: this is the operation through which we stress the architecture by retrieving data on IPFS. At each step, an N number of records is requested by a remote user in order to verify the response times of OrbitDB (linked to IPFS).
- **Write**: this is the operation through which we stress the architecture by injecting data on IPFS. At each step, an N number of records is inserted by a remote and local user in order to check the response time of OrbitDB and IPFS.
- **Replication**: IPFS and OrbitDB work in tandem as information storage and organizer, this means that when a user has his own IPFS node, he only has to connect to a second OrbitDB node to know all about his data. OrbitDB can set permissions, which means that knowledge of information on IPFS can be locked and selective. By establishing the connection with the other node, the data is replicated. During the replication, we test the latencies between the producer (writing node) and the consumer (the reading or receiver node) as the number of records increases.

We observed the following parameters and metrics during the tests:

- Controlled parameters: the $n$umber of records requested, inserted, replicated. In this experiment, the number of independent tests at each step i, with a step increment size of 50 records, was 10. This means that each experiment performed at step i is repeated 10 times and then averaged. The active nodes are always two: Producer and Consumer.

- Dependent metrics: we measure the *latency* to accomplish a request, the *CPU load*, and the *network traffic* of the machine running the Halo Node. Notice that the latency values also include the contribution due to the network transmission.

Results



(a) Read

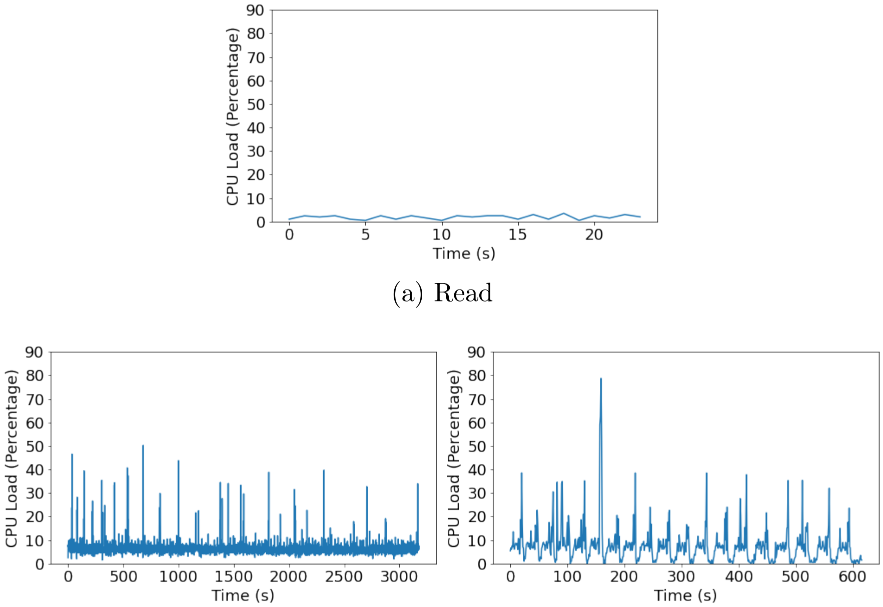
(b) Write

(c) Replicate

Figure 6.2: DFS requests latencies

Reading and writing were measured on the producer, while replication was evaluated on the consumer. The charts were produced with the following conditions:

- Figure 6.2: latencies were evaluated through a client that makes the request and waits for responses. The requests are issued in

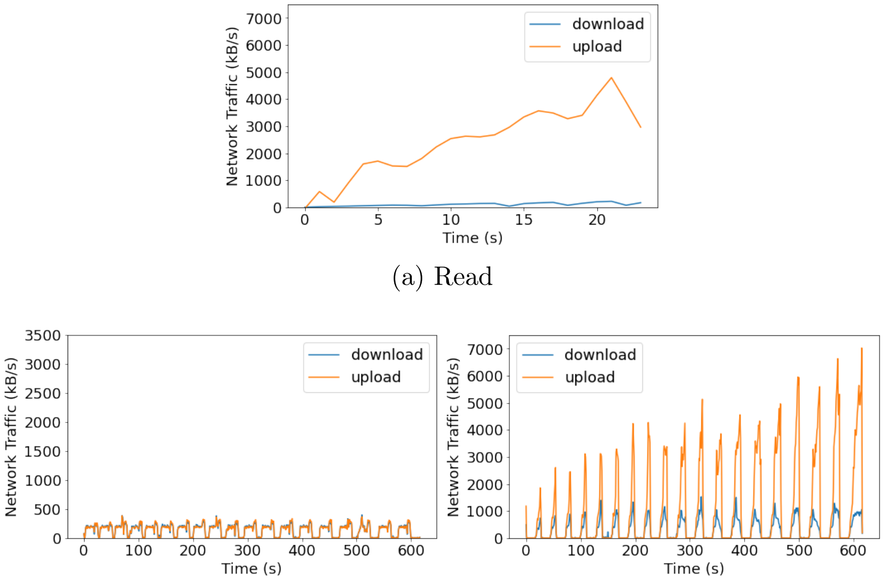


(a) Read

(b) Write

(c) Replicate

Figure 6.3: CPU load



(a) Read

(b) Write

(c) Replicate

Figure 6.4: Network traffic

parallel. The *Read* and *Replication* wait for data while the *Write* wait for ACKs.

- Figure 6.3 and Figure 6.4: CPU and network activity values were collected directly on the machine involved. The producer for *Read* and *Write* while in the case of *Replication*, the evaluation is done on the consumer.

Figure 6.2 shows the measured latency when varying the number of the records during read, write, and replicate operations. The results highlight that the three operations on OrbitDB are quite efficient, with the *Replication* being more burdensome than the others. Also, in the case of the *Replication*, the latency never reaches 500 ms, which is, however, an acceptable value.

This is an important detail because the *Replication* operation tells us how the size of the database to be replicated contributes to a deterioration of the general performance. Although the values are normalized, as the information increases, keeping the decentralized database integrity is presumably more expensive. For what concerns the *Read* and *Write* operations, we measure negligible latency values not exceeding about 300ms, and that shows no dependence on the number of records.

The CPU load is reported in Figure 6.3. Notice that the *Read*, *Write*, and *Replication* charts report in the x-axis the value of the time taken to perform the complete tests whose latency measures are shown in the previous figure. Since the latencies of the three operations are appreciably different, the total duration of every single test varies considerably, resulting in different time scales. In general, how efficiently it is read can be seen right away. In fact, it always remains quite low all the time despite the number of read records increases.

On the other hand, a higher load is visible for *Replication* and *writing*, showing easily distinguishable peaks in the correspondence of the execution of the query operations. Moreover, the former always seems more CPU demanding than the latter. The impression is that it occurs in chunks, requiring a higher computational effort that does not result in *writing*. This behavior is closely linked to the implementation of OrbitDB in the case of *writing* and *Replication*.

The results for network utilization can be seen in Figure 6.4. They confirm what was suggested by previous charts, with *Read* and *Write* being very efficient and with the *Replication* more wasteful. Also, in this case, during the execution of the *Replication* tests, the upload and download peaks are easily distinguishable.

The *Read* charts shows that although upload increases (as more records are returned), this has no real impact on total latency that is significant. Note that upload activity only has significance in *Read* and *Replication* because they contact the client to provide the data back. As for download activity, it is very low in the case of *Read*, which is justified by the fact that the operations do not involve write operations. In the case of *Replication*, on the other hand, it is much higher, indicating an increase in activity during *Replication*. The same thing occurs during *writing* but with significantly lower network activity (notice the scale is different to appreciate the behavior in this case). This is justified by the fact that in contrast to *Replication*, when *writing*, the data make use of the API as protocol (instead of OrbitDB replication protocol), probably leading to lower network activity values. The remaining upload activity is related to the IPFS, since once the node receives the data, it must forward it to IPFS.

6.2 Decentralized Data Sharing

This section describes the experiment performed on the IPHL when testing the Access Control Mechanism built to enable data sharing.

6.2.1 Experimental Setup

The setup consider the same experimental setup used in the previous experiment introducing the key distributions mechanism described in Chapter 4.

The network deploys four nodes geographically distributed: two of them in Europe, while the other two in the USA and China respectively, the idea is to represent a real case scenario. The virtual private servers used as node instances have the following specification: two cores, 4 GB of RAM, 50 GB storage, and run Ubuntu 18.04 LTS.

In order to perform the tests, we simulated the issuing of new keys and access to these from several data consumers. Specifically, we simulated a set of data owners injecting new capsules into the system and a variable number of data consumers wanting to access these capsules. The simulation starts with a client acting both as a data consumer and owner (a personal computer with internet access) and interacting with the DLT network.

6.2.2 Testing Workflow

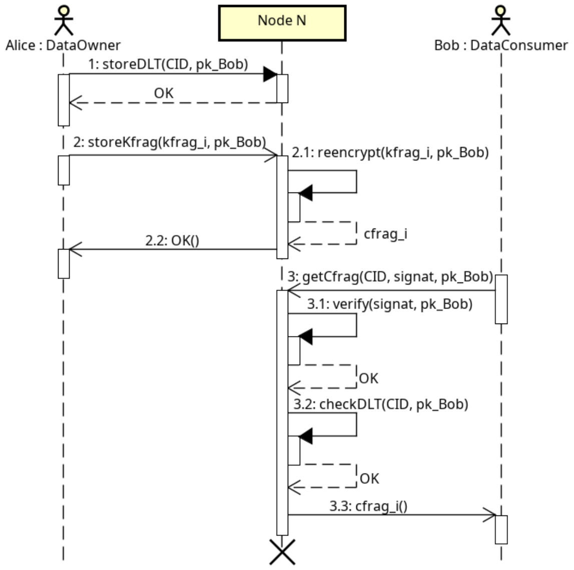

Alice : DataOwner
Node N
Bob : DataConsumer
1: storeDLT(CID, pk_Bob)
OK
2: storeKfrag(kfrag_i, pk_Bob)
2.1: reencrypt(kfrag_i, pk_Bob)
cfrag_i
2.2: OK()
3: getCfrag(CID, signat, pk_Bob)
3.1: verify(signat, pk_Bob)
OK
3.2: checkDLT(CID, pk_Bob)
OK
3.3: cfrag_i()

Figure 6.5: UML Sequence Diagram showing the main operations carried out during the testing by the simulated actors.

The testing flow needs several pre-processing steps. First, it is necessary to configure the environments of all the simulated entities. Then, for each actor, a set of asymmetric keypairs, e.g., (pk_B, sk_B), is created for encrypting-decrypting data and for digital signing. A piece of data is encrypted for a data consumer, and the associated capsule is created and distributed to the DLT nodes. This operation is independent of any data consumer request.

The foremost step is executed in parallel for each simulated data consumer. This step consists of a request composed of three primary operations shown in Fig. 6.5:

- **StoreDLT** - it is the operation where a data owner indicates to a DLT node to add a public key pk_B to the ACL in the smart contract for a specific CID, i.e., the owner instructs the DLT nodes to give access to the data represented by the CID to the consumer pk_B.

- **StoreKfrags** - it consists of a series of methods that perform the actual key distribution. During the pre-processing step, a capsule is created for each piece of data shared. The data consumer uses the capsule to create a fragmented re-encryption key, following the (t, n)-threshold scheme. The re-encryption key is unique for each data consumer. The single re-encryption key fragments are unique for each DLT node. We call these key fragments "kfrags" for simplicity. Each of the n DLT nodes, thus, receives a unique $kfrag_i$ and can perform a re-encryption for the indicated pk_B (step 2.1 in Figure 6.5). The result is a fragment of another capsule that will be used by the data consumer. We call these capsule fragments "cfrags" for simplicity.

- **GetCFrag** - The data consumer requires at least t *cfrag*s to reconstruct the capsule needed for the decryption. Thus, it performs a remote procedure call using the *getCFrag* operation to t DLT nodes. It also provides in such requests the signature of a message as a way to authenticate itself (in this step, we skipped the whole challenge-response mechanism in which the server, i.e., the DLT node, sends to the client, i.e., the data consumer, a challenge message with a nonce to sign). Each DLT node autonomously verifies the signature (step 3.1 in Figure 6.5) and checks if the related pk_B is present in the ACL related to the indicated CID (step 2.2). If so, it returns the unique $cfrag_i$ to the data consumer.

The final post-processing step involves each data consumer aggregating the $cfrag$s, obtaining the content key, and decrypting the piece of data.

6.2.3 Parameters and Metrics

We observed the following parameters and metrics in the testing:

- **Controlled parameters** - the number of DLT nodes n was set to 3. The number of independent tests was set to 3, and in each test, the main step described previously was repeated 10 times for each data consumer (from now on, this main step will be referred to as *request*). In this case, the time between a request and the next one was given by a Poisson Process with a mean $\lambda = 1000ms$.

- **Independent parameters** - the *threshold* t varies in the tests from 1 to 3. The number of *requests per second* depend on the data consumers, which vary from 10 to 100, with an increase of 10 each time.

- **Dependent metrics** - the *latency* for a response to a *request* is the measure we are interested in. As well as the *latency* in encryption, decryption, and *kfrag*s generation operations.

6.2.4 Results

We recorded the latency for each operation, including the latency of network transmissions. Only the kfrags generation and encryption/decryption latencies do not include network transmissions' latency. Moreover, no errors were recorded during the whole set of tests.

Requests per second

Recall that a request is the execution of the sequence: *StoreDLTs*, *StoreKFrags*, and *GetCFrags*. Thus, Figure 6.6 shows the results for each operation when the request per second is increased for different values of t. In general, results show a strong dependence on the requests per second

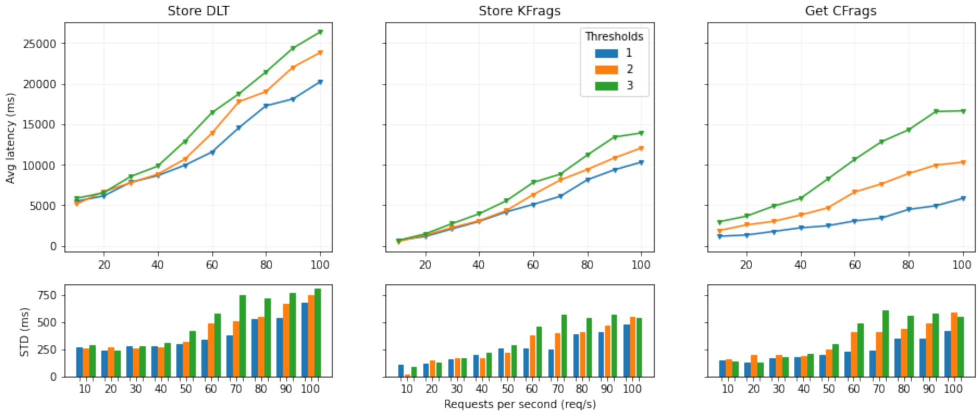


Figure 6.6: Decentralized Data Sharing Results: Average Latency per Operation

value and also on the t value, but the three operations behave differently. Moreover, we see a clear inflection point after 40 requests per second, especially for *StoreDLTs* and *GetCFrags* operations. The *GetCFrags* operation (rightmost plot in Figure 6.6) is the one where the difference in the three thresholds curves' spread is more evident. This is because the operation heavily depends on t, i.e. the data consumer makes a request to t nodes. In the other two operations, the effect of t is indirect because the number of nodes to which a request is made is fixed.

Threshold value

Figure 6.7 shows the results when increasing the t value and the requests per second for each i-th request, i.e. it shows the performances for each subsequent request instead of aggregating all requests through their mean. In this case, results show how the increase of t amplifies the response delay due to the increase in the requests per second. Specifically, this temporal point of view shows that a low t value (i.e., $t = 1$) keeps the response latency almost stable, while a higher t causes an accumulation of delay in the response, which worsens the performances (i.e. with $t = 2, 3$, from the 5-th request to the 9-th one).

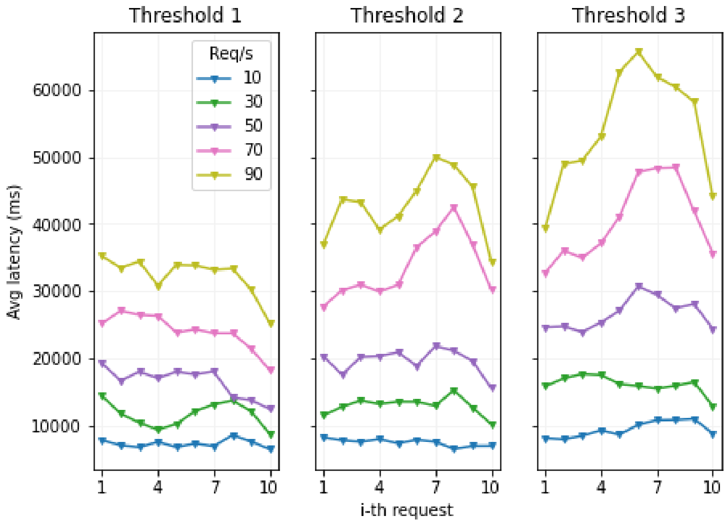


Figure 6.7: Average Latency per i-th Request Step and Requests per Second

Scalability

Figure 6.8 shows the results for the total average latency of all operations when the requests per second increase. The plot at the top normalizes the latency for the number of requests per second made to the network, i.e. the recorded average latency is divided by the requests per second. This gives a measure of scalability, meaning that when increasing the requests per second, the normalized latency values should remain equal to the previous (or best performing) step in an ideal scenario (the dotted lines in Figure 6.8 show the minimum normalized latency for each t). More in general, we obtained a linear dependency on the number of requests made concurrently. The optimal-case scenario is deducted by considering latencies below 20 seconds on average, which seems can be reachable when we set $t = 2$ and 50 data consumers. In this case, in the network of 3 DLT nodes, each node handles 16.7 requests per second. In the worst case, the average latency reaches almost 60 seconds, i.e. when the configuration is set to $t = 3$ and 100 data consumers, each DLT node handles 33.3 requests per second. The best-case scenario (in terms of acceptable request-response delay) seems to happen when each node

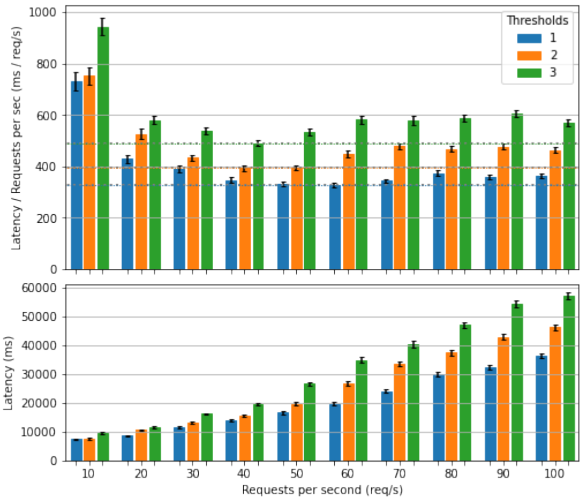


Figure 6.8: Average Latency per request per second.

handles about 13.3 request per second, i.e. 40 data consumers, with a response latency ranging between 13 and 19 seconds, depending on the threshold.

Finally, we focus on two operations executed only once per key or payload and happen only on the data owner or data consumer node. Thus we measured these without considering network transmission. In Figure 6.9 two plots are shown. The first one represents the average latency of the *kfrags* generation operation varying t. The second one represents the encryption and decryption operations latency when the payload size (i.e. the data shared) increases. Results show a linear dependency of the *kfrags* generation on the t value that does not cross the 100ms even at higher thresholds (50) and denoting an exponential behavior for the encryption and decryption operations, leading to a system under pressure when the payload's dimension overcome the 10kB.

6.3 Decentralized Data Availability

In the following, we describe the experimental environment and steps made for evaluating the Data Availability.

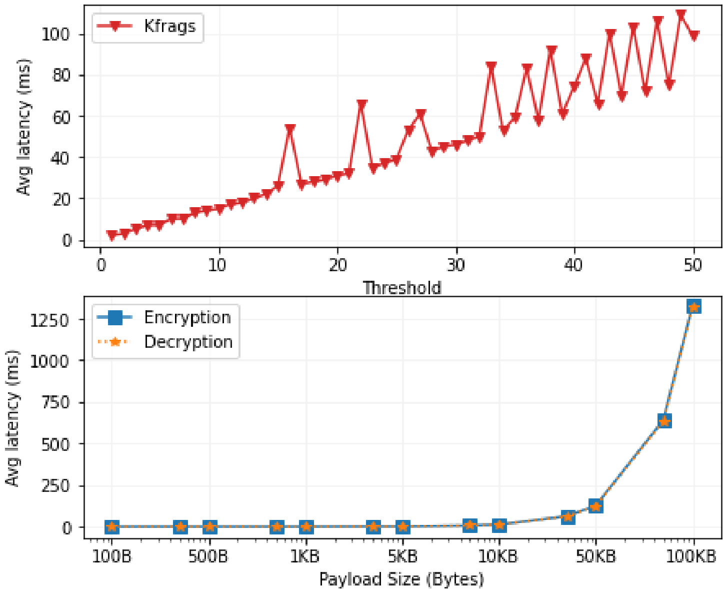


Figure 6.9: KFrags generation and Encryption/Decryption latencies

6.3.1 Experimental Setup

The setup consider the same experimental setup used in the previous experiments focusing on data availability issue described in Chapter 5.

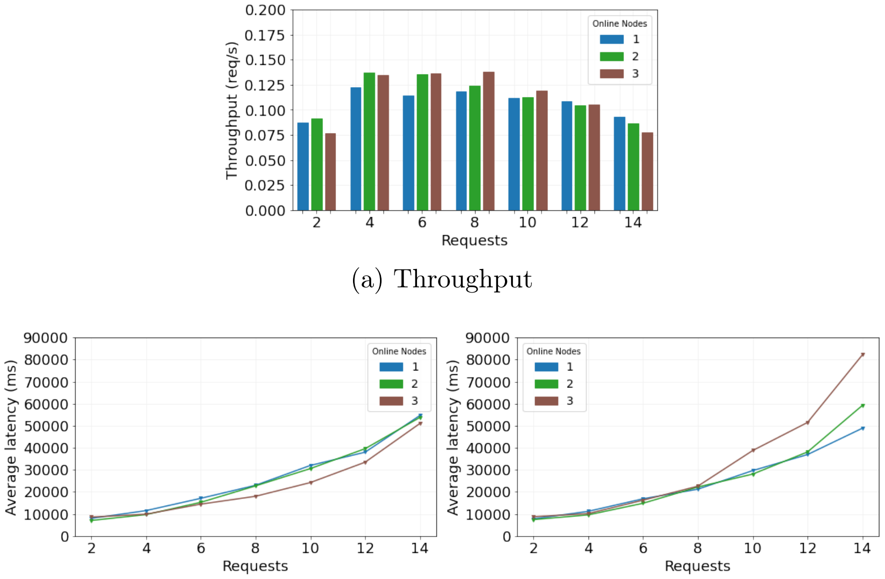
The network deploys four nodes geographically distributed: two of them in Europe, while the other two in the USA and China respectively, the idea is to represent a real case scenario. The virtual private servers used as node instances have the following specification: two cores, 4 GB of RAM, 50 GB storage, and run Ubuntu 18.04 LTS.

6.3.2 Testing the Distributed Ledger System

The experiment was conducted to verify the architecture's performance in case the data maintainers failed (i.e., they could not serve requests). The study was conducted following steps 1 through 8 described in Chapter 4 and consisted of three operations that interact with the smart contract: (i) Create(); (ii) Accept()/Reject(); and (iii) Get(). During the test, taking into account the delay of the real data maintainer in reacting to a new vote request with a parameter given randomly by a Poisson process with an average $\lambda = 1000$ms. We collected information on the following parameters and metrics:

- Fixed parameters: the maximum number of active data maintainers n was set to 3. For each test, the same queries were repeated five times. This means that we averaged the times of the same tests.
- Independent parameters: the active data maintainers t of the scheme (t, n) varies in the tests from 1 to 3, representing the increased availability of working nodes in the network. A second parameter is the number of requests per second generated by requesting users, which varies from 2 to 14.
- Element-dependent metrics: request latency, i.e., the time between sending the request and its actual completion.

6.3.3 Results



(a) Throughput

(b) Write on DLT

(c) Update on DLT

Figure 6.10: Data Availability: DLT Testing Results

Figure 6.10 shows the system *throughput*, *write* and *update* operations as requests per second and the number of online maintainer nodes increase. An increase in the number of maintainer nodes in the network corresponds to higher data availability, as they are reachable. If only one maintainer node is online, then all requests are redirected to it. The results show a clear dependence on the number of requests per second and the value of t.

The plot at the top shows the system's throughput as the number of requests per second increases. The throughput is lower when the nodes are not all active, and a peak performance increase is evident when we are in the presence of about 8 requests to the data maintainers. This is verified before the threshold mentioned. After that threshold, the overall performance deteriorates, and the throughput flips and gets worst globally and with more nodes involved. The chart provides a measure of scalability, meaning that the system is less efficient as the number of requests per second increases. Nevertheless, the results obtained are reasonable considering the conditions: the system remains resilient to failures and can always respond to requests even under stress. Another aspect to consider is how the throughput slowly gets worst and flips between different thresholds. This is a consequence of the increase in concurrent maintainers updating the same data and the number of requests to resolve, which causes concurrency issues that slightly affect the final performance.

The plots at the bottom highlights the read and write operations, and they keep slowly deteriorating. That is, we expect that we can be more efficient at maximum availability. In contrast, the plot at the bottom related to the Update operation shows an apparent worsening trend in the condition. The explanation for this is what was already mentioned before. It demonstrates that simultaneous update interaction of multiple data maintainers (higher value of t) causes longer wait times on the ledger, most likely related to ledger access conflict management.

In the best case, the DLT should establish about 8 concurrent network connections per node, as performance can be assumed to degrade beyond this number. This ensures latencies of about less than 4 seconds on average. By increasing the requests, we fall into the worst case where the average latency could double.

Chapter 7

Conclusions

This chapter reflects all the work developed throughout this thesis. We provide a summary of the conclusions on developing our contributions. Then, we provide a list of the main challenges encountered throughout the development of this work, as well as the methodology used to overcome them. At the end of the chapter, we describe possible future work that can be done to improve the solution developed, further supporting the process of sharing health data.

7.1 Summary

The objective of this work is to help decentralizing the Internet of Medical Things by leveraging the properties of DLTs in order to support the process of sharing and tracing health data. We wanted to implement the approach in a prototype that should be as customizable as possible to adapt to as many use cases as possible.

Therefore, to achieve the desired objectives, we first reviewed the healthcare field to identify the current issues, as presented in Chapter 2. We conducted a review to identify the solutions that attempt to solve the issues identified in the current system of sharing health research data.

Taking into account the current issues of the process of sharing health research data and the aspects that the solutions of the state of the art do not encompass, we formulated the following hypothesis: *"Providing the* *ability to trace data transformations without the need of trust in a central authority, can support the interests of the different parties involved and increase cooperation so that entities will have confidence over the data processing procedures of each other."* Through this statement, we were able to formulate several research questions which support the proposed solution.

We then provided our second contribution by introducing an application in the area of IoMT that would collect user-sensitive health data and build its backend for data collection called Balance. This application is supported by research work and it is a real use case in the IoMT, being available for download on the app store and freely accessible in its source code.

We then introduced our third contribution, the InterPlanetary Health Layer. Its main objective is to leverage the main conclusions of the previous analysis presented as our first main contribution and use blockchain to architect an approach encompassing all the desired aspects for solving the issues presented. The approach is described in detail in Chapter 4. The approach was implemented in a fully functioning prototype by building a test network with IBM HyperLedger, with the objective to be as customizable as possible. Hence, it adapts to multiple use cases and is open for expansion, allowing further research and improvements to be developed.

Following previous works, we tested the architecture from different angles, taking into consideration critical aspects of a decentralized system such as: security in sharing data, ensuring data availability, and delving into further support for data management according to the world's most stringent regulations. We also hypothesized the use of social networks with the purpose of enabling use cases where shared participation is required and with the dual purpose of having users participate in maintaining the system. Lastly, we provide an evaluation of all the work developed through focused experiments on real-world contexts and with use cases from the real world.

In the course of conducting our research, we successfully addressed and provided answers to the research questions that were initially raised:

- Can we provide a mobile application for collecting health data to be used as a use case in the Internet of Medical Things ecosystem? We have been able to provide a mobile application for collecting health data to be used as a use case in the Internet of Medical Things ecosystem called Balance. This would enable users to easily collect and manage their health data related to postural stability.

- Can we enable the ability to track data and determine their provenance without a central authority? By employing the proposed IPHL, which enables decentralized tracking of data and ensures its integrity, we ensured the ability to track data and determine their provenance without a central authority.

- Can we enable users to manage their data according to the regulations by introducing decentralized technologies? We could enable users to manage their data according to the regulations by introducing decentralized technologies. By using decentralized technologies, users can have greater control over their health data and manage them in a secure and transparent way.

- Can we increase users' awareness about their precious health assets and the availability of health data by introducing social networks? In our work, we have shown that it could be possible to increase users' awareness about their precious health assets by introducing social networks. Moreover, social networks could be a catalyst for sharing even more information and increase the availability of health data by letting users to act as DLT network maintainers.

Based on the works produced, we supported that providing a decentralized health layer improves the process of sharing health data, incentivizing cooperation in the ecosystem. In this sense, entities can be more confident of the processing procedures of each other without the need for trust in a central authority. This supports the interests of the different entities in a system where there are multiple entities with competing interests involved.

7.2 Contributions

The resulting analysis of both the current issues with the system and with the state of the art is our first main contribution which we evaluate through interviews with experts in the fields. We listed the current issues with the process of sharing health research data as well as some problems with the current platforms, which our solution aimed to solve. In order to solve these problems, we have built an approach that was then implemented in a prototype that works as a proof of concept, supporting the feasibility of the approach. Therefore, the contributions of this dissertation are:

- We provide a view on the implications of DLTs in the IoMT field, exploring the health research system's current aspects to identify its main issues.

- We provide a new shared, agnostic, and permissioned decentralized data layer with enhanced data availability. We use decentralized technologies for this purpose: a DFS layer as a medium of storage, an experimental DLT to provide smart contract functionality and tracing capabilities, smart contracts to manage access policies, and authentication mechanisms to manage user data;

- We implement the proposed architecture on a real-world use case represented by a traditional IoMT application called Balance connecting to the IPHL implementation;

- We provide experimental results of the work, demonstrating the feasibility of such an implementation;

- We propose the development of a social network on top of the IPHL for the dual purpose of increasing the availability of data and the accountability of individuals in maintaining the system.

7.3 Challenges

Throughout the development of this work, we had several challenges to overcome related to understanding the aspects and concepts of the field of health research data. Finding feasible current solutions through the literature review was also challenging. Finally, fully decentralizing the system was challenging because of many different matters. Consequently, through the formulation of a solution hypothesis, we developed a solution that encompasses several aspects, concepts, and features:

- Traceability of decentralized data: this is the first feature being implemented in the solution. It leverages the decentralized properties of DLTs to create a decentralized registry of traceability data, providing trust over the immutability of the data.

- Data auditing: the second feature implemented in the solution aims to provide a way for entities to be able to audit the traceability data (say whether it is valid or not) in order to further support the process of building trust by keeping all entities engaged on it cooperatively while also providing a computationally easier method to verify the information stored on the DLT.

- User-centered: a necessary architectural design choice for the overall system. Users can be active participant actors in the system and involved in their data management process. It is important since the process of verifying the traceability data is computationally hard, requiring a high incentive to be performed.

In addition, there are also several challenges related to technology that is not yet ready. Balance uncovers all the current issues with the health data and the problems with the solutions reviewed in the state of the art analysis. Understanding the system of sharing health data was the first challenge in developing this project.

The decentralization of the IoMT currently faces a technological hurdle in the form of mobile devices. The biggest problems are computational power and energy usage. To get around the obstacle and provide a practical implementation, we proposed a technique that decouples smartphones from distributed technology. At the same time, it is hoped that the decentralized web will progress in incorporating these technologies. Because mobile devices are the most widely utilized and may provide even greater availability of individual nodes, this stage is crucial. We believe that in the long term, these technologies will unquestionably make up the decentralized web of the future.

Another issue is the DLT growing pace. A DLT's ledger requires ongoing maintenance, which adds to its integrity costs. For these particular solutions based on mobile devices, additional research should be done on how to handle the storage as it grows over time.

7.4 Remarks and Future Directions

The work focused on the ability of blockchain to create specific solutions to enable the users to be the true owner of their data. What this thesis underlined is how these problems could be solved together, potentially paving the way to deal with the problem of data management for healthcare.

Moving the architectures from a system-centric paradigm, where a user is the consumer of the application, to a user-centric paradigm, where the user is more than a consumer but an active participant, basically consists in moving to the blockchain economy, that is to say: networks as a medium of active contribution to a community, as the one of the healthcare. This could be more interesting when these networks reward the participants and the ability to interact with each other.

This vision also aligns very well with the concept of rewards for data contribution. As said throughout the thesis, whenever an individual uses an IoMT system or an IoT device, he is never rewarded for the contribution it makes. Normally individuals pay to get professional help, but their contribution is higher than the mere performance received. In fact, they could also contribute to enhancing scientific knowledge. Today, this information is basically lost by not contributing or, when it occurs, does not mean reward. The Internet of blockchains could enable data sharing and crowdsourcing in an increasingly blockchain-enabled world.

Similarly to most research projects, some improvements can be made as future work to this work. Some of the improvements to the solution are: (i) Support for penalties for the entities that perform incorrect behavior. In this sense, entities are incentivized to learn from their mistakes so that they do not compromise the correction of the consensus of the system. (ii) Test the minimum voting threshold is the minimum number of votes necessary that, together with the minimum ratio between approvals and rejections, form the condition necessary to terminate the voting round.

Therefore, there was the need to architect a reward system to incentivize entities to verify each other's traceability data and to be honest in the process. In order to achieve honest behavior, rewards and penalties are issued to the entities in an approach similar to what available on blockchain protocols.

This leads to the idea of reputation as a good incentive resource to improve cooperation in the system. The doubts reside in determining whether reputation is a feasible consensus resource to support the approved incentive mechanism since trust over an incentive system can only be achieved if the resource supporting the structure has important value for the entities.

Moreover, modifying the architecture to allow the use of IoMT data with machine learning applications would be of great importance. These layers would enable advantageous purposes, such as advancements in the medical area and feeding future artificial intelligence, because these data are excluded from current datasets and those supplied by research organizations are not very thorough.

In addition to these advancements, the introduction of a Decentralized Identity (DID) could give users a clear means to identify themselves in such a network and keep track of comparable networks in other places. Future introduction of a Self-Sovereign identity, in our opinion, might offer a special reference technology for decentralized access to various data layers.

7.5 Conclusions

Nowadays, the heterogeneous resources, the massive amount of data coming from mobile devices, and people's privacy needs suggest new methods for storing data for effective sharing.

We have developed a solution based on a combination of DFS and DLT capable of ensuring communication, sharing, and participation. Combining the two allowed us to store and share data between trusted individuals without relying on a centralized entity.

Substantial help in going forward with the implementation came from OrbitDB; creating a layer above IPFS allowed the usage of IPFS as a database, making meaningful and complex queries. Starting from these technologies, we envisioned the InterPlanetary Health Layer through which research centers and institutions could safely retrieve personal medical data. The proposed implementation, called *Halo Network* has been extensively tested by connecting it with a modified IoMT application called Balance. The results confirmed the feasibility of the proposed solution showing good scalability and a modest impact on the application's performance. Moreover, the ability of users to create their network makes it possible to ensure the availability of data that otherwise, in a decentralized context, would remain doubtful at any instant. Such a network guarantees that the stakeholder always gains access to user data and avoids a single point of failure.

Assuming that the field will progress, we hope this work will incentivize new research works and implementations that allow the free flow of information and adequate tracking of information, which are also accessible by external authorities such as entities and institutes. Our vision is that these technologies, along with self-sovereign identities, will push further the development of increasingly secure user-centric applications. Moreover, the proposed InterPlanetary Health Layer could be used for beneficial purposes as medical advancements and as a data layer able to feed the future artificial intelligence that will need data to be used continuously.

Well-being is the foundation for the lifestyle of a healthy individual, and managing medical data could help users achieve this goal. In the past, the need for a large amount of data and privacy issues were not considered: the traditional method for data collection was through recordings on paper, and medical science was not supported by existing technology, usually leading to no explanation for several diseases and no solution. Quickly with the introduction of technology has become clear how it was possible to deliver and discover new solutions and how much the data collected by patient monitoring were helpful for him and the healthcare community. By introducing the Internet of Medical Things, potentially, any data collected by a user could be exploited with a specific goal.

Blockchain is finally a true candidate to radically change healthcare data management solutions. Through achieving specific objectives such as security, scalability, and interoperability, the blockchain can be the driving technology to develop lasting and independent data-sharing platforms that can give value to privacy.

Bibliography

[1] Alaa Awad Abdellatif, Lutfi Samara, Amr Mohamed, Aiman Erbad, Carla Fabiana Chiasserini, Mohsen Guizani, Mark Dennis O'Connor, and James Laughton. Medge-chain: Leveraging edge computing and blockchain for efficient medical data exchange. *IEEE Internet of Things Journal*, 2021.

[2] Ryno Adlam and Bertram Haskins. A permissioned blockchain approach to the authorization process in electronic health records. In *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, pages 1–8. IEEE, 2019.

[3] Cornelius C Agbo, Qusay H Mahmoud, and J Mikael Eklund. Blockchain technology in healthcare: a systematic review. In *Healthcare*, volume 7, page 56. Multidisciplinary Digital Publishing Institute, 2019.

[4] Tareq Ahram, Arman Sargolzaei, Saman Sargolzaei, Jeff Daniels, and Ben Amaba. Blockchain technology innovations. In *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*, pages 137–141. IEEE, 2017.

[5] Shiroq Al-Megren, Shada Alsalamah, Lina Altoaimy, Hessah Alsalamah, Leili Soltanisehat, Emad Almutairi, et al. Blockchain use cases in digital sectors: A review of the literature. 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 1417–1424. IEEE, 2018.

[6] Fadi Al-Turjman, Muhammad Hassan Nawaz, and Umit Deniz Ulusar. Intelligence in the internet of medical things era: a systematic review of current and future trends. *Computer Communications*, 150:644–660, 2020.

[7] Andreas M Antonopoulos. *Mastering Bitcoin: unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.", 2014.

[8] Rajakumar Arul, Yasser D Al-Otaibi, Waleed S Alnumay, Usman Tariq, Umar Shoaib, and MD Jalil Piran. Multi-modal secure healthcare data dissemination framework using blockchain in iomt. *Personal and Ubiquitous Computing*, pages 1–13, 2021.

[9] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30, 2006.

[10] Kebira Azbeg, Ouail Ouchetto, Said Jai Andaloussi, Leila Fetjah, and Abderrahim Sekkaki. Blockchain and iot for security and privacy: A platform for diabetes self-management. In *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, pages 1–5. IEEE, 2018.

[11] Adam Back et al. Hashcash-a denial of service counter-measure, 2002.

[12] Safa Bahri, Nesrine Zoghlami, Mourad Abed, and Joao Manuel RS Tavares. Big data for healthcare: a survey. *IEEE access*, 7:7397–7408, 2018.

[13] Mandrita Banerjee, Junghee Lee, and Kim-Kwang Raymond Choo. A blockchain future for internet of things security: a position paper. *Digital Communications and Networks*, 4(3):149–160, 2018.

[14] Luigi Baratto, Pietro G Morasso, Cristina Re, and Gino Spada. A

new look at posturographic analysis in the clinical context: sway-density versus other parameterization techniques. *Motor control*, 6 (3):246–270, 2002.

[15] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[16] Tim Berners-Lee. Linked data-design issues. *http://www. w3. org/DesignIssues/LinkedData. html*, 2006.

[17] Tim Berners-Lee and Mark Fischetti. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. Harper San Francisco, 1999.

[18] Gioele Bigini, Valerio Freschi, and Emanuele Lattanzi. A review on blockchain for the internet of medical things: Definitions, challenges, applications, and vision. *Future Internet*, 12(12):208, 2020.

[19] Gioele Bigini, Valerio Freschi, Alessandro Bogliolo, and Emanuele Lattanzi. Decentralising the internet of medical things with distributed ledger technologies and off-chain storages: A proof of concept. In *International Conference on Smart Objects and Technologies for Social Good*, pages 80–90. Springer, 2021.

[20] Ashly D Black, Josip Car, Claudia Pagliari, Chantelle Anandan, Kathrin Cresswell, Tomislav Bokun, Brian McKinstry, Rob Procter, Azeem Majeed, and Aziz Sheikh. The impact of ehealth on the quality and safety of health care: a systematic overview. *PLoS medicine*, 8(1):e1000387, 2011.

[21] Plamenka Borovska. Big data analytics and internet of medical things make precision medicine a reality. *International Journal of Internet of Things and Web Services*, 3, 2018.

[22] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437*, 2017.

[23] Ting Cai, Zetao Yang, Wuhui Chen, Zibin Zheng, and Yang Yu. A blockchain-assisted trust access authentication system for solid. *IEEE Access*, 8:71605–71616, 2020.

[24] Wolfie Christl, Katharina Kopp, and Patrick Urs Riechert. 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*, 2017.

[25] Emeka Chukwu and Lalit Garg. A systematic review of blockchain in healthcare: Frameworks, prototypes, and implementations. *IEEE Access*, 8:21196–21214, 2020.

[26] Benjamin Cisneros, Jiani Ye, Chol Hyun Park, and Yoohwan Kim. Covireader: Using iota and qr code technology to control epidemic diseases across the us. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0610–0618. IEEE, 2021. doi: 10.1109/CCWC51732.2021.9376093.

[27] European Commission. European commission. complete guide to gdpr compliance. available online: https://gdpr.eu, 2016.

[28] Jason Paul Cruz, Yuichi Kaji, and Naoto Yanai. Rbac-sc: Role-based access control using smart contract. *IEEE Access*, PP:1–1, 2018. doi: 10.1109/ACCESS.2018.2812844.

[29] Maryam Davari and Elisa Bertino. 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, 2019.

[30] Paul De Hert, Vagelis Papakonstantinou, Gianclaudio Malgieri, Laurent Beslay, and Ignacio Sanchez. The right to data portability in the gdpr: Towards user-centric interoperability of digital services. *Computer law & security review*, 34(2):193–203, 2018.

[31] Tushar Dey, Shaurya Jaiswal, Shweta Sunderkrishnan, and Neha Katre. Healthsense: A medical use case of internet of things and blockchain. In *2017 International conference on intelligent sustainable systems (ICISS)*, pages 486–491. IEEE, 2017.

[32] Alyssa Donawa, Inema Orukari, and Corey E Baker. Scaling blockchains to support electronic health records for hospital systems. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0550–0556. IEEE, 2019. doi: 10.1109/UEMCON47517.2019.8993101.

[33] Marcos Duarte, SMSF Freitas, and Vladimir Zatsiorsky. Control of equilibrium in humans-sway over sway. *Motor Control, Oxford University Press, Oxford*, pages 219–242, 2011.

[34] Ashutosh Dhar Dwivedi, Gautam Srivastava, Shalini Dhar, and Rajani Singh. A decentralized privacy-preserving healthcare blockchain for iot. *Sensors*, 19(2):326, 2019. doi: 10.3390/s19020326.

[35] Bhaskara S Egala, Ashok K Pradhan, Venkata R Badarla, and Saraju P Mohanty. Fortified-chain: a blockchain based framework for security and privacy assured internet of medical things with effective access control. *IEEE Internet of Things Journal*, 2021.

[36] Michael Egorov, MacLane Wilkison, and David Nuñez. Nucypher kms: decentralized key management system. *arXiv preprint arXiv:1707.06140*, 2017.

[37] Beatriz Esteves, Harshvardhan J Pandit, and Víctor Rodríguez-Doncel. 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, 2021.

[38] Beatriz Esteves, VÃctor RodrÃguez-Doncel, Harshvardhan J Pandit, and Pat McBennett. Using the ODRL profile for access control

for solid pod resource governance. In *To Appear on The Semantic Web: ESWC 2022 Satellite Events proceedings*, 2022.

[39] European Commission. A european strategy for data, 2020.

[40] European Union Agency for Cybersecurity. Data Pseudonymisation: Advanced Techniques & Use Cases. Online https://www.enisa.europa.eu/publications/data-pseudonymisation-advanced-techniques-and-use-cases. Technical report, European Union Agency for Cybersecurity, 2021.

[41] Andressa Fernandes, Vladimir Rocha, Arlindo F da Conceição, and Flavio Horita. Scalable architecture for sharing ehr using the hyperledger blockchain. In *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*, pages 130–138. IEEE, 2020.

[42] Tiago M Fernández-Caramés and Paula Fraga-Lamas. A review on the use of blockchain for the internet of things. *Ieee Access*, 6: 32979–33001, 2018.

[43] Tiago M Fernández-Caramés, Iván Froiz-Míguez, Oscar Blanco-Novoa, and Paula Fraga-Lamas. Enabling the internet of mobile crowdsourcing health things: A mobile fog computing, blockchain and iot based continuous glucose monitoring system for diabetes mellitus research and care. *Sensors*, 19(15):3319, 2019. doi: 10.3390/s19153319.

[44] Michèle Finck and Frank Pallas. They who must not be identified—distinguishing personal from non-personal data under the GDPR. *International Data Privacy Law*, 10(1):11–36, 2020. ISSN 2044-3994. doi: 10.1093/idpl/ipz026.

[45] Marco Furini, Silvia Mirri, Manuela Montangero, and Catia Prandi. Privacy perception when using smartphone applications. *Mobile Networks and Applications*, 25:1055–1061, 2020. ISSN 1572-8153. doi: 10.1007/s11036-020-01529-z.

[46] Neha Garg, Mohammad Wazid, Ashok Kumar Das, Devesh Pratap Singh, Joel JPC Rodrigues, and Youngho Park. Bakmp-iomt: Design of blockchain enabled authenticated key management protocol for internet of medical things deployment. *IEEE Access*, 8:95956–95977, 2020.

[47] Raúl Gracia-Tinedo, Marc S'nchez-Artigas, and Pedro Garcia-Lopez. F2box: Cloudifying f2f storage systems with high availability correlation. In *2012 IEEE Fifth International Conference on Cloud Computing*, pages 123–130. IEEE, 2012.

[48] Barbara Guidi, Andrea Michienzi, and Laura Ricci. Data persistence in decentralized social applications: The ipfs approach. In *Consumer Communications & Networking Conference (CCNC)*, pages 1–4. IEEE, 2021.

[49] Stuart Haber and W Scott Stornetta. How to time-stamp a digital document. In *Conference on the Theory and Application of Cryptography*, pages 437–455. Springer, 1990.

[50] Anton Hasselgren, Katina Kralevska, Danilo Gligoroski, Sindre A Pedersen, and Arild Faxvaag. Blockchain in healthcare and health sciences - a scoping review. *International Journal of Medical Informatics*, 134:104040, 2020.

[51] J. Herranz, D. Hofheinz, and Eike Kiltz. Kem/dem: Necessary and sufficient conditions for secure hybrid encryption. *IACR Cryptology ePrint Archive*, 2006.

[52] Marko Hölbl, Marko Kompara, Aida Kamišalić, and Lili Nemec Zlatolas. A systematic review of the use of blockchain in healthcare. *Symmetry*, 10(10):470, 2018.

[53] HM Hussien, SM Yasin, SNI Udzir, AA Zaidan, and BB Zaidan. A systematic review for enabling of develop a blockchain technology in healthcare application: Taxonomy, substantially analysis, motivations, challenges, recommendations and future direction. *Journal of medical systems*, 43(10):320, 2019.

[54] Renato Ianella. Open digital rights language (odrl). *Open Content Licensing: Cultivating the Creative Commons*, 2007.

[55] J Indumathi, Achyut Shankar, Muhammad Rukunuddin Ghalib, J Gitanjali, Qiaozhi Hua, Zheng Wen, and Xin Qi. Block chain based internet of medical things for uninterrupted, ubiquitous, user-friendly, unflappable, unblemished, unlimited health care services (bc iomt u 6 hcs). *IEEE Access*, 8:216856–216872, 2020.

[56] IPLD Team. Interplanetary linked data (ipld). *https://specs.ipld.io/*, 2016.

[57] Jim Isaak and Mina J Hanna. User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer*, 51(8):56–59, 2018.

[58] Mayssa Jemel and Ahmed Serhrouchni. 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, 2017.

[59] Shan Jiang, Jiannong Cao, Hanqing Wu, Yanni Yang, Mingyu Ma, and Jianfei He. Blochie: a blockchain-based platform for healthcare information exchange. In *2018 ieee international conference on smart computing (smartcomp)*, pages 49–56. IEEE, 2018.

[60] Seyednima Khezr, Md Moniruzzaman, Abdulsalam Yassine, and Rachid Benlamri. Blockchain technology in healthcare: A comprehensive review and directions for future research. *Applied Sciences*, 9(9):1736, 2019.

[61] David Koll, Jun Li, and Xiaoming Fu. Soup: an online social network by the people, for the people. In *Proceedings of the 15th International Middleware Conference*, pages 193–204, 2014.

[62] Suebtrakul Kongruangkit, Yu Xia, Xiwei Xu, and Hye-young Paik.

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, 2021.

[63] Hossain Kordestani, Kamel Barkaoui, and Wagdy Zahran. Hapichain: a blockchain-based framework for patient-centric telemedicine. In *2020 IEEE 8th International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–6. IEEE, 2020. doi: 10.1109/SeGAH49190.2020.9201726.

[64] Mirko Koscina, David Manset, Claudia Negri, and Octavio Perez. 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, 2019.

[65] Mahender Kumar and Satish Chand. Medhypchain: A patient-centered interoperability hyperledger-based medical healthcare system: Regulation in covid-19 pandemic. *Journal of Network and Computer Applications*, 179:102975, 2021.

[66] Randhir Kumar and Rakesh Tripathi. Towards design and implementation of security and privacy framework for internet of medical things (iomt) by leveraging blockchain and ipfs technology. *The Journal of Supercomputing*, pages 1–40, 2021.

[67] Randhir Kumar, Ningrinla Marchang, and Rakesh Tripathi. Distributed off-chain storage of patient diagnostic reports in healthcare system using ipfs and blockchain. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 1–5. IEEE, 2020. doi: 10.1109/COMSNETS48256.2020.9027313.

[68] Emanuele Lattanzi, Valerio Freschi, Saverio Delpriori, Lorenz Cuno

Klopfenstein, and Alessandro Bogliolo. Standing balance assessment by measurement of body center of gravity using smartphones. *IEEE Access*, 8:96438–96448, 2020. doi: 10.1109/ACCESS.2020.2996251.

[69] Jinyang Li and Frank Dabek. F2f: Reliable storage in open networks. In *IPTPS*, 2006.

[70] Dongtao Liu, Amre Shakimov, Ramón Cáceres, Alexander Varshavsky, and Landon P Cox. Confidant: Protecting osn data without locking it up. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 61–80. Springer, 2011.

[71] Markus Lücking, Raphael Manke, Markus Schinle, Lukas Kohout, Stefan Nickel, and Wilhelm Stork. Decentralized patient-centric data management for sharing iot data streams. In *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6. IEEE, 2020. doi: 10.1109/COINS49042.2020.9191653.

[72] Tim K Mackey, Tsung-Ting Kuo, Basker Gummadi, Kevin A Clauson, George Church, Dennis Grishin, Kamal Obbad, Robert Barkovich, and Maria Palombini. 'fit-for-purpose?'–challenges and opportunities for applications of blockchain technology in the future of healthcare. *BMC medicine*, 17(1):68, 2019.

[73] Mohammad Moussa Madine, Ammar Ayman Battah, Ibrar Yaqoob, Khaled Salah, Raja Jayaraman, Yousof Al-Hammadi, Sasa Pesic, and Samer Ellahham. Blockchain for giving patients control over their medical records. *IEEE Access*, 8:193102–193115, 2020. doi: 10.1109/ACCESS.2020.3032553.

[74] Damiano Di Francesco Maesa, Paolo Mori, and Laura Ricci. Blockchain based access control. In *IFIP international conference on distributed applications and interoperable systems*, pages 206–220. Springer, 2017.

[75] Martina Mancini, Arash Salarian, Patricia Carlson-Kuhta, Cris Zampieri, Laurie King, Lorenzo Chiari, and Fay B Horak. Isway: a sensitive, valid and reliable measure of postural control. *Journal of neuroengineering and rehabilitation*, 9(1):1–8, 2012.

[76] Vinodhini Mani, Prakash Manickam, Youseef Alotaibi, Saleh Alghamdi, and Osamah Ibrahim Khalaf. Hyperledger healthchain: Patient-centric ipfs-based storage of health records. *Electronics*, 10 (23):3003, 2021.

[77] Raghavendra K Marangappanavar and M Kiran. Inter-planetary file system enabled blockchain solution for securing healthcare records. In *2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, pages 171–178. IEEE, 2020. doi: 10.1109/ISEA-ISAP49340.2020.235016.

[78] Kei Masani, Albert H Vette, Motoki Kouzaki, Hiroaki Kanehisa, Tetsuo Fukunaga, and Milos R Popovic. Larger center of pressure minus center of gravity in the elderly induces larger body acceleration during quiet standing. *Neuroscience letters*, 422(3):202–206, 2007.

[79] Ahmad Akmaluddin Mazlan, Salwani Mohd Daud, Suriani Mohd Sam, Hafiza Abas, Siti Zaleha Abdul Rasid, and Muhammad Fathi Yusof. Scalability challenges in healthcare blockchain system—a systematic review. *IEEE Access*, 8:23663–23673, 2020.

[80] Menno Mostert, Annelien L Bredenoord, Monique CIH Biesaart, and Johannes JM Van Delden. Big data in medical research and eu data protection law: challenges to the consent or anonymise approach. *European Journal of Human Genetics*, 24(7):956–960, 2016.

[81] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.*, 21260:1–9, 2008.

[82] Satoshi Nakamoto. Bitcoin p2p e-cash paper. *The Cryptography Mailing list at metzdowd.com*, 31:2008, 2008.

[83] N Nanayakkara, Malka Halgamuge, and Ali Syed. Security and privacy of internet of medical things (iomt) based healthcare applications: A review, 2019.

[84] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and cryptocurrency technologies: a comprehensive introduction.* Princeton University Press, 2016.

[85] José Roberto Nascimento Jr, José BS Nunes, Eduardo Lucena Falcão, Lilia Sampaio, and Andrey Brito. On the tracking of sensitive data and confidential executions. In *Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems*, pages 51–60, 2020. doi: 10.1145/3401025.3404097.

[86] Nataliia Neshenko, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum, and Nasir Ghani. Demystifying iot security: an exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. *IEEE Communications Surveys & Tutorials*, 21(3):2702–2733, 2019.

[87] Dinh C Nguyen, Khoa D Nguyen, and Pubudu N Pathirana. A mobile cloud based iomt framework for automated health assessment and management. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6517–6520. IEEE, 2019.

[88] Dinh C Nguyen, Pubudu N Pathirana, Ming Ding, and Aruna Seneviratne. Blockchain for secure ehrs sharing of mobile cloud based e-health systems. *IEEE access*, 7:66792–66806, 2019.

[89] David Nunez. Umbral: A threshold proxy re-encryption scheme. online at: https://raw.githubusercontent.com/nucypher/umbral-doc/master/umbral-doc.pdf, 2018.

[90] Alexandra Olteanu and Guillaume Pierre. Towards robust and scalable peer-to-peer social networks. In *Proceedings of the Fifth Workshop on Social Network Systems*, pages 1–6, 2012.

[91] Ilhaam A Omar, Raja Jayaraman, Khaled Salah, Mecit Can Emre Simsekler, Ibrar Yaqoob, and Samer Ellahham. Ensuring protocol compliance and data transparency in clinical trials using blockchain smart contracts. *BMC Medical Research Methodology*, 20(1):1–17, 2020. doi: 10.1186/s12874-020-01109-5.

[92] Md Mehedi Hassan Onik, Chul-Soo Kim, Nam-Yong Lee, and Jinhong Yang. Privacy-aware blockchain for personal data sharing and tracking. *Open Computer Science*, 9(1):80–91, 2019.

[93] Deepa Pavithran, Khaled Shaalan, Jamal N Al-Karaki, and Amjad Gawanmeh. Towards building a blockchain framework for iot. *Cluster Computing*, pages 1–15, 2020.

[94] Marc Pilkington. Can blockchain improve healthcare management? consumer medical electronics and the iomt. *Consumer Medical Electronics and the IoMT (August 24, 2017)*, 2017.

[95] Manoharan Ramachandran, Niaz Chowdhury, Allan Third, Zeeshan Jan, Chris Valentine, and John Domingue. A framework for handling internet of things data with confidentiality and blockchain support, 2020.

[96] Fortune Business Rights. Fortune business rights. internet of medical things (iomt) market size, share & industry analysis, by product type (stationary medical devices, implanted medical devices, wearable external medical devices), by application (telemedicine, medication management, patient monitoring, others), by end user (healthcare providers, patients, government authorities, others) and regional forecast, 2019-2026. available online: https://www.fortunebusinessinsights.com/industry-reports/internet-of-medical-things-iomt-market-101844, 2020.

[97] Tharuka Rupasinghe, Frada Burstein, Carsten Rudolph, and Steven Strange. Towards a blockchain based fall prediction model for aged care. In *Proceedings of the Australasian Computer Science Week Multiconference*, pages 1–10, 2019.

[98] Arijit Saha, Ruhul Amin, Sourav Kunal, Satyanarayana Vollala, and Sanjeev K Dwivedi. Review on "blockchain technology based medical healthcare system with privacy issues". *Security and Privacy*, 2(5):e83, 2019.

[99] Andrei Vlad Sambra, Essam Mansour, Sandro Hawke, Maged Zereba, Nicola Greco, Abdurrahman Ghanem, Dmitri Zagidulin, Ashraf Aboulnaga, and Tim Berners-Lee. Solid: a platform for decentralized social applications based on linked data. *MIT CSAIL & Qatar Comput. Res. Inst., Ar-Rayyan, Qatar, Tech. Rep. MIT-QCRI-2016*, 2016.

[100] Emil Saweros and Yeong-Tae Song. Connecting personal health records together with ehr using tangle. In *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 547–554. IEEE, 2019. doi: 10.1109/SNPD.2019.8935646.

[101] Mohamed Seliem and Khalid Elgazzar. Biomt: Blockchain for the internet of medical things. In *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–4. IEEE, 2019. doi: 10.1109/BlackSeaCom.2019.8812784.

[102] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-free replicated data types. In *Symposium on Self-Stabilizing Systems*, pages 386–400. Springer, 2011.

[103] Munish Sharma. Sharif: Solid pod based secured healthcare information storage and exchange solution, 2021.

[104] Gautam Srivastava, Jorge Crichigno, and Shalini Dhar. A light and secure healthcare blockchain for iot medical devices. In *2019 IEEE Canadian conference of electrical and computer engineering (CCECE)*, pages 1–5. IEEE, 2019.

[105] Charalampos Stamatellis, Pavlos Papadopoulos, Nikolaos Pitropakis, Sokratis Katsikas, and William J Buchanan. A

privacy-preserving healthcare framework using hyperledger fabric. *Sensors*, 20(22):6587, 2020.

[106] Ralf Steinmetz and Klaus Wehrle. *Peer-to-peer systems and applications*, volume 3485. Springer, 2005.

[107] Thitinan Tantidham and Yu Nandar Aung. Emergency service for smart home system using ethereum blockchain: system and architecture. In *2019 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops)*, pages 888–893. IEEE, 2019. doi: 10.1109/PERCOMW.2019.8730816.

[108] Maria Trojano, Roberto Bergamaschi, Maria Pia Amato, Giancarlo Comi, Angelo Ghezzi, Vito Lepore, Maria Giovanna Marrosu, Paola Mosconi, Francesco Patti, Michela Ponzio, et al. The italian multiple sclerosis register. *Neurological Sciences*, 40(1):155–165, 2019.

[109] Mueen Uddin, MS Memon, Irfana Memon, Imtiaz Ali, Jamshed Memon, Maha Abdelhaq, and Raed Alsaqour. Hyperledger fabric blockchain: Secure and efficient solution for electronic health records. *Comput., Mater. Continua*, 68(2):2377–2397, 2021.

[110] Vincent T Van Hees, Lukas Gorzelniak, Emmanuel Carlos Dean León, Martin Eder, Marcelo Pias, Salman Taherian, Ulf Ekelund, Frida Renström, Paul W Franks, Alexander Horsch, et al. Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity. *PloS one*, 8(4):e61691, 2013.

[111] Junchao Wang, Kaining Han, Anastasios Alexandridis, Zhiyu Chen, Zeljko Zilic, Yu Pang, Gwanggil Jeon, and Francesco Piccialli. A blockchain-based ehealthcare system interoperating with wbans. *Future Generation Computer Systems*, 2019.

[112] Jie Xu, Kaiping Xue, Shaohua Li, Hangyu Tian, Jianan Hong, Peilin Hong, and Nenghai Yu. Healthchain: A blockchain-based privacy preserving scheme for large-scale health data. *IEEE Internet of Things Journal*, 6(5):8770–8781, 2019. doi: 10.1109/JIOT.2019.2923525.

[113] Zhu Yan, Guhua Gan, and Khaled Riad. 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, 2017.

[114] Jingyu Zhang, Siqi Zhong, Jin Wang, Lei Wang, Yaqiong Yang, Boyang Wei, and Guoyao Zhou. A review on blockchain-based systems and applications. In *International Conference on Internet of Vehicles*, pages 237–249. Springer, 2019.

[115] Mirko Zichichi and Gioele Bigini. miker83z/web5-health-data-sharing-tests: decentralized health data sharing tests, 2022. URL `https://doi.org/10.5281/zenodo.6636102`.

[116] Mirko Zichichi, Stefano Ferretti, and Gabriele D'Angelo. A distributed ledger based infrastructure for smart transportation system and social good. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2020.

[117] Mirko Zichichi, Stefano Ferretti, and Gabriele D'Angelo. On the efficiency of decentralized file storage for personal information management systems. In *Proc. of the 2nd International Workshop on Social (Media) Sensing, co-located with 25th IEEE Symposium on Computers and Communications 2020 (ISCC2020)*, pages 1–6. IEEE, 2020.

[118] Mirko Zichichi, Stefano Ferretti, and Gabriele D'Angelo. A framework based on distributed ledger technologies for data management and services in intelligent transportation systems. *IEEE Access*, pages 100384–100402, 2020.

[119] Mirko Zichichi, Stefano Ferretti, Gabriele D'Angelo, and Víctor Rodríguez-Doncel. Personal data access control through distributed authorization. In *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–4. IEEE, 2020.

[120] Mirko Zichichi, Stefano Ferretti, Gabriele D'Angelo, and Víctor Rodríguez-Doncel. Data governance through a multi-dlt architecture in view of the gdpr. *Cluster Computing*, pages 1–32, 2022.

[121] Haider Dhia Zubaydi, Yung-Wey Chong, Kwangman Ko, Sabri M Hanshi, and Shankar Karuppayah. A review on the role of blockchain technology in the healthcare domain. *Electronics*, 8 (6):679, 2019.

[122] Guy Zyskind, Oz Nathan, et al. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184. IEEE, 2015.