FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Blockchain-based Approach for Sharing Health Research Data

Dinis Filipe da Silva Trigo

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Filipe Correia

Second Supervisor: Bruno Tavares

July 23, 2020

Blockchain-based Approach for Sharing Health Research Data

Dinis Filipe da Silva Trigo

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Jorge Barbosa
External Examiner: Prof. Hélder Gomes
Supervisor: Prof. Filipe Correia

July 23, 2020

Abstract

The need for sharing health data among multiple parties has become evident in several applications. Research on large health data sets provide major opportunities for improving health systems and individual care. Traceability of the data transformations is required to guarantee the integrity of health information. Without traceability, patients and health companies cannot be confident enough to share health data, holding back disease treatment research projects. Furthermore, the competition between research entities, drives them to hold their data and procedures as secret to avoid loosing credit for it.

In this sense, this work aims to support health data sharing between research entities, providing awareness over what happens to health research data. Moreover, the decentralized storage of the traceability data, by leveraging the properties of blockchain, avoids the need of trust in a single entity, providing confidence over the immutability of the data in an environment where multiple entities with competitive interests are involved. We also provide a traceability data auditing system, powered by an incentive system, that encourages group auditing, further increasing cooperation in the ecosystem.

To achieve that, a study of the state of the art on data traceability was conducted in order to find the current most suitable solutions to the problems of the current system. A comparative analysis was performed, establishing the best trade-off between the mechanisms reviewed. This best trade-off, in order to achieve the best of both worlds to our context is the basis for architecting our approach to solve some of the problems in the current system of health research data. Also, we review the blockchain frameworks available, in order to select the one that most suits our goals.

In order to evaluate the feasibility and the benefits of this approach, we conducted live interviews with experts in the fields of blockchain and health research data sharing. The majority of the experts are members of a research project that requires data sharing between multiple entities with traceability. While the opinions of the experts support the feasibility of the approach, they also provide constructive criticism, with suggestions on how it could be further improved. The suggestion, by the experts, of features already possible to obtain by the user, through customization, support our success in providing a solution that adapts to multiple scenarios.

We consider that this dissertation has explored the concepts regarding health research data, leveraging the properties of blockchain to improve the process of sharing health research data. We have architected an approach that provides decentralized data traceability and a rewards system that incentivizes entities to audit each other's data, reducing the impacts of competition and further increasing cooperation. The approach was implemented in a prototype, providing a proof of concept of its feasibility.

Keywords: Blockchain, Data Traceability, Data Provenance, Data Lineage, Health Data Sharing, Health Research

Resumo

A necessidade de partilhar dados de saúde, tornou-se evidente em várias aplicações. A investigação que recorre a conjuntos destes dados, oferece grandes oportunidades para melhorar os sistemas de saúde e o atendimento individual. A rastreabilidade das transformações de dados é necessária, para garantir a integridade das informações. Sem rastreabilidade, utentes e entidades de saúde não podem ter confiança suficiente para partilhar os dados, atrasando o progresso de projetos de investigação. Além disso, a competitividade entre as entidades de pesquisa, leva-as a manter dados e procedimentos em segredo, para evitar perder o reconhecimento pelos mesmos.

Nesse sentido, este trabalho tem como objetivo apoiar a partilha de dados de saúde entre entidades de investigação, consciencializando-as sobre o que acontece com esses dados. Além disso, o armazenamento descentralizado dos dados de rastreabilidade, ao enaltecer as propriedades do blockchain, evita a necessidade de confiar numa única entidade, fornecendo confiança sobre a imutabilidade dos dados, num ambiente em que existem entidades com interesses competitivos. Também fornecemos um sistema de auditoria de dados de rastreabilidade, alimentado por um sistema de incentivo, que encoraja a auditoria de grupo, aumentando ainda mais a cooperação.

Para tal, foi realizado um estudo do estado da arte sobre rastreabilidade de dados, a fim de encontrar as soluções mais adequadas para os problemas do sistema atual de saúde. Uma análise comparativa foi realizada, estabelecendo o melhor dos dois mundos entre os mecanismos estudados, o que constituiu a base para arquitetar a nossa abordagem, com vista a resolver alguns dos problemas no vigente sistema de dados em saúde. Além disso, estudamos as estruturas de blockchain disponíveis, a fim de selecionar a que melhor se adequa aos nossos objetivos.

Para avaliar a viabilidade e os benefícios dessa abordagem, realizamos entrevistas com especialistas nas áreas de blockchain e investigação com dados de saúde. São, na sua maioria, membros de um projeto de investigação, que requer partilha de dados entre as várias entidades com rastreabilidade. As suas opiniões apoiam a viabilidade da abordagem, embora também constituam críticas construtivas, com sugestões sobre como a mesma pode ser melhorada. Sugeriram ainda, funcionalidades já possíveis de serem obtidas pelo utilizador, através da personalização, e que apoiam o nosso sucesso em fornecer uma solução que se adapta a vários cenários.

Consideramos que esta dissertação explorou os conceitos sobre dados de investigação na área da saúde, aproveitando as propriedades do blockchain, para melhorar o processo de partilha dos dados resultantes da investigação. Arquitetamos uma abordagem que fornece rastreabilidade descentralizada de dados e um sistema de recompensas, que incentiva as entidades a auditarem os dados umas das outras, reduzindo os impactos da concorrência e aumentando a cooperação, fator que poderá minimizar os custos e o tempo das pesquisas. A abordagem foi implementada num protótipo, fornecendo uma prova do conceito da sua viabilidade.

Palavras-chave: Blockchain, Rastreabilidade de Dados, Proveniência de Dados, Linhagem de Dados, Partilha de Dados de Saúde, Investigação de Saúde

Acknowledgements

Above all, I would like to express my special thanks to my Supervisor, Filipe Correia, and to my Co-Supervisor, Bruno Tavares for sharing their knowledge and for all the help always provided. I would also like to thank Professor Artur Rocha for being my Supervisor in the context of the research project, iReceptorPlus, where this dissertation fits and also Alexandre Almeida who has been not only a member of the iReceptorPlus team, but also has always been ready to help me, not only in the context of the project, but also in the dissertation.

To all my family and close friends, a special thanks for always being there when I needed them throughout the process of developing this dissertation.

Dinis Filipe da Silva Trigo

"Our greatest glory is not in never falling,
but in rising every time we fall."

Confucius

Contents

List of Figures

List of Tables

Abbreviations

P2P	Peer to Peer
DoS	Denial of Service
AES	Advanced Encryption Standard
ECC	Elliptic Curve Cryptography
EHR	Electronic Health Record
PoW	Proof of Work
PoS	Proof of Stake
PRNG	Pseudorandom Number Generator
CRUD	Create, Read, Update and Delete
UUID	Universal Unique Identifier
JSON	JavaScript Object Notation

Chapter 1

Introduction

This document is part of the development of a Master's thesis in Informatics and Computing Engineering at Faculty of Engineering of University of Porto (FEUP), done in partnership with the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC). This dissertation is part of a research project, the iReceptorPlus, which objective is to support the process of sharing health research data. Our contribution is focused on providing traceability of the health research data transformations.

1.1 Context

Nowadays, patients' health data is, like many other assets, stored and processed digitally, enabling the data to be accessed remotely by professionals very fast [97]. This allows for very important improvements in the health care industry as well as disease treatment research since health data may contain the keys to important breakthroughs [108]. However, if not handled correctly and with the right mechanisms and tools, we may not be able to achieve the full potential of this valuable digital asset. The health records are generated by *Health Companies* in the process of treating patients which makes them the owners of that data [112].

In the last few years, data regulations have become increasingly severe in protecting people's personal data [103]. This together with the appearance of solutions to provide decentralized data sharing without the need of trust in a central authority [73], is raising awareness of subjects to the importance and value of their personal data [90, 114].

In this sense, this work aims to support health data sharing between research entities, providing awareness over what happens to research data in an environment where multiple entities with different and competitive interests are involved. Traceability of health research data should help overcome these problems and increase cooperation in the system. Ultimately, this is expected to contribute positively to all the entities involved, leading to more progress in disease treatment research.

1.2 Motivation

Patients' health data is currently spread over multiple repositories, making the process of using it more difficult [2]. With this lack of organization, health professionals find it difficult to access patients' health data to provide them health care as well as for research projects to use it for disease treatment research [10].

In many cases, health companies are producing some of the most significant data, but they cannot make it available to other entities unless they can ensure its security and licensing. The competition between research entities, drives them to hold their data and procedures as secret to avoid loosing credit for it. This together with the increasing awareness of people to the value of their data raises the importance of providing a system to aid the sharing of health information while maintaining the its integrity and privacy when it is exchanged between different actors.

To fully enable the sharing of data it is often necessary to use a multilayer data security system, providing multiple levels of authentication, authorization, and auditing. Traceability and detection of any manipulation that may happen is required to guarantee the integrity of health information. Awareness of the processing procedures is also necessary in order to increase cooperation between the research entities but this awareness requires cooperation which requires trust. Without decentralized traceability and an incentive system, health companies cannot be confident enough to share health data, holding back disease treatment research projects.

1.3 Problem

This dissertation seeks to evaluate if providing a system with traceability of the data transformations resulting from the processing for scientific research for disease treatment will enhance the cooperation and, therefore, the progress of the ecosystem. The targeted ecosystem involves the entities that perform research on health data. The developed solution is expected to help the multiple entities of the system have awareness of the processing steps of each other. This awareness is provided by a decentralized system since there are multiple entities with different interests involved and it is necessary to avoid the need of trust. Ultimately, this is expected to further increase cooperation in the system, creating a virtuous cycle that improves the health of the ecosystem.

Thus, we have researched different solutions within the scientific community that tackle or explore the traceability problem such as platforms and projects. The large applicability of the context due to the magnitude of the medical ecosystem, together with lack of solutions related with health data sharing with support for traceability of the data transformations, demonstrates the innovative aspect of our approach.

Chapter 4 (p. 61) explores in detail the problem statement. Firstly, we go through the main issues of the current solutions for our problem. Then we identify the main hypothesis of this dissertation as well as the respective research questions. Lastly, we propose a solution to solve the problems and describe the respective validation methodology.

1.4 Objectives

The first objective of this dissertation is to explore the fields of blockchain and health data, in attempt to leverage the properties of blockchain in order to support the process of sharing personal health data. Our main focus is on understanding the interests of the research entities in order to support them and incentivize cooperation.

The second objective of this dissertation is to develop an approach that balances the interests of the different entities involved and increases cooperation between them, so that entities will have confidence over the data processing procedures of each others. We aim to incentivize cooperation and keep entities more involved in the process through an incentive system that should act as a feedback loop of cooperation increasing.

Thus, the main research questions seek to explore the possibility of improving the health data sharing process' quality by providing a system with traceability of data transformations.

Thereby, the main objectives of this dissertation are:

- Providing the ability to **trace** data transformations and to determine the data **provenance** without a central authority.
- Increase **awareness** of the processing procedures between the entities.
- Increase cooperation between the entities of the system.

We expect that the solution developed as well as its implementation description leads to breakthroughs in the processes of sharing health data.

Finally, we hope that this dissertation helps improving the process of sharing health research data, by leading to the discovery of important keys that could be in any person's health records, and lead to important breakthroughs in disease treatment in the future.

1.5 Contributions

Our first contribution will be an **analysis** of the current most suitable solutions to our problem in order to understand the advantages and disadvantages of each. Based on this analysis, we will **establish the best trade-off** between the key aspects of the state of the art as well as point out the problems with some of those aspects alone. Then, we will analyse the current frameworks available to develop blockchain applications, in order to **select** the one that suits best the objective we want to achieve.

As a result of the state of the art analysis, we **present an approach** that leverages the best trade-off between the current solutions to solve our problems. Then, we **implement the approach in a prototype**, using the framework that best suits our needs, in order to achieve the ultimate goal of improving the process of health research data sharing. Lastly, we provide an **evaluation** of the feasibility and benefits of the solution based on experts' opinions that provide constructive feedback about the approach and its respective implementation, showing how it could be further improved.

1.6 Document Structure

This chapter contains an introduction to the dissertation, describing its context, motivation, problem as well as its main objectives.

Chapter 2 (p. 5) (Background) describes the context where this dissertation is applicable, addressing the concepts that support this dissertation, namely the topics of Networks, Software Security, Blockchain, Health Data Sharing and Data Traceability as well the sub-topics inherent to them.

Chapter 3 (p. 23) (State of the Art) presents a literature review on multiple subjects related with the work that will be developed. Firstly, it describes the current solutions for solving problems related to data traceability, including solutions that target medical data sharing. Each solution description is preceded by an introduction, briefly explaining the context of the solution, its main objectives and problems targeted. After the description of the solution there is a summary, reviewing the solution, presenting its main advantages and disadvantages as well as a brief analysis of how much it fits our problem statement. At the end, this chapter establishes a comparison between the most important solutions (the ones closer to solving our problem), describing their advantages and disadvantages and the trade-offs between each of the solutions.

Chapter 4 (p. 61) (Problem statement) presents the issues with the current solutions described throughout Chapter 3 (p. 23), describes a solution proposal to solve them and to achieve the main objectives of this dissertation, under the assumptions which make the solution valid and relevant to the context where it fits. It also presents the hypothesis and research questions that support this dissertation.

Chapter 2

Background

This chapter contextualizes the most relevant aspects of the topic approached in this dissertation. We start by briefly explaining the wide topics like the architecture of the Internet (Section 2.1, p. 5) and Software Security and Privacy (Section 2.2, p. 6). Then we go into more specific topics including Blockchain (Section 2.3, p. 11) and BigData (Section 2.4, p. 17) and finally we delve into the topic of Health Data (Section 2.5, p. 18) that is the scope of this dissertation as well as the project on which this dissertation fits: iReceptorPlus (Section 2.6, p. 19).

2.1 The Internet

This section will briefly explain the evolution of the internet's architecture, describing concepts as Client-Server and Peer-to-Peer (P2P). This will contextualize the wide scope of this dissertation to help later introducing the appearance of Blockchain, in Section 2.3.

2.1.1 Client-Server

After its birth, the internet has quickly evolved towards a client-server architecture [9, 7]. This architecture is based on the initiative of clients to open a connection to the server which should be always online and listening for requests [94]. The server satisfies the requests of all clients and manages all the security and business logic. The architecture has broadly spread due to being the easier and most intuitive way of building distributed systems [92]. Although easy to implement and use by the average customer (due to the topology of the internet), this architecture has the drawback of centralizing the power of decision and the business logic implementation on the entity that owns the server [93] in the sense that all clients must trust the server. This puts users' security and privacy at risk [51] since there is no cryptographic protection against malicious modifications of the data stored in the server, allowing who is in control of the server, or an attacker that has managed to crack the server, to modify the data at will [23].

Figure 2.1 illustrates the Client-Server network architecture.

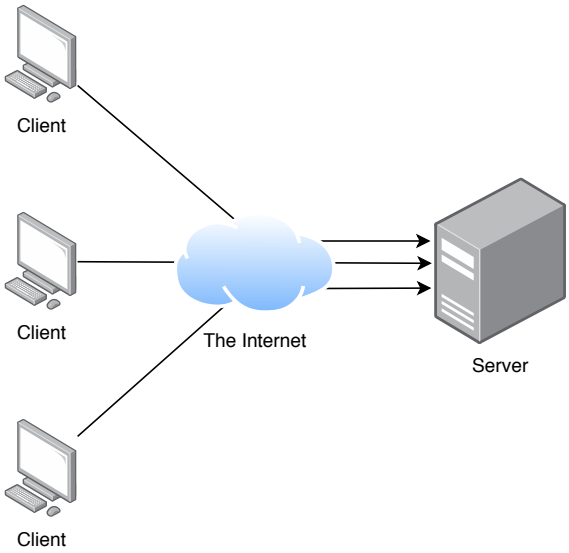

Figure 2.1: Client-Server network architecture

2.1.2 Peer-to-Peer (P2P)

These problems with the client-server architecture have driven subjects to seek for other alternatives such as Peer-to-Peer (P2P) [81]. In the beginning, this architecture has caught the terrible reputation of only being useful for piracy and other illegal actions due to its inherent decentralization [88]. However, that reputation was slowly lifted and P2P has quickly gained adoption in many legal applications such as multiplayer games [18] and content distribution networks [81].

The problem with P2P networks was need of trusting unknown peers. Trusting a server is not the best, since the entity operating it may manipulate data and business logic to its favor, but it is normally safer than trusting an entire network of people that we don't know. Normally, entities operating servers have a lot of incentives to be honest, so Client-Server architecture has prevailed for years without major incidents. In order to implement a P2P network, we need a way of being able to verify instead of trusting. That is the basic idea behind blockchain.

Figure 2.2 illustrates the P2P network architecture.

2.2 Software Security and Privacy

The security of a software requires attention to two major dimensions:

- Cryptography and information security
- Computer network and system security

The first one respects to the security of the information that doesn't necessarily need to be travelling across a network, whereas the second one can be seen as a composite of the first in the sense that it requires cryptography in order to be implemented and is more driven to networking [91]. Since this dissertation involves blockchain, which uses cryptography in order to provide

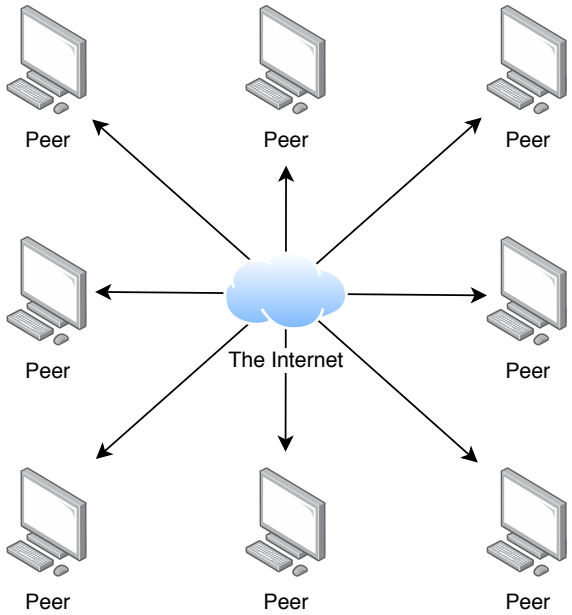

Figure 2.2: P2P network architecture

security and immutability of the transactions, we are going to briefly describe each so that we can understand the context behind the technology, in Sections 2.2.1 and 2.2.2.

Then, in Section 2.2.3, we will delve into the concept of **Hashing**, which is an important process for ensuring the integrity which is a crucial aspect of security. Finally, in Sections 2.2.4 and 2.2.5 we will present the two main types of cryptography, Symmetric and Asymmetric, which represent important processes to guarantee the confidentiality and integrity of information.

2.2.1 Information Security

Information security uses cryptography to guarantee the three most important aspects of security:

- **Confidentiality** ensures the non-disclosure of secret information so that only authorized entities can read it.
- **Integrity** ensures that the information has not been modified.
- **Authenticity** ensures the identity of an entity in some action taken by it. In our context, it allows to provide trust over who registered the information.

2.2.2 Computer Network and System Security

Computer network security uses cryptography and the specific business logic of the context to guarantee:

- **Authorization** which ensures that only who has permission over certain information can access it.

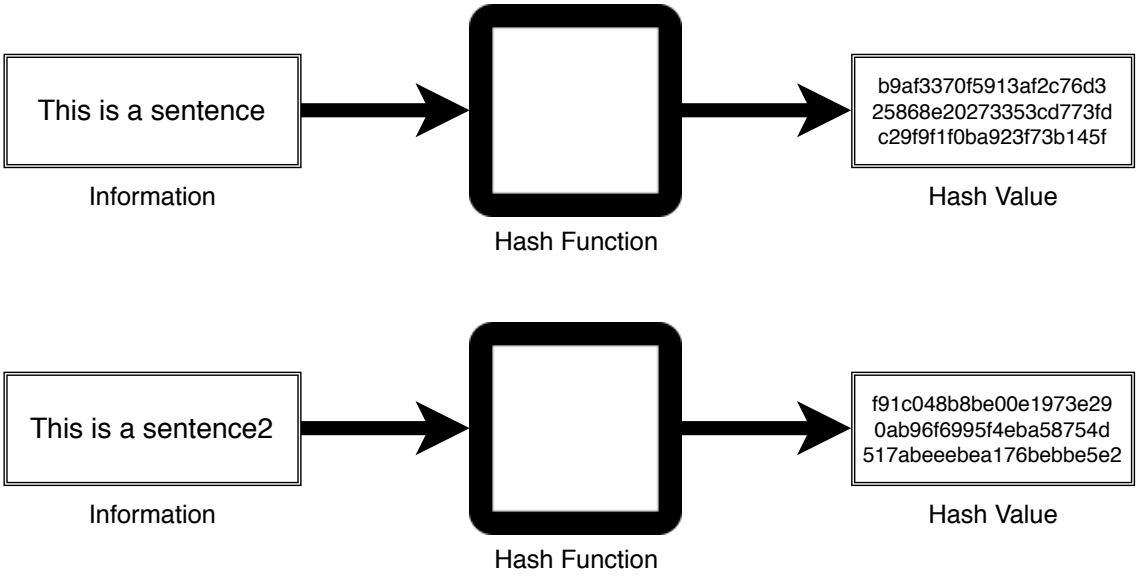

Figure 2.3: Cryptographic hash functions

- **Availability** which ensures that the services are available and operating normally to the users that have properly authenticated and are authorized access to that resource.

2.2.3 Hashing

Cryptographic hash functions are one-way functions that allow to create a value, called *hash*, which is a representation of some information [91]. The information (input to the function) can have any size and the *hash* (output of the function) will always have the same size.

The fact that they are one-way functions, allows to create an irreversible representation of the data which can be used to verify its integrity because it is computationally infeasible to change the message and produce the same hash: if we have confidence over the hash's integrity, then we can have confidence over the information's integrity [29].

Slightly changing the information, causes its *hash* value to change completely, making it impossible to be predicted and, therefore, manipulated, providing trust over the integrity of the information in the sense that it is computationally infeasible to find the original information that results in a specific *hash* value.

Figure 2.3 illustrates the operation principles of cryptographic hash functions.

2.2.4 Symmetric Cryptography

Symmetric Cryptography is a two-way function that, given a *key*, allows to transform some information (input to the function, also called *plaintext*) into another of the same length that is completely different (output of the function, also called *ciphertext*) and cannot be correlated with the initial *plaintext*. The effect of the function can be undone, allowing to get back to the *plaintext*

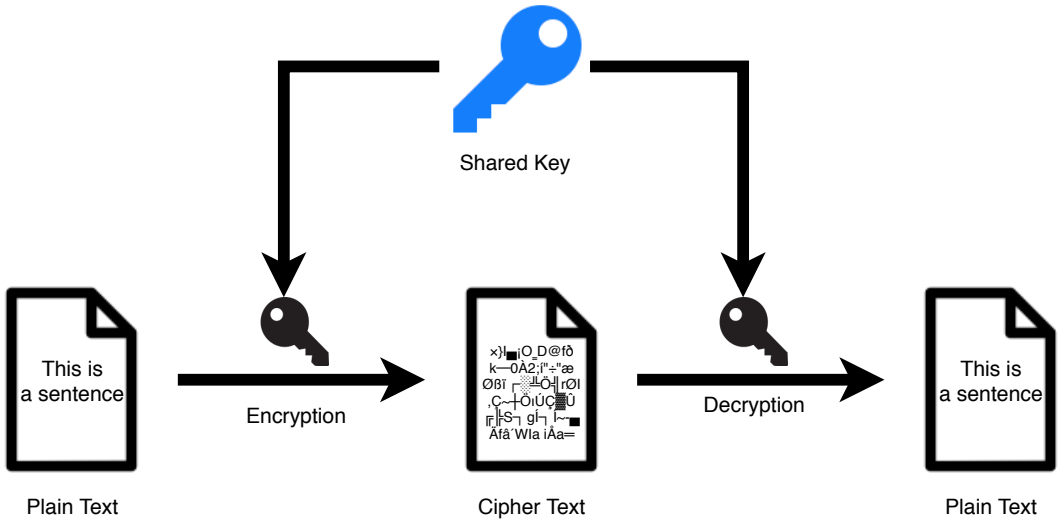

Figure 2.4: Symmetric cryptography

by supplying the same *key* and the *ciphertext*. It is called symmetric because the *key* is the same in both directions of the transformation [91].

The most used algorithm for symmetric cryptography is AES (Advanced Encryption Standard [28].

Figure 2.4 illustrates the principles explained above about symmetric cryptography.

This type of cryptography is most used for **confidentiality**, in which context it is called encryption. It allows two entities to communicate with confidence that anyone viewing the encrypted messages (ciphertext) will never be able to disclose the information as long as both have a shared secret that no one else has, called the *key*. Since the *key* needs to be shared prior to the secure communication, it is also called *shared key*. And because it needs to be kept secret in order to guarantee **confidentiality**. it is also called *secret key*. This type of cryptography can also be used to ensure **Integrity** of the information.

2.2.5 Asymmetric Cryptography

Asymmetric Cryptography is also a two-way function that allows to transform *plaintext* into *ciphertext* and vice-versa. The difference between symmetric and asymmetric cryptography is the fact that the *key* used to transform *plaintext* into *ciphertext* is different from the one used to transform *ciphertext* into *plaintext*. One *key* applies the inverse transformation of the other. (Applying the function to the *ciphertext* with the *key* used to generate that same *ciphertext* will result in a different *ciphertext*.) One of the keys is called the *public key* and the other is called the *private key* [91].

Figure 2.5 illustrates the principles explained above about asymmetric cryptography.

This type of cryptography is most used for both **confidentiality** and **integrity**.

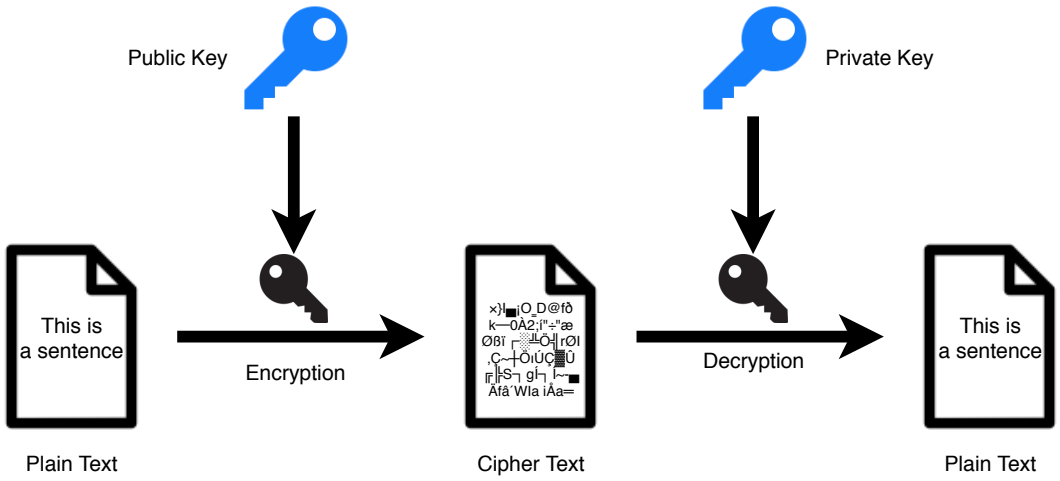

Figure 2.5: Asymmetric cryptography

Confidentiality When used for **Confidentiality** it is called encryption and, unlike symmetric encryption it not only allows two entities to communicate secrets but also that everyone communicates secrets with one entity, using the same *key*, the *public key*, to encrypt the data but only the entity is able to *decrypt* since it requires the *private key*.

Integrity When used for **Authenticity** it is called *digital signature*. It allows an entity to use its *private key* to digitally sign some information, allowing other entities to verify that *signature* with the *public key*. Everyone can verify the signature but only the entity who has the *private key* can generate it. Typically a *hash* (explained in Section 2.2.3, p. 8) of the information that we want to sign is generated and that same hash is signed, due to the computational effort necessary to cipher large amounts of data using asymmetric cryptography.

Some algorithms used for asymmetric cryptography are RSA [82], Elgamal [33] and ECC (Elliptic Curve Cryptography) [70, 56, 35].

2.2.6 Symmetric vs Asymmetric Cryptography

Symmetric cryptography is older and is typically preferable whenever possible mostly due to the fact that it is computationally easier to perform. Furthermore, microprocessor manufacturers are currently incorporating hardware acceleration mechanisms to symmetric encryption, which make it even faster to run on almost any device [104]. However, this type of encryption is rather limited since it requires that both entities share a secret prior to the communication, which, in most cases, is not possible. This type of encryption is also used for individual data confidentiality such as entire device encryption, since the owner creates the secret, no one else needs to know that secret and it is necessary to encrypt large chunks of data.

Asymmetric cryptography, on the other hand, is computationally heavier and is to be avoided when encrypting large chunks of data since it can make the process really slow. This type of encryption is used when entities need to have a common public secret, the *public key* and don't want to exchange secrets with each entity they want to be able to verify the digital signatures of.

2.3 Blockchain

Blockchain was born as a technology to support P2P electronic cash [42]. Firstly, we describe the concept and provide some background history about it, in Section 2.3.1. Then, we explain how blockchain was introduced, in Section 2.3.2 (p. 12). Furthermore, we present more details about the important aspects of the blockchain concept, in Sections 2.3.3 (p. 13) and 2.3.4 (p. 15). Finally, we summarize the aspects described about blockchain, in Section 2.3.5 (p. 17).

2.3.1 P2P Electronic cash

With the digitalization of the information came the digitalization of money. Following the current most widely used network topology of the internet, described in Section 2.1 (p. 5), it came in a client-server architecture. Solutions that provide direct money transactions between users (Consumer to Consumer - C2C) like Paypal [41] were born. The way they work is rather simple: the server decides who owns how much and dictates the approval of each transaction. All users of Paypal need to trust the server to have confidence that the number that it sends as "total balance" will be available for them to spend anytime. Companies using services like this need to trust the server when it says that a customer has enough money to pay. Although this normally works because the entities running the server have incentives to show honesty, it is not cryptographically supported and the systems are not fully transparent, which may lead entities to dishonest activity behind the scenes. Therefore, there was constant seek for better solutions.

The idea of a system where users didn't need to trust a server to know how much they own and verify if someone has enough money to pay existed since well before blockchain was introduced. This is known as P2P electronic cash. The problem is that if we can't trust a server that has a lot of incentives to be honest, we can't trust an entire network formed by users we don't even know. These systems require full *byzantine fault tolerance* [31], thus avoiding the need of trust, in order to work properly. Many solutions for electronic cash have been proposed even before 2000 [77, 20, 76]. The idea of *cash* in these systems is a constantly growing ledger with transactions. The whole history of transactions since the very beginning is the currency itself. These solutions use asymmetric cryptography to provide *authenticity* of the transactions, where users sign the outgoing transactions with their private key and the rest of the network can verify them with their public key. This solves the problem of broadcasting fraudulent transactions to the network, avoiding money stealing between users. The problem with this resides at precisely the opposite: What if the user does not broadcast the outgoing transactions to the network? This leads us to the double spending problem.

Double spending problem The double spending problem is the possibility of a user to deliberately not broadcast outgoing transactions to the network so that it doesn't know he spent that money, enabling him to spend it again or even multiple times.

There was no proper solution to this problem until 2008 when someone, or a group of people, under the name of Satoshi Nakamoto, released a white paper, proposing a P2P electronic cash system with name of Bitcoin [73].

2.3.2 Bitcoin

Bitcoin was the first digital currency to provide a proper solution to the double spending problem. The system can provide byzantine fault tolerance [31] as long as more than 50% of the resource used for consensus of the system is the hands of honest users. In order to provide this byzantine fault tolerance, Nakamoto has proposed an algorithm for the network to achieve consensus over which ledger is the legitimate one and which changes to accept. If the entire network can agree in which ledger to trust, there will be no double spending problem since honest users all agree in the same balance for every user in the network. The algorithm is called **Proof of work**. The ledger is called **blockchain** which a constantly growing list of blocks, which contain the *transactions* and are linked through **Proof of work**. The decision for the legitimate *chain* is according to the **longest chain rule**.

Proof of work The algorithm for achieving consensus is based on computational power. In order to create a new block in the chain, the users need to use computational power in order complete a cryptographic challenge. To incentivize users to participate in this validation mechanism, a reward is given to the user who completes the cryptographic challenge. This is also how new money enters the network.

Blockchain The blockchain is a constantly growing chain of blocks that are linked to each other cryptographically through **proof of work**. There is a cryptographic proof linking each block to the previous one, thus providing trust over the immutability of the ledger in the sense that if an attacker wanted to change a block (to maliciously corrupt the data), they would have to re-compute the **proof of work** of the whole chain.

Longest chain rule The concepts explained until now still don't 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 the valid chain. The algorithm that allows honest users to achieve consensus over which chain to trust is the **longest chain rule**. The rule says that the chain to trust is always the longest, which is the one that has 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 system *byzantine fault tolerant* as long as more than 50% of the computational power is in the hands of honest users.

Because the cryptographic challenge (**proof of work**) is hard to solve and it grants a reward, the process of solving it, and therefore the process of creating new blocks, is called *mining*.

2.3.3 Consensus Algorithms

After bitcoin, many other digital currencies, also called cryptocurrencies, have appeared [16, 84, 11]. The other solutions have brought new consensus algorithm for validating the blocks.

A blockchain consensus algorithm is the basis for the security of the blockchain, supporting the immutability and tamper resistance of the system. 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 resource to be used for the proof mechanism. This resource works as a proof that the majority of people is using the resource for a certain chain and, therefore, we should be able to assume that the majority of the network is looking at that specific chain as being the valid one. In that sense, we have consensus. In the case of Bitcoin 2.3.2 it is computational power.

Now we will explain briefly some of the most used blockchain consensus mechanisms.

Proof of work Proof of work is a consensus algorithm introduced by Nakamoto that uses computational power as the resource to provide immutability and tamper resistance to the blockchain [73]. This mechanism uses cryptographic hash functions, described in 2.2.3 to create a unique representation of each block, the *hash*. Every block is linked to the previous one through a cryptographic challenge that involves the use of the *hash* of the previous block. This way, the cryptographic challenge to solve always depends of the previous block and the content of that block are the transactions, which are created by the network. Therefore, the possibility of using precomputed (already solved) cryptographic challenges doesn't work because the network chooses which the cryptographic challenge will be the next one, naturally by submitting transactions.

Proof of stake Proof of stake is a consensus algorithm introduced by King and Nadal that uses the resource exchanged in the blockchain transactions to provide a proof of majority of the network [95]. In case the blockchain is being 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 [73, 86]. In this type of validation system, the blocks are linked through a hashing process that is done over a limited search space instead of an unlimited search space like in proof of work [95]. 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.

In Figure 2.6 we can have an overview of the current blockchain consensus algorithms.

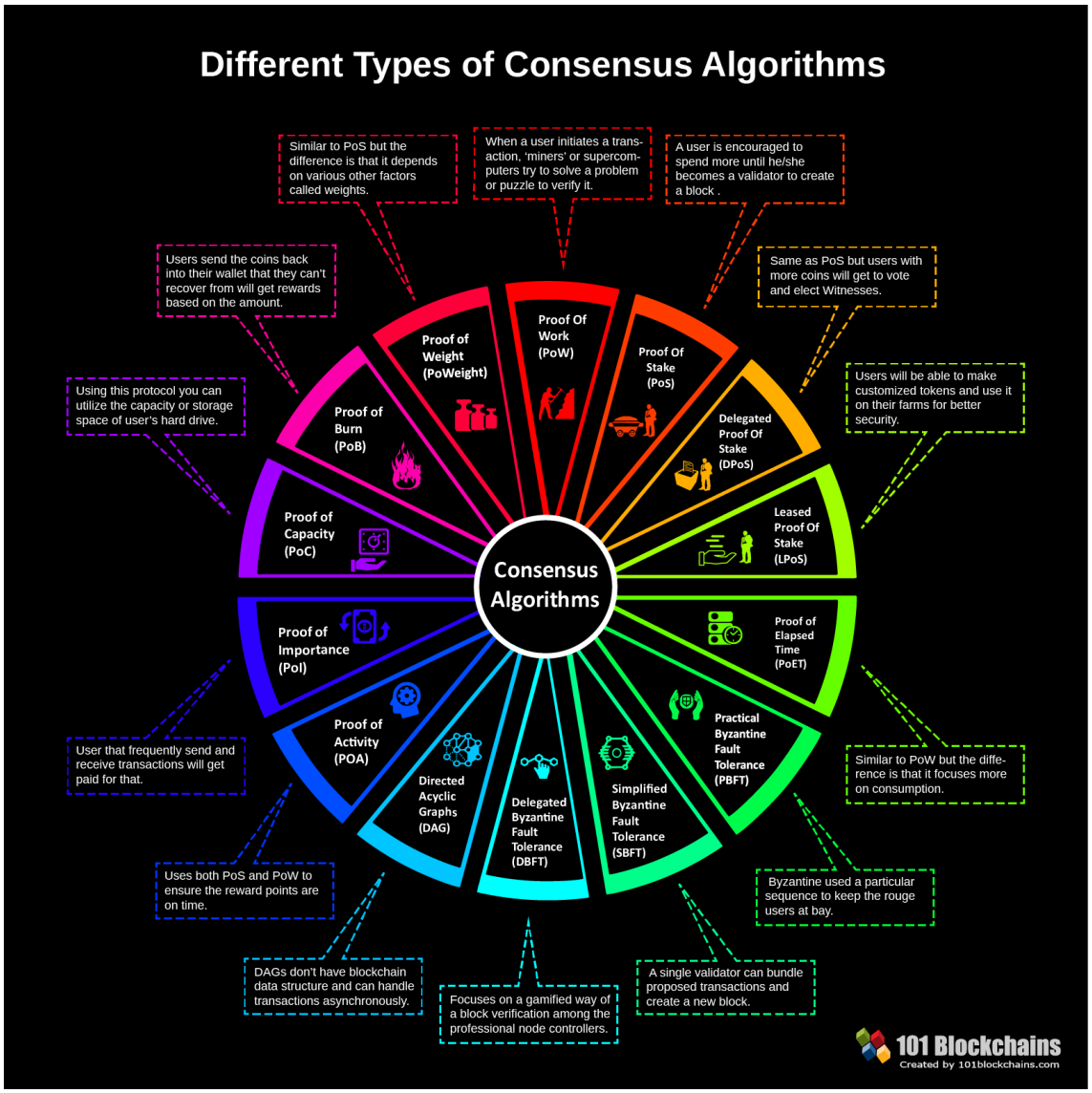

Figure 2.6: Blockchain consensus algorithms as presented by Anwar et al. [6]

2.3.4 The Security of Blockchain

Since blockchain is a P2P network, there are network layer possible attacks [46]. However, these are possible in every network and there is not much to do against them and they normally can cause only a Denial of Service (DoS) . Therefore, we will keep these out of our scope and focus on the security of the consensus algorithm instead.

The security of the blockchain is fully dependent on the consensus algorithm. It is based on the algorithm that the network agrees in who owns what and, therefore, validates if the transactions are valid. That is done without trusting a single central entity, thus providing decentralization. If the consensus algorithm is broken, it is impossible to be confident of the validity of transactions, therefore making the entire information on the blockchain not trustable [13].

Since there is no need of trust in a single authority, every user in the network deserves the same trust. This avoids centralization, but this confidence can quickly be vanished if an entity is able to cheat the system. The consensus algorithms provide byzantine fault tolerance [31] as long as more than 50% of the resource used for consensus is in the hands of honest nodes. The possible attack is very straightforward: an entity is able to control more than 50% percent of the consensus resource. The means and hardness to reproduce such an attack very depending on the consensus resource being used. The following paragraphs will explain the possible attacks to each of the major blockchain consensus algorithms as well what makes them secure against those attacks.

2.3.4.1 Proof of work

This mechanism uses computational power as the consensus resource to secure the network. The computational power is demonstrated through a process called *mining* which basically consists of running a hash algorithm (explained in Section 2.2.3, p. 8) many times. The nodes that are *mining* are often called *miners*. Since the *miners* are essentially performing *hashing*, the total computational power that is being used to secure the network is commonly called *network hashrate*. The difficulty of the challenge to solve changes based on the *network hashrate* through a process called *difficulty adjustment*, to ensure that the reward created remains the same regardless of the *network hashrate* [5]. A possible attack requires the attacker to have more than 50% of the computational power and, therefore, is commonly called a *51% attack* [63].

These attacks may have different intentions. There are possible DoS to the network using the consensus algorithm [26] but we will not go into much details 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 contents of the blockchain and re-compute the cryptographic challenges that link the blocks together in order to convince the network that the new content is the correct one. This type of attack is hard to reproduce since re-computing the cryptographic challenges from the block that the attacker intends to change be 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 [83]. The steps to reproduce the attack normally involve:

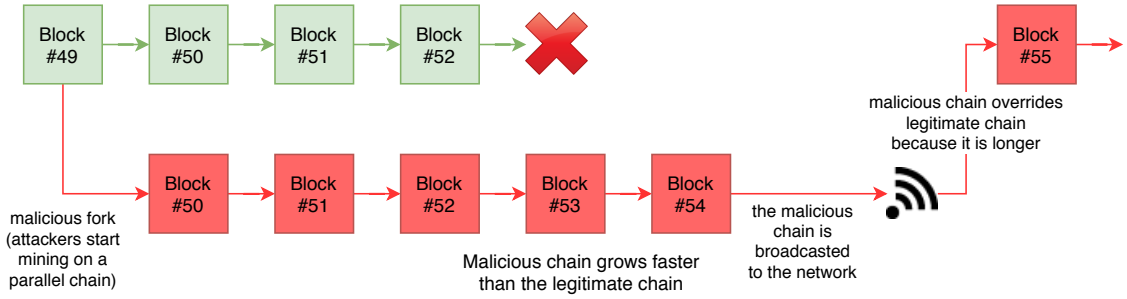

Figure 2.7: Blockchain 51% attack

1. The attacker starts mining on a parallel (malicious) chain without broadcasting the blocks to the network.
2. Eventually, the malicious chain will grow bigger than the legitimate chain because it the attacker has more than 50% of the hashrate.
3. The attacker then broadcasts the malicious chain to the network.
4. The network follows the longest chain rule and, therefore, trusts the malicious chain.
5. The malicious chain overrides the legitimate chain which is thrown away.
6. The malicious chain is now the main chain.

The Figure 2.7 illustrates how this chain is formed. This type of attack is commonly attempted in cryptocurrency blockchain projects with the intention perform double spending of money [53].

Therefore, what makes the networks secure is having the majority of the available computational power allocated to secure the network. It is a common mistake to relate this to the *network hashrate* [113]. This measure tends to increase with time due to the evolution of computer hardware, although the security of the network remains the same. Therefore, it is preferable to measure the security of the network as the electricity being used for the *mining* process [107]. This is dependent on the profitability of the *mining* process which is dependent on the value that the reward has. The value of the reward is dependent on the adoption of the blockchain network and value that people give to it. In the case of cryptocurrencies, this can be measured using the market capitalization of the currency [49].

2.3.4.2 Proof of stake

The proof of stake mechanism uses the resource exchanged in the transactions to secure the blockchain. Therefore, the attack also requires that the attacker gathers more than 50% of the resource. This is supposed to be more difficult to achieve than with the proof of work system due to the fact that the attacker would need to buy a huge amount of the resource that is exchanged in the transactions. This would make its value increase and make it even more difficult to buy such a

huge amount due to the supply and demand rule [32]. However, attacks are still possible on proof of stake consensus algorithms [36].

Following the same logic as proof of work, the networks are more secure as the amount of stake of validators increases. This comes from the incentive which comes from the value of the asset.

2.3.4.3 Summary

In conclusion, what makes a blockchain network secure is the value it has for people, regardless of the consensus algorithm. That value, increases the value of the validation process reward, therefore attracting more consensus resource to be used to secure the network.

2.3.5 Summary

Blockchain can be seen as a distributed and decentralized database that is immutable and tamper resistant. The technology works using a P2P network architecture, which does not depend on a central server, thus providing high service availability. Blockchain allows the network to reach distributed consensus, even between distrustful entities, and create an unique source of truth without depending on a single authority.

The blockchain is implemented as a list of blocks that are linked together through a cryptographic challenge mechanism that requires effort to be completed. Each block is made of transactions. The transactions represent transfer of ownership of assets between entities that are registered on the blockchain. These transactions are the content of the blocks and are chosen by the network. The cryptographic challenge that links the blocks together securely is given based on the contents of the blocks, which means that the attacker has no control over it, preventing replay attacks. The system can provide confidence over the data that is stored by putting the same trust on every node of the network.

The cryptographic challenge depends on the consensus algorithm that is used to secure the network. Many consensus algorithms have been proposed after the first *proof of work* algorithm introduced by Satoshi Nakamoto in 2008. The security of the blockchain is fully dependent on the consensus algorithm but, regardless of the algorithm, it always increases with the adoption of the asset that the network aims to support. This adoption requires trust in the blockchain security which only comes from adoption itself [50]. This, together with the need of computer knowledge to understand the system is holding back the blockchain adoption, causing a dead lock situation where people are waiting for trust that requires them to come.

2.4 BigData

The concept of Big Data is broadly spreading across every information system. It refers to large subjects' data sets with a complex structure which are challenging to store and analyze with the objective of finding secret patterns and correlations [87]. Companies are exploring how they can

take advantages of the discovery of these secret patterns to create and capture value for individuals, businesses, communities and governments [66].

Subject's data is becoming more and more valuable due to the advantages that can be taken from its analysis [74]. Regardless of the type being either emails, videos, online transactions, search strings, social networking, scientific data, mobile phone sensor data or health records, there is always an advantage to be taken from the analysis of it [21, 111]. The process of discovering interesting and valuable structures in large data sets is called data mining [44]. It is through this process and all its sub-processes that companies can take advantage over the competition by knowing and predicting in advance.

This attention from major companies has also drawn the attention of regulators into the field [30, 60]. In the last few years, data regulations have become increasingly severe in protecting people's personal data [57, 103]. This has raised awareness of subjects to the importance and value of their personal data, encouraging them to protect their data and take advantage of its value [90].

This together with the appearance of solutions to provide decentralized data sharing without the need of trust in a central authority [73], is raising awareness of subjects to the importance and value of their personal data [90, 114].

2.5 Health Data

The concept of Health Data, as a subset of BigData (described in Section 2.4, p. 17), is increasing in popularity. The need for sharing health data among multiple parties has become evident in several applications [78]. The use of subjects' data to make important decisions and improve people's lives is growing interest in the community [71, 110]. Research on large health data sets provide major opportunities for improving health systems and individual care [59].

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 the data an almost instant process. In order to take full advantage of the value resident in the subjects' medical data, there is constant need for the right mechanisms and technologies for ensuring the availability of this data. Currently, medical records are spread over multiple repositories, making the access to this data very challenging, compromising the individual health care as well as health research [2].

This sharing process, although essential for public health, patient care and clinical research, is raising privacy concerns due to the fact that health data is highly sensitive from the social and economic points of view [37]. These concerns are raising attention from the regulators of the health field [89]. The attention from research projects and regulators has raised subjects' awareness to the value of their health data and the importance in protecting it more than with any kind of personal data [78, 90].

This tightening in regulation together with the awareness of subjects to protect their health data, has lead to an increased demand for mechanisms to grant subjects' privacy when sharing health data [71, 52]. Similarly to BigData, blockchain is being an increasingly explored possibility

for supporting the sharing of health data [64]. Due to the high sensitivity of the information and the conflicting interests of the entities involved, a technology that provides decentralized trust over the integrity and veracity of information, while maintaining its privacy, is necessary. This leads to the exploration of blockchain as a mean to support health data sharing platforms [68].

2.6 The iReceptorPlus project

The iReceptorPlus is a research project that aims to support health data sharing by providing access control and traceability to the process of sharing health data [1]. The project is a Canadian and European partnership in which the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC) is part of.

The project aims to promote human immunological data storage, integration and controlled sharing for a wide range of clinical and scientific purposes by developing a platform to integrate distributed repositories [25]. It leverages the Adaptive Immune Receptor Repertoire sequencing (AIRR-seq) standard [105] to improve medicine and immunotherapy in cancer, inflammatory and autoimmune diseases, allergies and infectious diseases. The standard has become critical to several disease treatment research applications, as it supports the ability to replicate procedures by standardizing the representation of data, enabling cooperation between multiple research projects [24].

In the context of the project, three entities are considered:

- Patients
- Health Companies
- Research Projects

Each of these entities has different interests, which sometimes can be conflicting. Some of the interests of each entity are described below:

Patients want to

- Be provided individual health care by **Health Companies**.
- Have their health data as secure and private as possible.
- Have total *awareness* over what happens to their data.

Health Companies want to

- Provide individual health care to **Patients**.
- Be compliant to the data protection regulations.

[1] More information about iReceptorPlus can be found in the official website https://www.ireceptor-plus.com/about-us/overview/

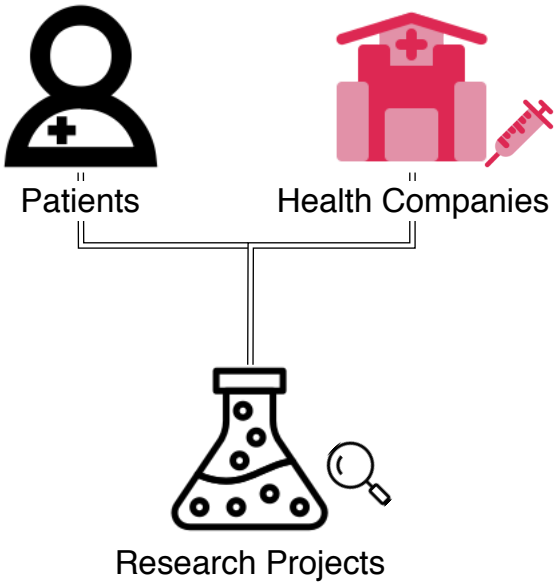

Figure 2.8: iReceptorPlus entities involved

Research Projects want to

- Have access to a lot of patients' health data (the more the better) so as to achieve more breakthroughs in disease treatment research.
- Get **Patients** more confident to provide them access to highest possible amount of data.

Figure 2.8 provides an overview of the entities considered in context of the project.

Some of the interests are conflicting. For instance, patients want to have their data as private as possible, while research projects want to have access to as much patients' data as possible, having no interest in protecting patients' data. They intend to achieve as many progress as possible, regardless of the impact in the patients' security and privacy. There are some data protection regulations [103] that the platforms need to be complainant with in order to ensure their operation.

The project aims to give patients the ability to have full traceability of what happens to their health data from the moment it is registered on the system as well as a mechanism to verify the validity of that traceability information. Traceabilty provides the ability to perform data lineage, tracing all the transformations that ocurr to the data as well as determining the provenance of the data, when it was first registered on the blockchain. The system should be able to provide trust over the provenance of the data, providing the ability to identify from which patient is the data that helped a certain research project, thus allowing the patients to be rewarded for their contributions.

The use of blockchain to provide this confidence with as least trust and as most decentralization as possible. These two aspects are key for the competitive context, that demands a decentralized database like blockchain which eliminates the need of trust in a central authority which could have interest in tampering the data. Lastly, we think that the solution also makes patients more confident to provide research projects with their health data, thus allowing them to make more progress.

2.7 Summary

In this chapter, we explained how the architecture of the internet started as client-server and why the alternative P2P was left behind in the beginning (Section 2.1, p. 5). We also covered the basic concepts behind the Software Security and Privacy (Section 2.2, p. 6).

We explained the trust problems with P2P networks and how client-server architectures proved to be the best solution in the lack of mechanisms for providing confidence in the security of P2P networks. With the appearance of technologies like Blockchain (explained in Section 2.3, p. 11) that take advantage of cryptography to provide confidence over the information in a P2P network without the need of trust, the P2P architecture has been conquering more adoption.

We described the concept of Big Data (Section 2.4, p. 17), covering the advantages of analyzing subjects' data in order to capture value for individuals and businesses. Then, we delved into the concept of Health Data (Section 2.5, p. 18), as a subset of Big Data that consists of exploring the DNA secrets of subjects in order to make important breakthroughs in disease treatment.

Finally, we closed out describing the more specific context where this dissertation fits: The iReceptorPlus project (Section 2.6, p. 19). This projects aims to support health data sharing by providing access control and traceability to the process. We explained the problems of having multiple entities with different interests involved and how it may lead to data manipulation, compromising subjects' security and privacy.

Taking advantage of the Blockchain technology that is able to provide confidence in a P2P network, we can avoid trusting a single entity to provide confidence over the integrity and veracity of the data. In the field of Big Data, data manipulation is very likely to occur, as entities have their own objectives. Furthermore, in a project that deals with Health data, with multiple entities with different interests involved and with patients' security and privacy at stake, it is crucial to avoid the need of trust in a single authority to provide confidence over the integrity of the information.

Chapter 3

State of the Art

Numerous research works are being conducted around blockchain-related topics [98]. This work addresses traceability in the process of sharing personal health data in particular. A research on the state of the art for the context of our problem was performed. The research was limited to blockchain solutions only, since other solutions use a centralized database, which is controlled by a single entity. This type of storage is not suitable for our context (described in Chapter 2, p. 5) due to the fact that there are multiple entities with different interests involved and, therefore, there is the need to avoid trusting a single entity.

In Sections 3.1 (p. 23), 3.2 (p. 29), 3.3 (p. 41) and 3.4 (p. 46), the current solutions for data traceability are described in detail.

Then, in Section 3.5 (p. 48), we present a comparison between these solutions, pointing out the key advantages and disadvantages of their mechanisms, **establishing the best trade-off** between them. We also point out the key problems with each of the mechanisms alone.

Finally, in Section 3.6 (p. 54), we analyse the current frameworks available to develop blockchain applications, in order to **select** which one suits more the objective we want to achieve.

3.1 MedicalChain Solution

MedicalChain is a cryptocurrency project that uses blockchain technology to create a platform to support secure electronic health record (EHR) sharing. It aims to support patients in the process of giving health care professionals access to their health data [2].

The platform provides patients some awareness over the interactions made with their health data and control over which health care professionals have access to it. The interactions with the data are recorded in an auditable, transparent and secure way on the MedicalChain's distributed ledger.

3.1.1 Blockchain

MedicalChain is built using a dual blockchain structure. The first implements access control to the electronic health records and is built using the Hyperledger Fabric [4]. The second is powered by

an ERC20 token and backs the applications and services of the platform.

Hyperledger fabric is a framework for developing blockchain projects and allows permission-based access to resources using its access control languages. It is meant to have a modular architecture, thus providing flexibility and scalability [19]. Hyperledger Fabric allows the owner of a data set to control which parts of the data is accessed, making very suitable for managing access to electronic health records since it provides awareness to the patients, thus increasing their confidence in providing health care professionals access to their personal health data [2, 4].

Ethereum is a global, open-source platform for developing decentralized applications [17]. While also being a cryptocurrency, Ethereum was the first to introduce the ability of users to write smart contracts [16]. Smart contracts are code that is executed on the Ethereum blockchain, they can been seen as a more flexible, more generic version of transactions [16, 58]. This allows to program generic contracts between parties and generic asset types on the blockchain, rather than only allowing money transferring.

3.1.2 User types

MedicalChain defines user types (actors) that are allowed different actions based on their role. Each user is associated with an asymmetric key cryptography pair (private + public key) (described in Section 2.2.5, p. 9) as well as with its type. This allows to determine the actions that each user is allowed to perform and, therefore, validate if it is permitted.

The actions allowed for each actor are shown below.

Patient

- Read their EHR
- Grant a Practitioner/Institution to Read/Write their entire EHR or a portion of it
- Revoke a Practitioner/Institution's Read/Write access to their EHR
- Permission next kin/emergency contact to ro Read/Grant permission
- Write high-level health attributes to EHR, such as: Amount of tobacco consumed daily, alcohol consumed daily, weekly exercise.

Practitioner

- Read/Write EHRs he has permission over
- Request EHR Read/Write access permission for other Practitioner/Institution

Research Institution

- Read EHRs it has access to.

3.1.3 Encryption Mechanism

Electronic health records are encrypted using symmetric key cryptography (described in Section 2.2.4, p. 8) (for example, AES [28]). The record is stored (encrypted) in a repository, under the conditions of the appropriate regulation. It is not specified who generates the key and encrypts the record. To ensure maximum privacy, it should a device trusted by the patient. If the key used on the symmetric encryption process is known, it is trivial to decrypt the record, allowing anyone storing it (encrypted) to disclose all of its contents. Because of this, the key used in this symmetric key cryptography is not stored in the blockchain in plain-text, but instead it is stored encrypted. The next paragraphs explain how this encryption mechanism takes place.

The key used in the symmetric encryption of the health record is stored (encrypted) in the blockchain each time an entity is given permission to access a health record. The original medical chain white paper states that the first step of this process is to decrypt the record with the owner's private key. However, in order to use a private key, asymmetric encryption must be used. Since it is stated that the record is encrypted using symmetric encryption and never asymmetric encryption, we suppose that the author meant that the symmetric encryption key is decrypted with the owner's private key. The following paragraphs explain how this process takes place, although it is not strictly specified as such in the Medical Chain white paper. We take into consideration the state of the art of encryption in general (described in Section 2.2, p. 6) as well as the information on the paper.

Asymmetric encryption has two keys: public and private. Each one of the keys undoes the effect of the other: what is encrypted with the private key can only be decrypted with the public key and what is encrypted with the public key can only be decrypted with the private key.

The information regarding access authorization (to the health records) is stored on the blockchain. This information consists of the key used to encrypt the records (using symmetric encryption), encrypted with the public key of the entity that has access to that same health record. Only who has the private key (the entity which was given access to the record) corresponding to that public key can decrypt that information. This is how medical chain allows entities to access health records but at the same time ensures that, although everyone can view that information (the ledger is public), only the entity with access permission can decrypt and use it to access the record.

Although it is not specified in the medical chain white paper, in order to allow the storage of the symmetric key used for the encryption of the health record (ensuring the key and, therefore the access to the record, is not lost), the first access authorization being registered on the blockchain should be for the Patient. The symmetric encryption key should be stored on the blockchain, encrypted with the record owner's public key, ensuring only the record owner can access the key and, therefore, the record information. This also ensures that the key is not lost, although no one other than the owner is given access to the data because it is registered on the blockchain.

3.1.4 Encryption Mechanism Analysis

This mechanism has the security advantage that it does not require trust in the repositories that store the data, because they store it encrypted and they don't need to know the symmetric key to decrypt the data they are storing. However, the author does not strictly state who has access to the key neither that the repositories don't have access to it.

3.1.5 Record Access Process

Another advantage is that the procedures necessary to access and to grant access to a health record are very simple and easy to compute. The procedure of accessing a health record involves:

1. The information that represents the symmetric key encrypted and corresponds to the entity that is accessing the record is retrieved from the blockchain.
2. The private key of the entity that has access to the record is used to decrypt the symmetric key stored on the blockchain.
3. The symmetric key is used to decrypt the health record.

The procedure of granting access to a health record involves:

1. The symmetric key (the one that was used to encrypt the record using symmetric encryption) is decrypted with the owner's private key (asymmetric encryption).
2. The symmetric key is encrypted with the public key of the entity that is being given access to the record.
3. The result is stored on the blockchain.

However, the mechanism also has its disadvantage which is demonstrated in the process of revoking an entity's access authorization to a health record. Since all users that have access to a record know the symmetric key used to encrypt it, they can all decrypt it. Thus, the only secure way of revoking access is to re-encrypt the whole health record with another key which is computationally demanding. Thereby, the process of revoking access authorization to a health record involves:

1. The (first registered) information that represents the symmetric key encrypted is retrieved from the blockchain.
2. The private key of the owner is used to decrypt the symmetric key.
3. The record is decrypted using the symmetric key.
4. A new symmetric key is generated and the record is encrypted with that key.
5. The public key of the owner is used to encrypt the symmetric key and the result is stored on the blockchain.

6. In order to ensure that all the entities that are supposed to keep having access to the record, remain with that same access, the symmetric key must be encrypted with each and every single one of the entities public keys and the result stored on the blockchain. This step does not require the presence of any of the entities that have access because their private keys are not necessary in the process.

3.1.6 Record Access Process Analysis

Besides the drawbacks in the complexity of revoking access to a record, there is also no reference to how it preserves the integrity of information, ensuring that only the Patient can grant and revoke access to a record. Since all the users that have access to the record can retrieve it and decrypt it, they could also encrypt it with a new key and attempt to revoke access to all entities including the Patient. A modification to the consensus protocol would be necessary in order to guarantee that nodes would only accept access authorization granting or revoking transactions that are signed with the private key of the owner of the data. This would increase the complexity of the block validation, requiring all full nodes to check the first transaction (when the data is first registered on the blockchain) in order to get the private key corresponding to the owner of the data.

3.1.7 Blockchain Transactions

All interactions with the health records are supposed to be stored in the blockchain in the form of transactions. However, in some cases, it is not specified how this is guaranteed and that interactions with records never end up happening off the chain with no trace of it on the blockchain as we explain in the transaction examples below.

The transactions are private, meaning that only the entities associated with the transactions can view them. This is implemented using the features provided by the Hyperledger Fabric framework [19].

The medical chain white paper lists three transactions:

- Granting Access Authorization
- Revoking Access Authorization
- Practitioner referring patient

3.1.8 Blockchain Transactions Analysis

The first two are self explanatory and they trigger the processes already described in the Encryption Properties section, retrieving and registering the required information from and to the blockchain. The last one takes place when a practitioner grants another practitioner access authorization to a patient's health record.

There are some missing details regarding these transactions. The following paragraphs explain them in detail.

Missing record access transaction As mentioned above, it is specified that all interactions with health records are stored on the blockchain. However, there is no specification of the transaction associated with an access to a health record. This is clearly not as trivial as it might seem since there must be mechanism to guarantee that an access does not take place without being registered on the blockchain.

The proof that a user has access to a resource resides in the blockchain transaction that contains the symmetric key of the health record encrypted with the user's public key. The mechanism of retrieving this information from the blockchain, decrypting it with the correspondent private key, requesting the record from the repository and decrypting it with the symmetric key does not require any registration of any transaction. For a transaction to be registered upon a record access, it is necessary that the user accessing the record deliberately registers that same access. The user may not want to register this access and we shouldn't need to trust on good will.

A possible approach is to take advantage of the fact that the user accessing the record is not the only party involved in the process. The process requires the repository, that needs to give the user the data that represents the record (encrypted). However, there is no way of completely solving the problem. Requiring both parties to sign the transaction doesn't work since the user could refuse to sign it and without he's signature, the transaction would be invalid. Trusting only on the user is not an option, for the reasons already mentioned above. Trusting only the repository is also not an option since the repositories could be registering access transactions from arbitrary users to arbitrary resources. Trusting either one of the parties is the best approach we can see, however it could still cause conflicting information on the blockchain. Using a transaction that should be sign by both but could be registered on the blockchain signed by only either would cause problems in the case that it is not signed by both. It would be necessary to adapt the consensus protocol in order to deal with these kind of situations and decide what the full nodes should do when validating the blockchain when there are such inconsistencies.

Missing security guarantee on 3.1.7 transaction On the 3.1.7 transaction it is possible for a practitioner to give another practitioner access to a health record. This without any further security information (which is not mentioned on the white paper) clearly has its security problems since it suggests that any practitioner with access authorization to a health record can give permission to another practitioner to access the same health record without any consent from the patient that owns that data. In order to solve this, a system where the access permission is distinguished from the access giving permission would be necessary. There no such definition of the difference between these two types of permission on the white paper neither a solution to the problem.

3.1.9 Summary

MedicalChain is a platform more optimized to serve the patients and attempt to provide them with a solution to securely store their health. The solution should provide them control over who can access that same data and traceability over what happens to it.

Using a symmetric followed by asymmetric encryption mechanism for the health records and the access control based on the Hyper Ledger Fabric, the project aims to ensure that the health records cannot be accessed without the patient's permission. The intent to store every interaction with the health records on the blockchain provides traceability but it also has its limitations since it is not clear how some interactions are ensured to be stored on the blockchain in the form of transactions, compromising the traceability that the platform aims to provide and, therefore, the trust that the patients have on it.

The platform also aims to help health research projects, providing support for research institutions to be registered on the platform and be provided access permission to the health records by the patients. However, the support is limited to that and offers no further traceability over what the research projects do with that same data, how it gets processed and what contributions has it helped make to cure a certain type of disease.

3.2 DataProv Solution

DataProv uses blockchain to create a platform for facilitating the collection of data provenance information and also its verification and management in a distributed and decentralized way. The system uses smart contracts in order to enforce data integrity constraints.

The platform supports the manual introduction of provenance data through a user interface. The veracity of the provenance information is validated by the users, through a voting system.

Ramachandran and Kantarcioglu provide some illustrative examples of how the system can be used to validate the provenance data and how integrity is encouraged through a system that guarantees to be effective as long as more than 50% of the of the participants are honest. The validity of the information is determined by the various nodes of the network that want to participate on it and not by a central authority [80].

3.2.1 Background

The solution targets the context of scientific research which is very close to the context of the iReceptorPlus project. They reference the factors of ensuring the quality of the information as well as preventing data manipulation as very important to provide trust over research results and highlight the importance of provenance data in scientific research [47]. To support the statements, the author mentions an example of an audit conducted by the Cancer and Leukemia Grouahere 0.25 of the trials conducted were fraudulent. [39]

It is stated that the data provenance should be defined as meta-data that describes where the raw data originated, who owns it and what were the transformations done to the data.

This provenance data is expected to support the objectives of the scientific research and provide transparency over the procedures of the research so that people who were not involved in it don't need to trust on the information provided by the researchers, but can verify it instead.

It is highlighted that the integrity of the provenance data is a very important concern. The provenance data should be verifiable without the need of trust on a central authority. This requirement is filled by the Blockchain technology since it is an inherent characteristic of it. The technology also provides fault tolerance and high availability of the information since it is stored by all the full nodes of the network.

There is a reference to the specific type of data that we are exploring: medical records. Since this this type of data is highly sensitive, they state that they aim to provide access control to ensure no unauthorized access occurs. The solution also aims to be able to provide provenance information without compromising the privacy of the data in order to be suitable for contexts like ours where private data sharing mechanisms are required. In order to achieve this, asymmetric encryption is used together with access control policies, implemented using smart contracts.

3.2.1.1 Provenance Data Model Definition

The platform defines the provenance data model using the Open Provenance Model (OPM). Each entry of provenance data is called an action. Each action has three parameters:

- Artifact: the data of which to save the provenance record. This parameter involves the artifact before and after the change.
- Agent: the initiator which is the user who is inserting the provenance data entry.
- Process: the transformation made to the artifact which caused it to go from the previous state to the new state.

3.2.1.2 Threat Model

This section birefly explains the possible attempts to attack the system.

The object of attack is an artifact since neither the agent nor the process can be attacked. The DataProv paper defines two types of attackers: internal and external. These two types are defined according to the relation between the attacker and the specific artifact under attack.

An external attacker only has access to the blockchain and is part of the users that don't have access to the artifact under attack (he are external to the artifact). Not having access to an artifact means not having the key to decrypt it neither the location where the it is stored. The fact that the attacker does not have the key should make it cryptographically impossible for the attacker to read or change the artifact.

An internal attacker is part of the users that have access to the artifact under attack (he is internal to the artifact). Since the attacker has access granted by the owner of the artifact, he can submit provenance data regarding that same artifact. Since only the owner of an artifact can grant access permission to other users and as long as the owner of an artifact never wants to attack it, we can assume that an attacker can never grant access permission to an artifact to other users. This means that the attacker cannot forge users (controlled by himself) with permission to vote for an artifact with the intent to cheat the voting system to his favor. Therefore, and as long as more than

half of the users that were granted permission to access a document are honest, the provenance information that has been already approved should be able to be considered truthful.

3.2.2 System Overview

In the following sections, we delve into some of the most important components, providing an overview of the DataProv system.

3.2.2.1 Blockchain

DataProv is built on top of the Ethereum blockchain [16] and uses smart contracts [58] to implement the business logic in a distributed and decentralized way since these contracts are code run by all nodes of the Ethereum network.

3.2.2.2 Artifact Storage

The authors assume a scenario where the artifacts are stored in the cloud encrypted by their owner. It is not specified which type of encryption is used on this procedure (symmetric or asymmetric) (explained in Section 2.2, 6). It stated that access to the document is controlled using public key encryption. Taking into account the current state of the art of encryption where symmetric encryption (for example AES [28]) is used to encrypt large files (as explained in Section 2.2, p. 6), we assume that the mechanism is similar to the one in the MedicalChain platform [2] where the records where encrypted using symmetric encryption and the key was encrypted using asymmetric encryption. Similarly, we assume that DataProv uses symmetric encryption to encrypt the artifact (since it can be a large file) and then the key is encrypted using asymmetric encryption.

3.2.2.3 Provenance Data Consensus

The system uses the versioning approach for editing the artifacts. Each modification made to the artifact is stored as a version. The modifications are always made relative to the latest version of the artifact. The system discards changes that are not linked to a previous version (except for the creation of the file), always ensuring traceability is maintained.

When a user makes a change to a file, a new voting round is initiated. The voting round aims to provide trust over the integrity of the provenance information, hopefully causing wrong provenance data to be rejected. The system also attempts to prevent spamming of wrong provenance data (causing a lot of pending vote rounds and possibly overloading the system) by penalizing users whose provenance details get rejected and rewarding the voters who find wrong provenance details.

3.2.2.4 Provenance Data Input Flow

The flow of introduction of provenance data is executed whenever a user changes an artifact and registers the modifications in the system. It involves the following steps:

1. A user that wants to introduce provenance data in the system modifies the file and uploads the new version to the cloud server. (The old version is kept in the cloud since a rollback may be necessary.)

2. The user opens a voting round for the provenance data to be validated. That is done by submitting a request to Vote Contract through the user interface. This consists of a hash of the following:

   - Id of the artifact
   - Hash of the previous version of the artifact (encrypted)
   - Hash of the new version of the artifact (encrypted)
   - Pointer to the location of the artifact in the cloud storage
   - Timestamp of the change made to the artifact

3. The users with permission to access the artifact vote for the validity. In order to check the validity they use a script residing in cloud storage.

4. At the end of the vote round, the system decides if the change is truthful or not. This decision is based on whether the majority of users voted for or against the change, respectively. If the change is rejected, this causes the initiator to be monetarily penalized, loosing the amount he placed as deposit for the round. This amount is distributed among the voters, as a reward for denouncing fake modifications, incentivizing voters denounce such malicious introductions of fake provenance data.

5. If the vote round is approved, the change is recorded in the artifact tracker contract. The content of each change registered consists of the following:

   - The user responsible for the change
   - The hash of the new version of the artifact
   - OPM representation of the new change of the artifact
   - The digital signature for this information

The flow of input of provenance data is shown in Figure 3.1.

3.2.3 Implementation Details

The DataProv paper describes the architecture of the platform as having two high-level modules: on-chain and off-chain. The on-chain module runs on the Ethereum blockchain and mainly consists of smart contracts to implement the business logic: access control, storage of provenance data and orchestration of the voting process.

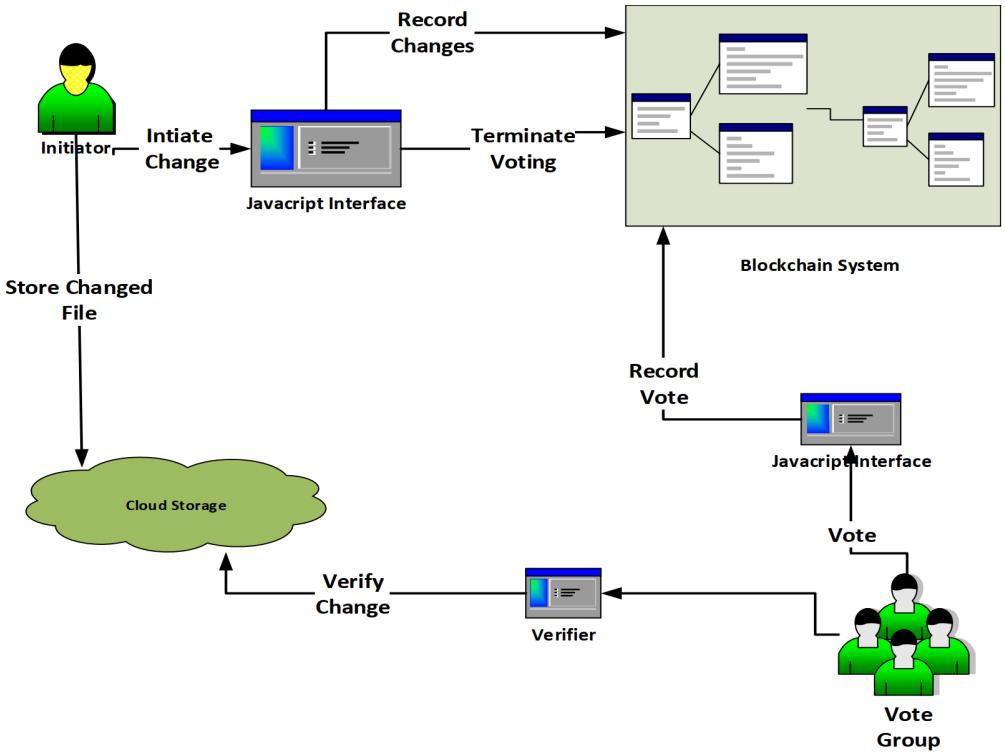

Figure 3.1: DataProv provenance data input flow as presented by Ramachandran et al. [80]

3.2.3.1 On-chain module

This module implements all the necessary security and privacy constraints. Since, the security of the system depends on this module, we analyze it in more detail. It is implemented using smart contracts which are code on the Ethereum blockchain that is run by all the nodes in the network, decentralizing the responsibility of the security of the system among the nodes of the Ethereum network. The code has the necessary instructions to implement the business logic. The module consists of two smart contracts as described in the following sections.

Document Tracker Contract

This contract implements the logic necessary to interact with the artifacts. This involves the access control policies and user access information. The contract also stores the provenance data in the form of events which content is signed by the user who created it, using asymmetric encryption. The access control management actions implemented by this contract are:

- Create an artifact
- Manage (grant or revoke) users rights to add provenance data to an artifact
- Generating and adding provenance data to an artifact

It is stated that this contract does not store any sensitive information in plain text (unencrypted) on the blockchain (which is publicly viewable).

Provenance data lifecycle The provenance data lifecycle implemented by this contract starts when a user creates an artifact and adds it to the system, becoming its owner. The contract enforces the constraint that only the owner can change the access rights to the artifact. This contract supports has methods that allow to view the user access permissions to an artifact as well as changing the owner of an artifact. The process of submitting provenance data to an artifact is implemented by the ChangeDocument method of this contract. As already mentioned, the provenance data has to be validated through a voting round. This validation is implemented by the Vote Contract. Therefore, the ChangeDocument method of the Document Tracker Contract can only be called by the Vote Contract. The method assumes that is only called after the Vote Contract has determined that the provenance data is valid, which means that that there is no trust boundary between these two components, given that the Document Tracker Contract trusts the Vote Contract.

Vote Contract This contract implements the logic for the voting rounds. There are two types of rounds: simple majority voting and threshold voting. The voting process consists of the following steps:

- The user that changed the file submits the provenance data (encrypted) as well as the signature for that provenance data (generated with his private key).
- The Vote Contract is called and verifies the change.
- After verifying the change's validity, the contract initiates the voting round by generating a log event. The voting round will be open for a certain amount of time during which the participants can vote for the validity of the change.
- For each vote submission, the contract verifies its validity (user access permissions to the artifact and vote round).
- When the time for the voting round ends, the Vote Contract checks if the number of votes have reached the minimum threshold of required votes. If not, the contract does the necessary procedures to restart the voting procedure and then restarts it, going back to step 1. If yes, the contract checks if the majority has approved or rejected the change. If the majority voted no, the change is discarded. If the majority voted yes, the change if submitted by calling the ChangeDocument method of the Document Tracker Contract, as already mentioned above.

This contract enforces the constraint that only one voting round can be active at the same time for the same artifact, ensuring that all provenance data submissions are ACID compliant.

The execution flow for voting rounds is shown in Figure 3.2.

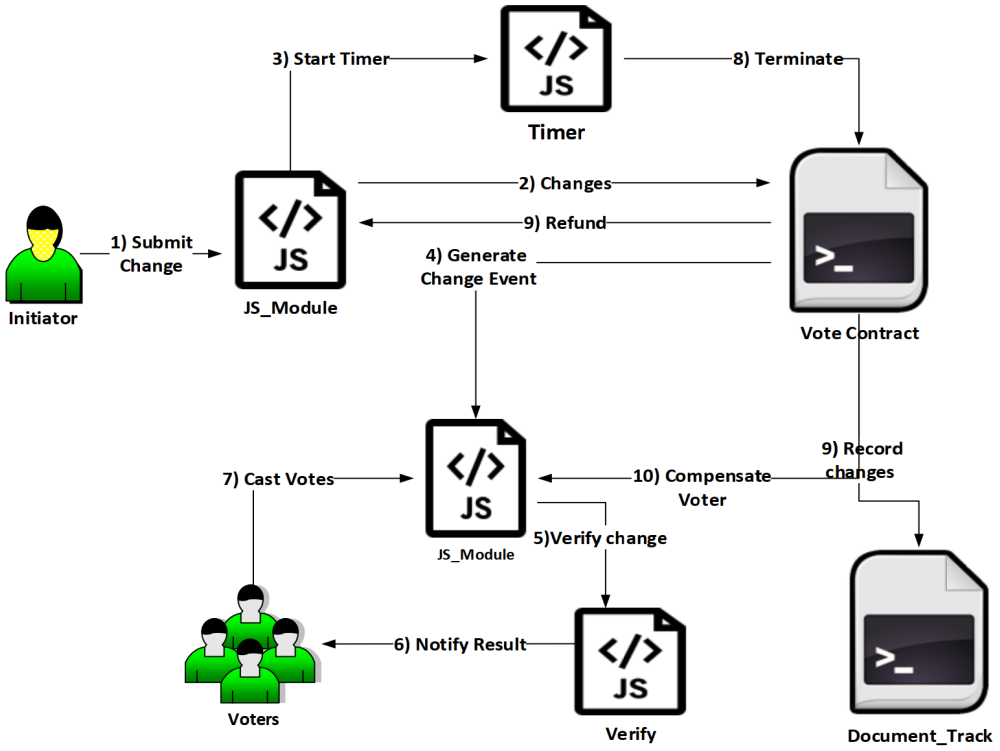

Figure 3.2: DataProv voting round flow as presented by Ramachandran et al. [80]

3.2.4 Off-chain module

The off-chain module does implement the security of the system as it is just a client application, running independently in every user of the system and which actions cannot affect other users as long as the security constraints are well enforced in a decentralized way by the smart contracts running on the Ethereum network. Any malicious modification on this application should not be able to affect the security of other users of the system. Therefore, we don't describe this module in detail as we are only interested in understanding how the system enforces the security constraints.

The module is implemented using JavaScript and runs on the browser of each user. It works as a client-side application that interfaces with the back-end implemented as Ethereum smart contracts. In addition, this module is also composed by a script residing not in the client-side application but in the cloud storage location where the artifact is stored. This script allows to check the validity of the each provenance data submission of a particular artifact. This module can be divided into submodules, as described below.

3.2.4.1 Client Interface Module

This module acts as an interface between the user and the smart contracts. It has support for the user to perform the following actions:

- Create new artifact

- View history of artifact changes

This module also generates the digital signatures for the users's operations, using the private key.

3.2.4.2 Event Watcher Module

This module watches the Vote contract. It is notified whenever a new voting round starts and checks if the user that is using the interface is one of the voters. If yes, the module decrypts the information, verifies if it is valid (by checking the signature) and calls the verifier script. If the script returns that the change is valid, this module votes yes for the user. The process is automated because the script should have the necessary instructions to verify if the provenance data is valid or not.

3.2.4.3 Timer module

This module implements the timer for the voting rounds. It is set when the round begins and generates an event when the round ends.

3.2.4.4 Verification Script

This module is not part of the JavaScript client-side web interface but resides in the cloud storage instead. This script is responsible for verifying the changes made to an artifact. This verification script implements the validation of the hash submitted as being the hash of the new version of the artifact (checks if they match). However, the script should also implement other verifications that are not possible to generalize and should be implemented according to the needs of each research scenario. This automatic verification of changes may not be trivial to implement and sometimes, especially in the context mentioned by the author, need the intervention of a human in order to be possible to perform.

3.2.5 System Analysis

This is the most important section. In this section we analyze the security and privacy of the system. Some of the aspects are already mentioned in the DataProv paper but there are other considerations missing that are worth to be pointed out.

The author organizes the section in types of attacks. The terminology used to describe the type of attackers is similar to the one described already in the Threat Model section 3.2.1.2: external and internal attacker.

3.2.5.1 Security Analysis

Security consists of protecting the system against a potential attack. In this section we provide a more detailed analysis of the possible attacks to the system and how the system should react to such threads. Firstly, we analyse the possible attacks by an **External attacker**. Then, we analyse the possible attacks by an **Internal attacker**.

External attacker The author analyses the effect of an attack by an external attacker, covering two scenarios:

Unauthorized modification attempt Firstly, the author describes the attack where an external attacker attempts to modify a document that he doesn't have permission to. As already mentioned above, this attack is very easy to mitigate since the user does not have access permission to the artifact. This constraint is enforced by the Document Tracker contract which would immediately reject any change requests from users that don't have access permission to the document attempting to attack. This contract only accepts change requests from users who have been granted permission to modify the document. The contract also penalizes the users attempting to attack the artifacts since to open up a change request a deposit amount is required. It is part of the contract that whenever a user signs a modification request to an artifact, the deposit amount is at stake and will be lost if he attempts an attack. The consensus protocol will make sure to remove that amount from the user's account. In this case, it is not mentioned where that money goes. A possibility would be to go to the owner of the document. This penalty system helps preventing denial of service attacks by flooding the network with fake artifact change requests, mobilizing a lot of computing power for nothing. With this penalty system, flooding the network has a cost and the attacker would eventually run out of money before overloading the network. A temporary overload could be possible but that would only reward the honest users and remove money from the attacker's account.

Replay attack This type of attack is also described by the author and consists of replicating a legitimate modification of a document by an honest user that has access permission to that document, by submitting earlier versions of the artifact, causing it to be changed possibly multiple times. This is type of attack is mitigated by the Document Tracker contract which logs the timestamp of the latest change to the document and immediately rejects any change which has a timestamp which respects to an instant that is prior to the one of the latest change. (Re-submitting the latest change is also a possibility, although not mentioned in the DataProv paper, but that is also easy to check by changing the rejection constraint to less than or equal to instead of only less than. Another possibility would be to check if the document is equal to the latest, which should already be a constraint for reasons other than this specific type of attack. However, the first approach is preferable to mitigate this attack since it is computationally easier to perform, just as security checks should be in order to prevent the attack from being transformed in a Denial of Service attack, using a lot of computational power of the network.)

Internal attacker The author also analyses the effect of an attack by an internal attacker. An internal attacker has been granted access permission by the owner of the document. Furthermore, it is stated that the owner of a document could be an attacker, but we don't consider that because it not only doesn't make sense but also there are no defense mechanisms against this particular case. We will now describe the possible scenarios of attack by an internal attacker:

Fake changes to a document This type of attack consists of the attempt of a user that has access permission to an artifact to submit fake changes to an artifact, possibly corrupting the provenance data. This type of attack is mitigated by the voting system which mitigates this attack as long as more than 50% are honest. The author has a proposed voting approach, as mentioned in Section 3.2.3.1, called threshold voting. This voting system randomizes the voters for the round, so that the user that changes the artifact does not know who will be the voters prior to the round. This helps reducing the probability of corruption through which the attacker would attempt to control the voters so they would vote for the change and eventually make it get accepted by system, corrupting the provenance data since the change was malicious.

Is is important to note that the attack, although similar to a 50% attack on any blockchain consensus protocol, is not nearly as hard to perform as such. A 50% attack on a blockchain requires that the attacker(s) can gather more than 50% of the resource used for consensus of the entire blockchain. The bigger the network, the more attackers will be interested in attacking it, but also the more secure against 50% attacks it will be. Small blockchain projects tend to be very vulnerable to 50% attacks, because there is low amount of the consensus resource securing the network. This is described in more detail in Section 2.3.4. Here, the scenario is different: the security of the provenance data validation does not scale with the size of the network, only with the size of the community of the specific artifact that is the target. The community of an artifact will never be as big as an entire network, causing it to be more vulnerable than typical blockchain consensus protocols to 50% attacks, making the problem similar to the one with small blockchain projects.

Voting no for profit There is another type that we find possible to this system that is not mentioned in the DataProv paper. The security implemented on the smart contracts attempt to prevent spamming the platform with fake changes to a document (even though assuming they would be accepted by the voting system, it could still cause harm to the system: a denial of service attack because it would spam it with fake data). This is implemented using monetary penalties for the users who submit such fake data. The user looses the amount if the change request does not get pass the initial validity verification or if it gets rejected in the voting process. If the second case occurs, the voters (which helped spot the malicious change) are rewarded in order to incentivize denouncing of such attacks. This has its advantages in fighting denial of service attacks based on submitting fake changes to documents, however it opens the doors for another type of attack.

The attack consists of always voting no for every change request, in order to attempt to get profit from it. The steps of the voting process are described here 3.2.4.2. These steps take place in the JavaScript web client. It would be very easy to change the JavaScript code (or even develop one from scratch) to interface with the smart contracts in order to perform this attack. Since the JavaScript client downloads the newest version of the artifact, runs the script on it and submits the vote according to the result, it would only be necessary to change all of these steps to one single step: vote no.

The back-end logic that implements the security using the Ethereum smart contracts would not be able to determine if the vote is truthful or not because they don't run the script on the artifact. They can't even determine if the script was even run, since it all takes place in client-side (JavaScript web application).

There is no penalty for the users who vote no or at least the DataProv paper does not mention any. This makes the attack very aptable since the user does not loose anything in voting no. If the user is never penalized for voting no and he can even be rewarded, there would be an incentive to make the application always vote no. This incentive could quickly spread over the community of the artifact, causing all of them to change the application and eventually when more than 50% had, they would be making profit out of it, corrupting the provenance data of the system and even leaving the all initiators of changes (honest users) loose all the deposit amounts for each voting round.

A possible approach to mitigate this attack would be to penalize the users who vote no when the majority decides yes. However, this is not easy to achieve since there is not the requirement of a deposit in order for a user to vote for an artifact change. The penalty for the initiator of a change works very well because they need to place a deposit amount in order to open the change request and the consensus protocol causes the amount to be transferred to the voters in case they loose the round. However, for voting there is no deposit amount required, there is nothing at stake that can be lost so it would not be possible for the network to agree which amount to remove from the attacker's account nor who it should it be transferred to. In order to be able to penalize the attackers, a deposit amount would need to be required for the users to vote for the validity of a change.

Another possible approach to mitigate the attack would be the smart contracts validating the change using the script. This would be possible since the voting process is automated. However, this would not only defeat the purpose of the whole system because the innovative aspect of the system is a decentralized network of users that run scripts because they have interest on it and smart contracts are used in order to implement some security to the system, but it would also be a heavy computation for the Ethereum nodes to run, probably making the system not profitable (which should have been the incentive to develop DataProv in the first place).

The first approach seems to be the best, however it may have other consequences on the system that we are not able to determine.

3.2.5.2 Privacy Analysis

The author also analyzes the privacy of the system. Following the same terminology, the author uses the internal and external to describe the users potentially compromising the privacy of the system. Here we will not use the term "attacker" because there is not deliberate action taken against the system, just the ability to see information that could not be supposed to be seen.

External user The external user is not part of the users who have access permission to the artifact. Therefore, he will be cryptographically unable to view the contents of the changes. It is

stated on the DataProv paper that the only unencrypted information stored in the event logs is the document id (because it is needed for the users to know if the document belongs to them, for the voting purposes). All the other information that contains the change is registered encrypted, so he cannot see the actual changes neither the content of the document. The other information that is part of the change log entries is the digital signature of the change requester, but that contains no information, it is simply an encryption of the hash of the contents (irreversible twice). The information that an external user can infer is the users that are associated with a document id by observing the iterations of the voting contract.

Internal user The internal user is part of the users who have access permission to the artifact. Therefore, he will be cryptographically able to view the contents of the changes. There is no support for privacy between users that have access to a document. Every user in the group of users that have access to an artifact can view all the modifications that all the others members of the artifact. There is a need of trust in terms of privacy between the users that have access to the same artifact. This is not necessarily a problem since the trust is not transitive to other artifacts and we can assume that a user does not mind that other users see the changes made to other artifacts.

Zero knowledge proofs It is stated that the provenance data should be defined as meta-data that describes where the raw data originated, who owns it and what were the transformations done to the data. This provenance data is expected to support the objectives of the scientific research and provide transparency over the procedures of the research so that people who were not involved in it don't need to trust on the information provided by the researches but can verify it instead. In order to achieve this, the author mentions the possibility of using zero knowledge proofs to provide verifiability of the provenance data without the need of having access to it, referencing this paper of a system that tries to combine smart contracts with zero knowledge proofs [58]. However, it is stated that the existing efficient zero knowledge proofs mechanisms are not general enough in order to be applicable to their system. This should be due to the fact that there is the possibility of customizing the verification scripts and the zero knowledge proof mechanism would need to be capable of proving the customizable conditions of those scripts which are very general. On the other hand, they state that generalizing zero knowledge proofs in order to be compatible with their system could make them computationally heavy to run (zero knowledge proofs typically require a lot of iterations in order to achieve a high probability of validity, thus making them heavy to compute [40]).

3.2.5.3 Concurrency

Concurrency problems could be considered on modifications to the artifacts in the event that two voting rounds were to occur at the same time for the same artifact. The DataProv system solves this problem by ensuring mutual exclusion between rounds for the same artifact, enforcing the constraint that only one round can be active at a time for the same artifact. This is the easiest but most effective solution for the problem, however it may have its impact in usability in the

sense that users could have the need to make concurrent changes to the same artifact. The author mentions that the system could be further improved in order to accept this feature using a diff analyzer for the artifacts and as long as the diff returned no conflicts, the voting rounds could take place concurrently.

3.2.6 Summary

DataProv is a platform more optimized for research and directed to data traceability. The system provides the user with a solution for manually introducing provenance data and for the network to achieve decentralized consensus over the validity of that same data through the execution of automated validation scripts.

The innovative aspect of the system consists of using Ethereum smart contract for enforcing the important security constraints in a decentralized way and using the consensus of the entire Ethereum network while keeping the heavy computation (verification of the integrity of the provenance data) only in charge of the nodes that are interested in that specific type of data. The security is not as good as if all nodes made the computation, but the system tries to keep a balance between the costs and security because making nodes of the Ethereum network run the heavy validation scripts would not be maintainable in terms of costs. The system also has a small security problem as pointed out in this section 3.2.5.1 but which we believe could be fixed.

This system is directed to our objective of providing data traceability for research. However, the context is somewhat different in the sense that we want the registration of the traceability information to be done seamlessly upon some processing of data instead of manually being introduced by a human. We also have privacy concerns that, although considered by the author, they are not explored enough to the point of providing solutions to our problems. The system also lacks some support for the usability by medical patients which is necessary in out context.

3.3 Approach by Moeniralam

This solution is aims to provide trust over the lineage and provenance of satellite data. Satellite data is continuously transformed in order to be usable in research, just like medical data. The system aims to provide traceability over data sets that are processed as well as a mechanism of verifying those modifications [69].

The solution uses blockchain to provide this trust in a decentralized and trustless way. The solution is intended to be transparent in the sense that all the users should be able to see the data, its provenance trails as well as verifying those. There is no mention to privacy in the system as the author intends to make transparency mandatory.

3.3.1 Background

The author states that in scientific research it is crucial to be able to trace back all modifications made to the data and verify their validity. This is a very wide scope since different scientific research fields require different solutions in order to adapt to their specific data structure. However, the objective of this solution is the closest to our topic: providing traceability when the data undergoes transformations. The project aims to provide data traceability to the Sentinel-2 Copernicus satellite of the Copernicus project. The raw satellite data is processed in order to be studied and useful in scientific research.

3.3.2 System Requirements

The author defines two key requirements for storing the satellite data:

- The data stored is immutable after its registration.
- The production environment and the production process have to be recorded in order to allow for later reproducibility of the steps, allowing confirmation of that data.

There is a distinction made clear between methods reproducibility, results reproducibility and inferential reproducibility. Methods reproducibility is defined as "the ability to repeat as precisely as possible all the processing steps done, with the same data and tools to arrive at the same results". Results reproducibility is defined as "arriving at the same results from the same data, using a different method". Inferential reproducibility is defined as"drawing the same conclusions, based on the same results of a similar study".

The author states that their focus is on "methods reproducibility in an autonomous manner". This is exactly our focus with this dissertation. The DataProv solution 3.2 lacks this "autonomous manner" part.

The author states that they want to achieve traceability which is composed by data lineage and data provenance.

There is a distinction made between workflow lineage and dataflow lineage. "Workflow lineage describes how derived data has been calculated from the original dataset, while dataflow lineage describes how data has moved through the processing chain". They aim to incorporate both in order to achieve complete data traceability [85].

3.3.3 Solutions considered

The author mentions two existing solutions as the most promising for storing and tracing digital assets: BigChainDB [12] and Ethereum [16]. BigChainDB stores the data sets on the blockchain, removing the need of trusting in a third party for storing the data and making the data storage as decentralized as the traceability data storage, but this obviously has drawbacks for high volumes of data, requiring all of the full nodes to allocate size for all the data of the system (because they need to have the entire blockchain stored) [67]. Ethereum does not store the data sets on the blockchain

and requires that a third party registers the data just like in normal database in a centralized fashion [16]. One option to achieve this could be cloud storage just like in the DataProv system 3.2. Hash representation of the data would be stored in the blockchain along with pointers to the cloud storage location so that the users, after downloading it from cloud storage, would compute the hash and compare it with the one in the blockchain, so they could verify the integrity of what is stored in the cloud storage against the hash that is on the blockchain (and over which we can have decentralized trust).

Another solution considered by the author is a project called Quality Assurance for Essential Climate Variables (QA4ECV) [72]. This solution provides support for tracking data, storing entries that compose the traceability chain. The entries contain the following content:

- Input Data
- Processing Step
- Output Data

Optionally, there may be additional meta data stored, in order to provide better knowledge of the transformation process applied to the data:

- Processing Step Details
  - Purpose of applying the step.
  - Principles of underpinning step.
  - Assumptions, simplifications and approximations.
- Principles underpinning the step
  - Dataset name and version number.
  - Justification for use.
  - Variables used.
  - Gaps, trends, discontinuities data which impact the product.

However, the author states that the system lacks the ability to verify the integrity of the data as well as data provenance. To our understanding the system consists of loose data tracking entries in the sense that the outputs are not connected to inputs of any entry. The only way of providing this full data lineage (that consists of being able to determine the data provenance) is by placing useful information in the additional meta data stored associated with each tracking entry. But that does not only lacks an automated mechanism of programmatically determining data provenance but also requires trust in who wrote that information in order to validate it.

3.3.4 Data type and structure

The system is supposed to provide traceability for the satellite data and it will be tested using the Sentinel-2 Copernicus satellite of the Copernicus project.

The types of data that need to be stored in order to achieve full reproducibility and traceability of the satellite data are:

- Datasets that are recovered
- Production environment which can include the OS where it was run
- Production process which is a list of the processing steps with human readable comments explaining why each step was made

The structure of the data processing flow for which they aim to provide traceability by 5 levels: ranging from 0 to 4. The level 0 consists of raw satellite data, without any processing made to it. The following levels represent data that was processed and that originated from the raw data of level 0. Each level transition represents processing made to the data in a pipe line fashion: level 0 data gets processed and originates level 1 data which gets processed and originates level 2 data and so on.

The data of which the project aims to provide lineage over is stored in the Google cloud service. The blockchain would store pointers to the location of each data set in the Google cloud storage location.

3.3.5 Proposed Design

The system aims to not follow the BigChainDB approach where the data sets are stored on the blockchain [12]. Instead, the proposed design consists of maintaining the data stored in the cloud (just as it currently is for their specific case of the Sentinel-2 satellite). The blockchain will contain pointers to that data as well as hashes of the data in order to achieve decentralized trust over the integrity of the data.

Proof of Stake The consensus mechanism that the system aims to implement is proof-of-stake, already explained in the Background chapter of this document 2.3.3.

The author states that each block should consist of:

- Dataset
  - Pointer to dataset (allows to retrieve the dataset from the cloud storage location).
  - Hash of the dataset (allows to check the integrity of the dataset after being download from the cloud storage location).
- Production environment

  - Pointer to the data that represents the production environment (allows to replicate the environment where the data was processed in order to validate the integrity of the lineage data).
  - Hash of the data that represents the production environment (allows to check the integrity of the production environment data after being download from the cloud storage location)

- Production process
  - Pointer to the data that represents the production process (allows to replicate the transformations that were done to the data in order to validate the integrity of the lineage data).
  - Hash of the data that represents the production process (allows to check the integrity of the production process data after being download from the cloud storage location)

The author states that he aims to implement the solution using Ethereum, in order to provide better decentralization, as a bigger network secures the blockchain, making it more immutable as explained in Section 2.3.4, p. 15.

3.3.6 Proposed Design Analysis

Firstly, this design seems to limit the traceability to one dataset per block. However, we believe the author means that each block should consist of entries having the strucutre explained above, therefore providing support for multiple entries per block.

This design makes the block validation algorithm computationally heavy since, in order to achieve as most confidence over the data stored as possible, every full node running the blockchain would need to apply the processing steps for each entry and verify if the input data of the following entries correspond to the output data of an entry before it. This could be done by having a map that associates the hash of the output (result) data to the hash the input data of all entries. Therefore, validating the blockchain would involve the following steps, for each data entry:

1. If the input data to the entry corresponds to level 0 (raw) data, skip this step. If not, verify if the input data has a corresponding entry in the map, associating it to an output of a previous entry.
2. Retrieve the dataset, the production environment and the production process from the cloud storage location.
3. Apply the transformation to the dataset, using the production environment and the production process.
4. Compute the hash of the result data.
5. Store this value in a map that associates the hash of the input data to the hash the output (result) data.

Although stated by the author that the system aims to facilitate the process of finding the provenance of the data, to our understanding the design could make it computationally demanding to find that information since the entries do not have a pointer to the entry which produced their input data. This can still be programmatically checked by searching for the hash of the input dataset and find the entry that has that same hash for the output data. This is not computationally easy since, with the proposed design, it would involve going through all entries of the blocks prior to the one we are trying to find the provenance of and applying steps similar to the ones described above. If the block validation algorithm has been already run and the map used for that same validation contained pointers to the location of the entry (block number and transaction number), it could make the process easier, but it would still require that pre-computation.

The solution to this problem is not trivial since including a hash of the output data would not provide as much confidence over the provenance data since in order to avoid trust as most as possible we would still need to apply the transformations to check if the hash is valid. A voting system for the hash of this output data could provide a computationally easier way of checking the provenance of the data just like the one proposed on the DataProv solution, explained in Section 3.2, p. 29. However, this would not be able to provide as much decentralized trust as applying the transformations and checking their validity.

3.3.7 Summary

The solution proposed by Moeniralam uses blockchain to create a system that aims to provide full traceability of each data transformation, resulting from processing of that same data as he believes that it is crucial for ensuring trust over scientific research. The solution is to be implemented using Ethereum, in order to provide better decentralization.

This solution is tight to the inherent and most important characteristic of blockchain: full decentralization. There are no workarounds provided to ease the process of the verification of the lineage data that could compromise the decentralized trust over that data. Every entry should be checked with as least trust as possible. However, this approach also has its disadvantage of being computationally heavy to verify the provenance data information.

The platform assumes that every user that is in the network has permission to view the satellite data, since there are no references to access control and private data sharing mechanisms. This is where the solution does not adapt to our context, since medical data is highly sensitive and support for private data sharing is a must have.

It is important to note that this solution has not yet been implemented and the document consists of a solution proposal. Some of the disadvantages of the design could come to attention and solutions to them be provided.

3.4 Supply Chain Solutions

This section explores some solutions for supply chain traceability.

3.4.1 Introduction

The quality of food has always been a concern of humanity. In recent years, with increasing incidence of food-related health diseases, consumers have been demanding for more awareness over the traceability of the supply chain [79]. This solutions tend to target the problem of proving that what is registered on the blockchain is indeed what happens in the real world [75].

Full awareness of the traceability of food is better obtained using a decentralized database like blockchain [73]. Since the same problem of having multiple parties with different interests involved (as the medical research sector has) there is the need of using a decentralized technology in order to avoid manipulation and corruption of data. The applications of the blockchain in supply chain are wide as they make the process more transparent [14].

3.4.2 Solutions

The author Feng Tian has proposed a solution in his paper that consists of combining the blockchain with RFID (Radio-Frequency IDentification) in an attempt to give more credibility over what is registered on the blockchain [102].

There is also an application of an ontology driven design to the design of blockchain by Henry m. Kim and Marek Laskowski [54] They store the characteristics of the products on the blockchain such as:

- time when produced
- location where produced
- environment where produced
- description

The ownership of the products used is stored in the form of linked lists that contain the successive owners of them, following the time order. Therefore, the element that appears last corresponds to the current registered owner. The timestamps of the entries are used to control user rights.

Birgit Clark and Ruth Burstall have applied the traceability to the pharmaceutics industry [22]. They explore methods for anti-counterfeiting of drugs and enforcement of authentic products.

3.4.3 Summary

Supply chain management traceability solutions aim to provide full transparency over the supply chain data and the validation that they aim to perform is not regarding operations with data already stored on the blockchain. Instead, they want to store on the blockchain what happens on the real world. We do not intend to solve this problem, as already stated before, aiming only to provide trust over data lineage to the point where it was first registered on the blockchain. All other operations should be validated by applying processing to the data, without the need of mechanisms to validate that what happens in the real world. Therefore, we don't explore the mechanisms provided

to solve this in detail, as we focus more on the validation of data that is already in the blockchain. The traceability provided by these solutions is the one inherent to blockchain in the sense that every transaction's inputs must come from a previous transaction's outputs. Therefore, the supply chain solutions provide no traceability of data transformations, making them not suitable for our case.

On the other hand, supply chain management solutions aim to provide full transparency of the information stored in the blockchain, lacking support for private data sharing, which is essential for our medical context.

3.5 Data Traceability Platforms Analysis

The state of the art research performed consists of 12 solutions in total. Firstly, in Section 3.5.1 (p. 48) we provide an overview of the 12 solutions studied in detail. We did not dive into the details of every solution, only into the most significant, the ones that are closer to solving the current issues identified. Then, in Section 3.5.2 (p. 52,) we compare the 3 most significant solutions, analysing the advantages and disadvantages of each as well as the trade-offs between them.

3.5.1 All solutions comparison

The state of the art research performed consists of 12 solutions in total. We did not dive into the details of every solution. Table 3.1 provides an overview of the solutions investigated, comparing them according to some criteria. Each one of the criteria is explained below:

Trace transformations This criterion evaluates whether the solution has support for tracing data transformations. This includes being able to verify the veracity of the traceability information.

Private data sharing This criterion evaluates whether the solution has support for controlling which users can access the data. On the first blockchain implementation of Bitcoin (as described in Section 2.3.2, p. 12) all the data on the blockchain was public. However, more implementations have been developed which allow for controlling access permission. These implementations have support for private transactions, described in the respective criterion.

Structured information This criterion evaluates the type of information for which it is provided traceability for. It specifies whether the information is free (raw and uninterpreted data) or follows some predefined structure. The type of structure influences the algorithm to validate the integrity of the data. Free data requires additional customization for the integrity verification algorithm, which can make the appliance of mechanisms like zero-knowledge proofs [40] more difficult.

Private transactions This criterion evaluates whether the solution has support for private transactions. Blockchain is a ledger that must be synchronized between all nodes of the network. The

first blockchain implementations required the transactions to be visible by everyone in the network in order to be verified. Private transactions are transactions that are only visible to certain entities, although they can be validated by everyone on the network. This is implemented using zero-knowledge proofs which allow nodes to validate information without having access to it [40].

Sharing One to One This criterion evaluates whether the solution has support for sharing data between two users only, meaning that only those users can see the data although it is registered (directly or indirectly in the form of a pointer to the storage) on the blockchain that the entire network can access. Therefore, this criterion necessarily implies having support for private transactions.

Sharing Many to many This criterion evaluates whether the solution has support for sharing data between two groups of users, meaning that only those users can see the data although it is registered (directly or indirectly in the form of a pointer to the storage) on the blockchain that the entire network can access. Therefore, this criterion necessarily implies having support for private transactions.

Anonymous transactions This criterion evaluates whether the solution has support for anonymous transactions. Anonymous transactions are similar to private transactions. While private transactions respect to the content of the transaction only being visible to the certain users, anonymous transactions respect to the source and destination entities of the transaction only being visible to certain users. Furthermore, the entire blockchain network still needs to be able to validate those transactions, although they cannot see the intervenients, which makes this mechanism also require the use of zero-knowledge proofs [40].

Group signature This criterion evaluates whether the solution has support for group signatures. Group signatures allow to create groups of users that can represent each other in the process of signing the transactions in the blockchain. The group signatures are valid without the signature of every group member. The number of members that are required to sign, in order to consider a transaction valid, is customizable.

Data repository server This criterion evaluates how the solution stores the shared data. The two alternatives considered are storing the actual data sets on a data repository server or storing them on the blockchain. The first alternative requires a pointer to the repository to be stored on the blockchain along with a hash representation of the data, in order to be able to validate the integrity of the data, without trusting on the third party that controls the repository server and relying only on the blockchain. The second alternative leads to more availability and decentralization since it doesn't rely on any third party server and all the information is stored in the blockchain, but it requires all the full nodes of the network to store all the data, thus compromising scalability. Since our problem requires storing large data sets of health data (several GB), we prefer to store the data

Solution	Trace transformations?	Private data sharing?	Structured information?	Private transactions?	Sharing One to one?	Sharing Many to many?	Anonymous transactions?	Group signature?	Data repository server?
1 - Albeyatti (MedicalChain) [2]	✗	✓	✗	✓	✓	✓	✓	✗	✓
2 - Moeniralam [69]	✓	✗	✓	✗	✗	✗	✓	✗	✓
3 - Ramachandran, et al. (DataProv) [80]	✓	✗	✓	✗	✗	✗	✓	✗	✓
4 - Xia, et al. (MedShare) [112]	✗	✓	✗	✓	✓	✓	✓	✗	✓
5 - Azaria, et al. (MedRec) [10]	✗	✓	✗	✓	✓	✓	✓	✗	✓
6 - Huang, et al. [48]	✗	✓	✗	✓	✓	✓	✓	✓	✓
7 - Creydt, et al. [27]	✗	✗	✓	✗	✗	✗	✗	✗	✗
8 - Vinay, et al. [106]	✗	✗	✓	✗	✗	✗	✗	✗	✗
9 - George, et al. [38]	✗	✗	✓	✗	✗	✗	✗	✗	✗
10 - Liao, et al. [62]	✗	✗	✓	✗	✗	✗	✗	✗	✓
11 - Tian [102]	✗	✗	✓	✗	✗	✗	✗	✗	✓
12 - Clark, et al. [22]	✗	✗	✓	✗	✗	✗	✗	✗	✗

Table 3.1: State of the Art solutions comparison

in a repository server, otherwise the size of the blockchain would be impractical to store by every full node of the network.

The solutions that are more suitable to our general context of sharing personal health data (described in Section 2.5, p. 18) are (1 - MedicalChain), (4 - MedShare) and (5 - MedRec). Thereby, the solutions are similar in the sense that they aim to serve patients, providing them the ability to control who can access the data, focusing the access control problem. Thus, both solutions are strong in fulfilling our second requirement of **Private data sharing**, by providing **Private Transactions** and **Anonymous transactions**. They provide the feature to control the sharing mechanism, allowing to share the data to single or multiple entities, which is useful to our context since we may want multiple parties to be able to access the data or just one. However, these solutions lack support for scientific research in the sense that they don't provide data traceability when the data undergoes transformations (that are necessary for the research process). This support is required since there is the need to avoid trust in our more specific context where there are multiple entities with different interests involved (as explained in Section 2.6, p. 19). Therefore, both solutions fail at **Trace transformations** criteria, which is the most important for our problem.

The solutions that are more suitable to our specific goal of data traceability are the solution proposal by Moeniralam (number 2) and DataProv (number 3). However, these solutions don't target our specific context of health data (described in Section 2.5, p. 18) and, therefore, they allow access to the data by the entire network. This leads to the lack of privacy of the data which is to be traced. In the case of the solution proposal by Moeniralam, verification of the provenance

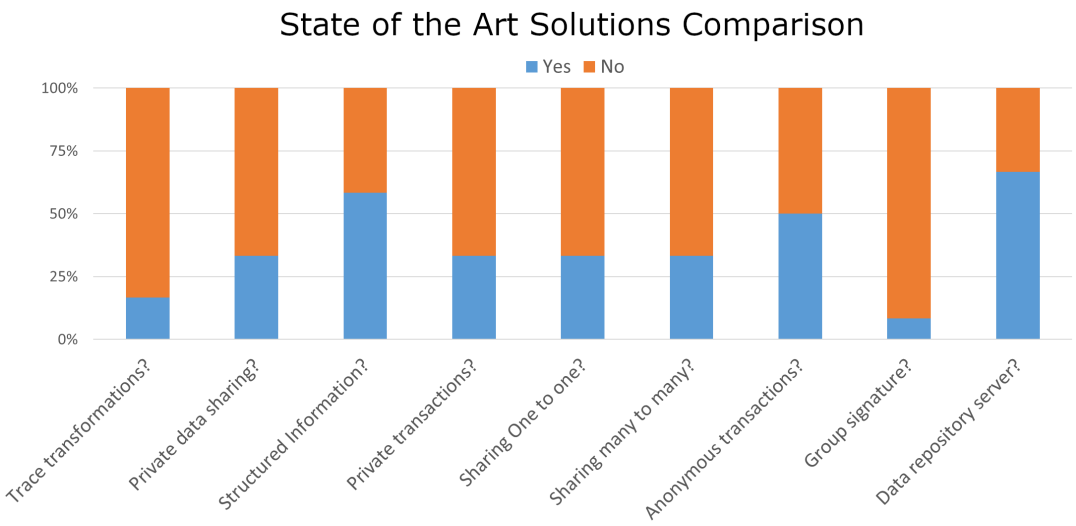

Figure 3.3: State of the Art solutions comparison chart

data also leads to lack of privacy of the data itself, due to the fact that it requires access to the data in order to perform the verification. The trade-offs between the two solutions resides in the amount of decentralization and ease possible in the verification process (as explained in Section 3.5.2, p. 52).

The solution number 6, by Huang, Chen and Wang (Blockchain-based multiple groups data sharing with anonymity and traceability), aims to support the process of sharing data for general purposes. The solution is strong at customizing access permissions, allowing for the user to customize who can access the data. The solution allows to share the data to single or multiple entities, which is useful to our context since we may want multiple parties to be able to access the data or just one. Since the solution does not target the context of Health Data (described in Section 2.5, p. 18), it has the same problems as the solution proposal by Moeniralam (number 2) and DataProv (number 3): lack of privacy of the traceability information and, in order to allow verifying it, lack of privacy of the data itself, which makes it not suitable for the health data sharing context. This solution was the only one to be able to provide the **Group signature** feature.

The remaining solutions aim to provide traceability to the supply chain context. Although providing traceability, these solutions target the problem of verifying that what is registered on the blockchain is actually what happens on the real world, which is a problem that we are not trying to solve. The problem that we are trying to solve is supporting data traceability and providing verifiability of that information, knowing the transformations that were done to the data.

The Figure 3.3 provides an overview of the trends between the solutions analysed, using the same criteria as the table. The sample is the same as the table: 12 solutions in total.

As can be seen on the chart, only 2 (16%) of the solutions can support traceability of data transformations, which is one of the most important requirements for our specific context where there are multiple entities with different interests involved (described in Section 2.6, p. 19). Another

important trend is that only 4 (33%) of the solutions support **Private data sharing** mechanisms. There is no solutions that can combine both of these criteria, which are the most important for our context. Approximately half (58%) of the solutions impose rules regarding the structuring of the information. There are 6 (50%) of the solutions supporting **Anonymous transactions** but only 33% of the solutions support **Private transactions**. Less than half (33%) of the solutions allow to customize the sharing of data to just a single or multiple entities. There is only one solution implementing the **Group signature** feature. And the most filled criterion is the use of a **Data repository server** to store the data, with 66% of the solutions using a data repository server. Note that there are solutions that don't mention how they store the traceability information which means that the other 33% may not necessarily store all the information on the blockchain.

3.5.2 Top 3 solutions analysis

The 3 solutions explained in detail are the ones that most adapt to our context and requirements. The multiple supply chain solutions provide traceability but target the problem of enforcing the consistency with the real world, instead of being able to cryptographically provide trust over data lineage and data provenance.

The solution that is closer to our context of Health data sharing is MedicalChain [2]. It provides the patients with control and awareness over what happens to their data.

However, it lacks our most important requirement: support for research projects. This support requires data traceability in order to be able to provide data lineage and trust over data provenance. This is where the solution proposal by Moeniralam [69] shows its strength, aiming to provide data traceability for data transformations.

Another important solution that aims to provide traceability is the DataProv platform [80]. However, this solution is based on the manual user input of traceability information which is not suitable to our context.

The Moeniralam solution proposal [69] and the DataProv platform [80] are the ones more driven to data traceability for scientific research. Their main difference is in the trade-off between decentralization and ease of verification. The Moeniralam solution proposal aims to provide full decentralization, not storing any information to help the verification of the integrity of the data that could possibly not be truthful. The verification of the data integrity in this solution is done by applying all the necessary computations, therefore avoiding the need of trust as most as possible, taking full advantage of the most important characteristic of blockchain: decentralization [73]. This has the drawback of making the verification process computationally heavy. On the other hand, the DataProv platform implements a voting system where the users interested in a certain type of research data vote for the validity of the traceability information. The voting group is very small when compared to the size of the entire blockchain network. This is important to consider since in decentralized systems that are secure if more than 50% of the users are honest, the bigger the network, the more difficult it is to gather more than 50% of the consensus resource, making it more difficult to cheat the system, as explained in Section 2.3.4 (p. 15). Since the Moeniralam solution proposal takes advantage of the entire Ethereum network to secure the information, we

	Private data sharing	Traceability of data transformations
MedicalChain (2017)	✓	✗
DataProv (2017)	✗	✓
Moeniralam (2018)	✗	✓

Table 3.2: Top 3 solutions comparison

can clearly put more trust into its validity than in the small group of voters that approved the traceability information in the DataProv platform. However, in order to verify the information in the Moeniralam solution proposal, we would need to go through a lot more computation than if we were using the DataProv platform.

The DataProv platform [80] also provides some privacy, over the data but not as much as MedicalChain [2], that is highly focused on privacy through access control. On the other hand, the Moeniralam solution proposal [69] assumes that every user with access to the blockchain can see the data and its traceability information, providing no support for private data sharing.

MedicalChain [2] is the solution closer to our requirements in terms of data privacy due to the context being the same but it is also the one farther from our data traceability requirement for scientific research. The Moeniralam solution proposal [69] is the solution closer to our requirements in terms of data traceability for scientific research but it is also the one farther from our data traceability for scientific research requirement. The DataProv platform [80] is the one in the middle providing better data traceability but worse privacy than MedicalChain and better privacy but worse data traceability than the Moeniralam solution proposal.

Table 3.2 provides an overview of the differences between the MedicalChain, DataProv and the solution proposal by Moeniralam.

Table 3.3 provides an overview of the main differences between DataProv and the solution proposal by Moeniralam, showing that they are complementary solutions. We want to levarage the strong points of each to create a solution that most adptable several use cases as well as further adapts to our context of health research data sharing.

Neither of the solutions encompass the desired goal: support for private data sharing and traceability of data transformations. We also aim to provide the best trade-off between decentralization and ease of verification, providing both mechanisms, giving users the ability to choose between the two.

	Quick and Less Decentralized Validation Mechanism	Slow and Fully Decentralized Validation Mechanism
DataProv (2017)	✓	✗
Moeniralam (2018)	✗	✓

Table 3.3: Data transformations traceability solutions comparison

3.6 Frameworks Analysis

Blockchain is a recent technology with growing interest and research around it, mostly due to its immutable and decentralized capabilities [98]. Although in an early stage, there are frameworks available to develop blockchain applications [100].

This section presents technologies for developing the prototype and the rationale used to pick the most suitable ones. The prototype implements the approach to solve the problems explained in Chapter 4 (p. 61). Since the approach uses blockchain, described in Section 2.3 (p. 11), frameworks to develop blockchain applications were investigated and experimented to see which one was more suitable to implement the approach. The two main technologies investigated are Hyperledger Fabric [4, 19, 1] and Parity Substrate[1].

3.6.1 Parity Substrate

Parity Substrate was investigated and tested first. The development language supported is Go programming language [109] and was first released in May, 2018. The following paragraphs explain the most important aspects about Parity Substrate for our decision.

3.6.1.1 Architecture

Parity Substrate aims to follow a modular architecture, with its components organized in pellets. Every component of the system should be implemented in a new pellet. A pellet is just a pluggable piece of code that implements a certain module or component of the system. This is expected to provide a modular and extensible architecture [96] with responsibilities of a certain module well isolated from the other modules. The database is implemented as a key-value storage [96] and on top of it a modified Patricia Merkle tree is implemented [55] in order to provide quick existence check of a certain item. The network layer is implemented using libp2p [61], a modular P2P network protocol (described in Section 2.2 (p. 7)). The available consensus engines are Proof of Work (described in Section 2.3.3 (p. 13)), Aura (Authority Round) [8] and Polkadot consensus [101].

3.6.1.2 Usage

According to the documentation [2], substrate is supposed to be used in one of three ways:

Substrate Node is useful when the least amount of customization is necessary and only allows to change the genesis parameters of the included runtime modules (balances, staking, block-period, fees, governance, etc). It provides the user with the most ease of development at the cost of loosing customization.

[1] More information about substrate can be found in the official website https://www.parity.io/substrate/
[2] Substrate's documentation can be consulted at http://https://substrate.dev/docs/en/

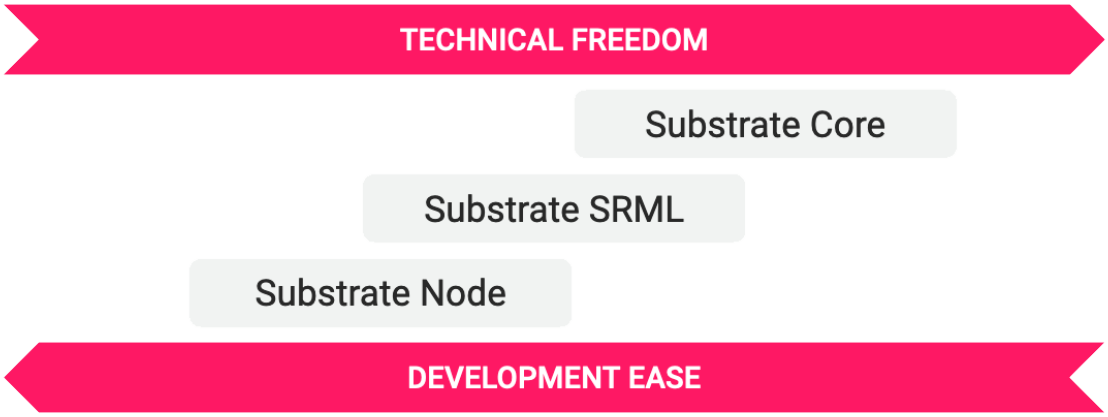

Figure 3.4: Parity Substrate Usage Trade-offs as presented in the official substrate documentation

Substrate FRAME is useful when the moderate amount of customization is necessary and provides the user with a large amount of freedom over the blockchain's logic, allowing to customize the datatypes, add modules from the predefined library and add custom modules. It provides the user with some customization at the cost of loosing development ease.

Substrate Core ignores the entire FRAME and all the runtime logic should be designed and implemented by the developer. It is necessary when a high amount of customization is required and is the most difficult to develop.

The tradeoffs between technical freedom and development ease can be easily seen on Figure 3.4.

3.6.1.3 Testing

A test network setup was used to test the technology by following the 5 setup tutorials[3]. The setup allowed us to perceive the development and code style of the framework by analyzing the examples as well as practice by attempting to introduce minor changes. Our impression was that substrate code style, combined with steep learning curve of the Go programming language, would require a significant amount of time to learn.

In order to implement the voting rounds, through which the network would be able to validate the traceability data without a lot of computational work, a relatively complex logic involving state changes would need to be programmed. We quickly realized that, in order to implement our approach (that requires customizing the blockchain and consensus logic), we would need to program (at least) at FRAME (intermediate) level, in order to customize the blockchain logic. This

[3]More information about Fabric can be found in the official documentation https://substrate.dev/en/tutorials

would require us to write a solution that is strong in logic with a coding style and language that already seemed challenging. This immediately turned into a drawback of this framework.

3.6.2 Hyperledger Fabric

Hyperledger Fabric was investigated and tested after. The development languages supported are Java, JavaScript and Go programming language and was first released in September, 2016. The following paragraphs explain the most important aspects about Hyperledger Fabric considered for our decision.

3.6.2.1 Architecture

Hyperledger Fabric aims to provide pluggable consensus, allowing applications to be more customizable so that it's more adaptable to particular use cases and trust models [1]. This framework was also architected with modularity in mind, providing pluggable consensus, allowing applications to be more customizable so that it's more adaptable to particular use cases and trust models as well as identity management protocols such as LDAP or OpenID Connect [65, 34]. The developer should implement the business logic using smart contracts. Fabric is composed by the following modular components:

- Pluggable ordering service that establishes network consensus on the order of transactions, bundles them into blocks and broadcast them to peers.
- Pluggable membership service provider that provides identity management by associating entities of the network with cryptographic certificates [15].
- Optional peer-to-peer gossip service that disseminates the blocks (output by the ordering service) to other peers.
- Smart contracts (also called Chaincode) that run within a container environment (e.g. Docker). These smart contracts can be written in any of the general purpose programming languages like Java, JavaScript and Go.
- Database Management system that stores the information that the blockchain network has saves and can be accessed by the smart contracts.
- A pluggable endorsement and validation policy enforcement that can be independently configured per application and defines how strong is the validation process of the results of the smart contracts.

3.6.2.2 Usage

According to the documentation [4], in order to develop using Hyperledger Frabric, some of the most important design features that were built into Hyperledger Fabric Model are:

[4]Fabric's documentation can be consulted at https://hyperledger-fabric.readthedocs.io/en/release-2.0/whatis.html#hyperledger-fabric

- Assets
- Chaincode
- Privacy
- Security Membership Services
- Consensus

Assets are anything with value in the real world ranging from the tangible (real estate and hardware) to the intangible (contracts and intellectual property). They can be seen as classes in Object Oriented Programming Languages like Java and have multiple properties just like attributes of classes.

These assets can be stored and then accessed and modified by the smart contracts, through chaincode transactions. Every modification to an asset represents a state change to that same asset. The ledger stores information of assets' last states as well as all its state changes.

Chaincode is general purpose programming language code that runs on the blockchain (on peers that belong to that blockchain). This code has access to the assets stored on the blockchain as well as all its history of state changes. The smart contracts can also modify those assets, through transactions. The amount of nodes that needs to run a certain smart contract in order for its result to be accepted is customizable through the Security Membership Services.

State transitions of assets are a result of chaincode invocations by external applications. These chaincode invocations are called *transactions* and the code that is run upon invocation is defined in a function, similarly to endpoints in a REST API. Each transaction results in a set of asset key-value pairs that are committed to the ledger as creates, updates, or deletes.

Privacy Channels and private data collections enable private and confidential multi-lateral transactions.

Security Membership Services Hyperledger Fabric uses a permissioned membership approach that provides a trusted blockchain network, where participants know that all transactions can be detected and traced by authorized regulators and auditors. All participants must be authenticated through the identity providing process and should have a digital certificate [15] through which all their actions are authenticated through signing and can be verified by other nodes of the network. For more information on how this process occurs, please read Section 2.2.5 (p. 9).

Consensus is achieved ultimately when the order and results of a block's transactions have met the explicit policy criteria checks.

	Parity Substrate	Hyperledger Fabric
Language	Go	Java, JavaScript, Go
First Release Date	May, 2018	September, 2016
Community	Small	Large
Coding style Learning Curve	Steep	Light
Language Learning Curve	Steep	Light
Pluggable Consensus	Yes	No
Identity Management	Limited	Wide

Table 3.4: Framework solutions comparison table

3.6.2.3 Testing

A test network setup was used to test the technology by following the tutorials in the documentation. The setup allowed us to perceive the development and code style of the framework by analyzing the examples as well as practice by attempting to introduce minor changes. We immediately noticed the familiar fabric coding style, combined with the possibility of using the Java programming language that beyond already being familiar, is a well structured language, given its strong typing quality gave a strong advantage to Hyperledger Fabric. The wide language support was an inevitable strong point in favor of fabric, especially given the strong logical nature of the code needed to develop that can be more effectively implemented in strong typed languages.

3.6.3 Comparison

The Table 3.4 provides a comparison between Hyperledger Fabric and Parity Substrate according to some criteria.

Hyperledger Fabric offers a wider variety of languages supported for writing smart contracts and also proves to be more mature with a bigger community than substrate that was released almost two years after. At the moment of choosing (March, 2020) the Hyperledger Fabric framework was 3 years and 9 months old whereas Parity Substrate was only 1 year and 10 months old. The learning curve of the programming language as well as the coding style are much more steep in the Parity Substrate framework, requiring investing time on it that could be used to improve the solution. Hyperledger Fabric supports Pluggable consensus that can be important to develop the incentive system we aim to implement. The identity management system is very limited on Parity Substrate whereas on Hyperledger Fabric it supports OpenID Connect and LDAP which are important in order to well identify the entities of the ecosystem as well as to adapt to the current authentication system so it can later be in1tegrated in the context of health data research. These aspects have put Hyperledger Fabric well ahead Parity Subtrate, causing it to be picked to implement the solution.

3.6.4 Language Choice

After picking Hyperledger Fabric over Parity Substrate it was time to decide in which general purpose programming language would the chaincode be written, since it supports as much as 3

Language	Learning Curve	Typing	Concurrent
Go	Steep	Strong	Yes
JavaScript	Very Light	Weak	Yes
Java	Very Light	Strong	No

Table 3.5: Hyperledger fabric languages comparison table

programming languages: Go, Java and JavaScript. As already state before, Go quickly falls out of choice due to its steep learning curve (which has also played a role in picking Fabric over Substrate), but lets keep it in the list of compared languages so we can provide the widest comparison possible.

Table 3.5 provides a comparison of the three languages supported by Fabric to write the chaincode, which implements the blockchain logic. The learning curve of Go is the steepest, whereas Java and JavaScript both have a light learning curve, allowing us to invest more time in architecting and implementing the incentive system, which is an important component and requires a complex architecture, being the most time demanding component of the system. The incentive system we aim to develop also has an architecture that is complex in logic, involving state changes, appropriate actions based on the state as well as byzantine fault tolerance [31]. The logic complexity of the system makes it more bug prone. The use of a strongly typed programming language spares a lot of development time, since the programmer would not loose track of the object types and the compiler would impede some of these bugs at compile time, rather leaving the programmer write almost anything and mask the bugs, making them only detectable at runtime and, therefore, more difficult to spot. This clearly puts Go and Java languages ahead of JavaScript since the first two are strongly typed and the last one is weakly typed. The concurrency in languages, when not needed, can be a drawback especially for code that has a complex logic architecture [3] since the programmer can easily loose track of the asynchronous execution flow while being focused at the business logic. The use of a synchronous language can provide confidence that the code will be executed exactly in the order it is written, leaving the programmer concentrate fully on the logic. This becomes even more important in implementations that have a complex logic architecture, like our case. Since Go and JavaScript both are concurrent and Java is synchronous, this places Java at the top of list, as long as this criterion is concerned. After considering all the criteria, Java is the only familiar language with strong typing and no concurrency, causing it to be the first pick as the language for writing the chaincode in Hyperledger Fabric.

This ends the process of choosing the framework and programming language for developing the prototype, being Hyperledger Fabric and Java the chosen ones.

Chapter 4

Problem Statement

This work aims to support health data sharing by supporting traceability of the data transformations and provide the ability to determine the provenance of the data. Electronic health records are highly sensitive data as described in the Background (Chapter 2). The current solutions described throughout the state of the art analysis (Chapter 3) are not complete enough to support the traceability of important data. In Section 4.1 (p. 61), we identify the limitations of the solutions presented and state the assumptions made for the problem that we are trying to solve, in Section 4.2 (p. 62). Then, we establish the hypothesis, in Section 4.3 (p. 63) and research questions, in Section 4.4 (p. 64). In addition, we will also make a proposal and establish the respective validation methodology.

4.1 Current Issues

In the state of the art research (Chapter 3), some solutions analysed that are intended to provide privacy and traceability to support sharing of important data. However, none of these solutions encompasses fully the desired goals. The main issues with these solutions are:

Lack of support for private data sharing Most of the solutions analyzed were not able to suit the needs of the health data sharing context since the data is highly sensitive from the social and the economic point of view and therefore requires a private data sharing mechanism. All of the solutions except MedicalChain [2] and the solution proposed by Huang, Chen and Wang [48] have no support for advanced private data sharing mechanisms. The requirements demand different levels of privacy and the ability to control which users are allowed to see which data since patients want to keep their data as private as possible. The solution by Huang, Chen and Wang [48] is the one less suitable because the only requirement it fulfills is the private data sharing, but lacks all the others. MedicalChain [2] does fit other privacy requirements but lacks support for tracing data transformations which is essential to our scientific research context.

Lack of support for tracing data transformations None of the solutions analyzed are able to suit our context of health data research in the sense that they lack features to support scientific research. One of the requirements is being able to support tracing data transformations and being able to provide trust over data provenance. This requires a mechanism to verify the veracity of the traceability information.

The solutions closer to solving these problems are the solution proposed by Moeniralam [69] and DataProv [80]. Yet, they still fail at fully encompassing the desired objectives. The following subparagraphs explain the problems found with these solutions.

Lack of a computationally easy mechanism to verify the traceability data Moeniralam [69] targets more the traceability of the data transformations problem. However, this solution assumes that everyone can see the data as well as the traceability information of it, making it not suitable for our context for the reasons already described above. This solution is strong in decentralization but it also lacks a computationally easy process of verifying the traceability information's veracity, as explained in detail on the state of the art, Section 3.3.6 (p. 45).

Manual traceability data introduction DataProv [80] also targets the data traceability for scientific research problem. However, this solution is based on manual user input of the traceability information through a user interface and our project requires this process to be more automated.

Lack of fully decentralized mechanism to verify the traceability data The solution allows for validating the traceability information with low computational effort but it also sacrifices decentralization in order to achieve this, providing no choice for the user to decide the trade-off between decentralization and ease of verification. Furthermore, this solution does not provide different levels of privacy for the users to choose when they share data, making it not suitable for sharing health data.

4.2 Assumptions

To solve the problems listed above, we take into account some assumptions of the health data sharing context as well as the iReceptorPlus project that support the validity of the proposed solution and make it relevant for the context.

Research projects want to have access to as much data as possible We assume that the disease treatment research projects want to have access to as much data as possible in order to make more progress in scientific research.

Research projects are competitors of each other We assume that the research projects are competitors of each other in the sense that they want to achieve as many progress as possible and faster than other entities so that they can get recognition of their work.

Research projects don't trust each other As a result of being competitors, research projects don't trust each other with information they haven't yet used to achieve credit (for example, through publications). This disincentivizes cooperation in the system, holding back progress for these research projects.

Research projects are not sure of which are the processing procedures of each other Partially due to the lack of trust and consequent secrecy of procedures and partially due to a lack of a feasible way of sharing processing procedures, research projects are currently not sure of which are the processing procedures of each other. This hold back cooperation and, therefore, progress in the sense that it makes them unable to reproduce the processing procedures of each other.

The traceability information respects to what is registered on the blockchain We do not intend to provide traceability of everything that happens in the real world which is the main focus of the Supply Chain traceability solutions discussed in Section 3.4, page 46. We will provide traceability of the data transformations that are registered in the blockchain as well as a mechanism to verify that traceability information with as least trust as possible. The traceability provided starts at the moment the health data is registered in the blockchain.

The patients could be rewarded by the research projects Traceability of the health data provides patients trust over what is done with their data. This helps making them more confident that research projects will not do anything that endangers their privacy. However, providing trust over the fact that nothing bad will happen may not be enough to convince people to supply their health data if they don't have any advantage over it. This advantage could come from a reward in the event that their health data contains important information to make progress in disease treatment research. We assume that this reward does not come from the blockchain and that patients will not be block validators, but instead it would come from the research projects, since they would profit with that patient's data.

4.3 Hypothesis

The problem can be defined as a high-level hypothesis that serves as a basis for this research:

"*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.*"

This work aims to provide traceability over the health research data transformations from the moment it is registered on the system as well as a mechanism to verify the veracity of that traceability information. This traceability data balances the interests of the different entities involved and increases cooperation between them, so that entities will have confidence over the data processing procedures of each other. We aim to incentivize cooperation and keep entities more involved in the process through an incentive system that should act as a feedback loop of cooperation increasing.

The system also provides the ability to trace the data transformations until the origin of the data, providing data provenance regardless of how many transformations occurred to that data. This provides the ability to identify from which source is the data that helped a certain research project, thus allowing the patients to be rewarded for their contributions. The possibility of tracing until the patient that provided the data, depends on the extension of the information registered by the entities on the blockchain.

The solution will use blockchain in order to be able to provide this confidence with as least trust and as most decentralization as possible. These two aspects are key for the context since we have multiple entities with different interests involved and research projects are not trustable, thus making it very important to use a decentralized database like blockchain which eliminates the need of trust in a central authority which could have interest in tampering the data. Lastly, we think that the solution will make patients more confident to provide research projects with their health data, thus allowing them to make more progress.

4.4 Research Questions

Taking into account the statement made before, we now decompose the hypothesis presented, into several research questions:

RQ1 "*Does providing a decentralized registry where entities can register traceability information for public data sets provide a feasible way of evaluating an entities processing procedures?*" This question aims to verify if providing a service that consists of a decentralized registry (blockchain) where entities can register the processing steps and the output for public datasets which other entities can view provides a feasible way of evaluating an entity's processing procedures, in order to build trust on that entity given the transparency provided.

RQ2 "*Does providing a mechanism to quickly agree on the validity of the entries of that registry (with less decentralization, through the voting rounds) further improve solving the problem?*" This question aims to verify if providing a mechanism to quickly validate the entries of that registry (with less decentralization than the one with data storage, but still decentralized, through the voting rounds) further improve solving the problem (without nodes having to run the transformations themselves, which is computationally expensive).

RQ3 "*Does providing full validation and voting rounds make the system more suitable for all use cases?*" This question aims to verify if providing both validation mechanisms (trusting the entire network and only the data that is stored and running the processing steps, increasing decentralization but increasing computational effort used) and (trusting the validators of the voting round which will always be less the all the nodes running the network) makes the system more suitable for all use cases in the sense that different scenarios require different levels of trust.

RQ4 "*Is reputation a good incentive resource to improve cooperation on the system?*" We aim to analyze if using reputation as an incentive resource that is rewarded upon honest behavior and removed upon dishonest behavior is a good incentive for entities to be honest and to keep everyone engaged in the process of validating the blockchain.

4.5 Methodology

Firstly, we explore the fields of blockchain and health data, in order to support the process of sharing personal health data using blockchain. Our main focus is on understanding the interests of the research entities in order to support them and incentivize cooperation. Then, we **architect an approach** that balances the interests of the different entities involved and increases cooperation between them, so that entities will have confidence over the data processing procedures of each others.

In order to fulfill these objectives, we produce our first contribution which is an **analysis of the current most suitable solutions** to our problem in order to balance the trade-off between the existing alternative approaches as well as point out the problems with some of those aspects alone. Then, we **analyse the current frameworks** available to develop blockchain applications, in order to determine which one suits more the objective we want to achieve. As a result of the state of the art analysis, we **architect an approach** that leverages the best trade-off between the current solutions to solve our problems, which is explained in more detail, in Section 4.5.1 (p. 65). Then, we **implement the approach** in a prototype using the framework that most suits our needs, in order to achieve the desired objectives. The methodology for the prototype is explained in more detail in Section 4.5.2 (p. 66).

Lastly, we provide an **evaluation** of the feasibility and benefits of the solution that should provide constructive feedback about the approach and its respective implementation, showing how it could be further improved.

4.5.1 Approach

The goal of this dissertation is to propose a new approach that supports the traceability of health data throughout the transformations applied to it by the processing made by disease treatment research projects.

The approach to solve the problem is in-line with previous work and uses blockchain technology [99]. This will allow decentralized trust over the information stored on it as well as

immutability and tamper resistance. The solution should provide traceability of the data transformations. The verification of that traceability information should find the sweet spot between decentralization and computational effort. In order to achieve such sweet spot, we will provide a solution that has a combination of two of the best mechanisms reviewed in Chapter 3 (p. 23) while also having some improvements to each of those mechanisms alone so it further adapts to our problem. One of the mechanisms will be as decentralized and with the least need of trust as possible whereas the other will provide a computationally faster way of verifying the information with loss of decentralization.

The consensus algorithm will be a modified version of Proof of Stake, explained in Section 2.3.3 (p. 13) which will use reputation as a consensus resource, as it is an important asset for the entities of the system. For more information about consensus algorithms and resources, please see Section 2.3 (p. 11).

For more detailed information about the approach, please see Section 5.3 (p. 68).

4.5.2 Prototype

The approach will be implemented as a prototype, providing a proof of concept of its feasibility. This dissertation is part of a research project, the iReceptorPlus, explained in Section 2.6 (p. 19), which objective is to support the process of sharing health research data and our contribution is focused on providing traceability of the health research data transformations. Thereby, the prototype should be as interoperable as possible in order to be integrated in the future.

4.5.3 Interviews with Experts

In order to validate the approach, we conduct live interviews with experts in the fields of blockchain and health research data sharing, with the majority of them being members of the iReceptorPlus.

Their answers will help evaluate the benefits and the feasibility of the approach as well as the respective prototype. They should also provide constructive feedback that allows to further improve both the approach and respective prototype developed.

Chapter 5

Solution

This chapter describes the proposed solution that aims to solve the problems explained in Chapter 4 (p. 61). The solution is composed by the approach and its implementation in a prototype.

We start by explaining the context where this solution fits, in Section 5.1 (p. 67), followed by the objectives of the solution, in Section 5.2 (p. 67). Then, we explain the approach architected to solve the problems, in Section 5.3 (p. 68). Furthermore, and as a proof of concept, supporting the feasibility of the approach, we proceeded to the implementation of it, using Hyperledger Fabric, as described in Section 5.4 (p. 76).

5.1 Contextualization

This dissertation is part of a research project, the iReceptorPlus, in the context of the process of sharing personal health data, carried out by the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), as explained in Chapter 2, Section 2.6 (p. 19).

While the research project addresses many aspects regarding the health data sharing process, this dissertation is focused in providing traceability of the data transformations that occur in the process of sharing health data, being independent from other research projects and dissertations related to the context. Furthermore, the traceability provides awareness of the data transformations that occur as a result of the research process. As an extra, we implement a rewards system to encourage other entities to audit the traceability data of an entity, through voting rounds, in order to facilitate the process of verifying the information that is stored on the blockchain.

Since the solution leverages the properties of blockchain to provide data traceability and is integrated in the iReceptorPlus research project, the solution is called iReceptorChain.

5.2 Objectives

The main objetive of this dissertation is to improve the process of sharing personal health data for scientific research by providing traceability of the data transformations as well as the ability to determine data provenance. This traceability is expected to help overcome the problems resulting

from having multiple entities with different and competitive interests involved, as explained in Chapter 4 (p. 61) and help building trust in the ecosystem, ultimately leading to more cooperation and, therefore, progress.

Therefore, we aim to implement our approach in a prototype that is able to provide a decentralized registry of traceability data, enabling entities to determine health data provenance and trace the data transformations resulting from scientific research. We also aim to evaluate the feasibility and the benefits of the approach to the context through interviews with health data sharing and blockchain specialists which will be presented the features provided by the prototype.

These objectives where obtained through an analysis of the advantages and disadvantages of the each of solutions investigated in the State of the Art (Chapter 3, p. 23) as well as the current problems in the context, having defined a set of research questions, in Section 4.4 (p. 64).

Thus, by the end of this dissertation, the prototype developed should provide support for registering traceability of the data transformations and have its feasibility and benefits evaluated through interviews with specialists in the fields of sharing health data and blockchain. Achieving the objectives explained, we should be able to respond positively to each research questions presented in Section 4.4 (p. 64), proving the justifying that bases this dissertation.

5.3 High Level Approach

In order to fulfil the objectives explained above, we architect a decentralized registry (the blockchain) that provides the ability for entities to create traceability data entries which can be accessed by other entities. The proposal of this first component is explained below, in Section 5.3.1 (p. 68).

Also, and as an extra, we also architect a rewards system to encourage other entities to audit the traceability data entries of an entity in an attempt to further improve the system by making it easier to validate the information registered on the blockchain. The description of the architecture of this second component is explained below, in Section 5.3.2 (p. 70).

5.3.1 Traceability Data Registry

Starting by explaining the proposal for the decentralized registry of traceability data.

The data stored should be the same for every node in the network through consensus that should be achieved automatically by the entire network.

In order to simplify the process, the traceability data is represented in the form of entries, the unit of traceability data. Data lineage can be achieved by linking different entries. Each entry of traceability data consists of the following:

- Input dataset *
- Output dataset *
- Processing steps details
  - Executable program used *

- Version of the program
- Configuration parameters

In order to represent the datasets we considered two options: storing the entire datasets on the blockchain, in an approach similar to BigChainDB [12, 67], or store a unique representation of them along with a pointer to the repository where they are stored so they can later be retrieved. Since Electronic Health Records are large files (raw sequences of data can be as big as 4GB), we immediately ruled out the first possibility, otherwise all the full nodes of the entire network would have to store all datasets, even the ones they are not interested in, making it unpractical to run a full node. We followed the same approach for the executable program that processes the files, since although the files are not nearly as large, they would be repeated many times on the blockchain since the same program could be used to process the health data. Thus, the components marked with a * are represented by a hash value (cf. Section 2.2.3, p. 8) and pointer to repository where they are stored.

Input dataset consists of a representation of the EHR that was processed, meaning the one that was given as input to the processing software.

Output dataset consists of a representation of the resulting EHR, meaning the one that was returned as an output by the processing software.

Processing steps details consist of a representation of the software that was used to make the processing, having multiple components:

Executable program used consists of a representation of the binary executable of the software that was executed. This is useful in order to have confidence (upon checking the veracity of the registered traceability data) that the executable is the same as the claimed to be used when the traceability data entry was registered. By keeping a unique representation of it on the blockchain, the verifier can be certain that he is using the correct executable as well as verify the integrity of the executable downloaded from the repository.

Version of the program consists of the version of the program. This is useful because, due to updates, there may be differences to the outputs of previously existing algorithm to the same input dataset.

Configuration parameters consists of the configuration parameters (arguments) given to the processing software. Since different configuration parameters can produce a different output for the same input dataset, they are required in order to verify the information traceability data entry.

By associating the **Input dataset** to another entry's **Output dataset**, we are able to trace the data transformations to the previous level of processing. By recursively repeating the process, we

are able to provide data lineage and determine the provenance of the data when we reach level 0 data (raw sequence data). Upon reaching the level 0 data, the **Input dataset** is not linked to another **Output dataset**.

Similarly, by associating the **Output dataset** to another entry's **Input dataset**, we are able to trace the data transformations to the next level of processing, if exists. By recursively repeating the process, we are able to further trace front the data transformations.

This traceability data decentralized registry already provides a solution to some of the problems by enabling entities to register and view other entities' traceability data aiming to support the process of building trust and encourage cooperation.

5.3.2 Traceability Data Auditing and Incentive System

Finally, and as an extra, we aim to provide a way for entities to be able to audit the traceability data (say whether it's valid or not) in order to further support the process of building trust by keeping all entities engaged on it in a cooperative way while also providing a computationally easier method to verify the information stored on the blockchain.

Since health data transformations are computationally expensive tasks, support for auditing others' traceability data without an incentive for it, wouldn't be enough to achieve the desired goal. Therefore, an incentive system is required to reward other entities that audit an entity's traceability data, so they are encouraged to perform such computationally expensive tasks.

In Section 5.3.2.1 (p. 70) we present a description of the structure of the Incentive System used to encourage entities to audit each other's traceability data. Then, in Section 5.3.2.2 (p. 71), we provide a description of the possible actions for entities that are part of the traceability data auditing system.

5.3.2.1 Incentive System

All blockchain incentive mechanisms require some resource that is rewarded upon correct behavior, as explained in Section 2.3.3 (p. 13). In some mechanisms, it might be necessary to remove that same resource in case of incorrect behavior and stake it when performing some tasks that involve waiting for the consensus decision in order to grant byzantine fault tolerance [31]. An example of a mechanism that requires rewards, penalties and stakes is proof of stake, explained in Section 2.3.3 (p. 13), in which our incentive system is inspired. In the case of cryptocurrencies like Bitcoin, explained in Section 2.3.2 (p. 12), the rewards are always in money, since the blockchain manages the circulation of a currency. In our case, and since there is no currency involved, money isn't available as an incentive resource that can be represented and managed by the blockchain network.

Therefore, there was the need of finding another incentive resource. We started by looking into what was important for the entities of the ecosystem of health data that was also representable and manageable by the blockchain. According to these desiderata, we considered using reputation as an incentive resource. Reputation is important for the entities in the given context since they

Action	Fee	Stake	Reward	Penalty
Create traceability data entry	Yes	Yes	No	Yes
Up vote traceability data entry	No	Yes	Yes	Yes
Down vote traceability data entry	No	Yes	Yes	Yes

Table 5.1: Appliance of incentive system's concepts to the possible actions of the auditing system

are well identified and cannot generate new identities on demand. The reputation idea also fits our approach of building trust since the more reputation an entity has on the system, the more other entities are willing to trust it. Therefore, there is an inherent incentive for entities to accumulate as much reputation as possible. Being a resource that is important for entities to keep and gain as much as possible, we are confident that it works well as an incentive resource.

In order to ensure the health of the ecosystem and incentivize correct behavior, there are four main key concepts of the incentive system:

- **Fee** is required when an entity is requesting working from the network.
- **Stake** is required while final decision (on whether the behavior was correct or not) has not yet been made, avoiding accumulation of incorrect behavior.
- **Reward** is performed upon correct behavior.
- **Penalty** is issued upon incorrect behavior.

The process of charging a fee involves removing the amount of the fee from the reputation of the entity.

The process of staking involves moving some of the reputation of the entity to a different counter, the staked reputation, to ensure that the entity cannot use that reputation to perform any more tasks while the network has not decided yet if the behavior was correct or not. This avoids the entity to compromise the entire system by introducing noise with incorrect behavior, being only able to compromise some part of the system - until all its reputation is at stake.

The process of rewarding is rather simple: add reputation to an entity. The process of penalizing is also simple: subtract reputation from an entity.

5.3.2.2 Traceability Data Auditing System

Table 5.1 provides an overview of the appliance of the key concepts of the incentive system, described above, in Section 5.3.2.1 (p. 70), to the actions that entities can perform on the iReceptorChain system.

The meaning of each column is as follows:

- **Fee** tells whether the entity is required to pay a fee in order to perform the action.
- **Stake** tells whether the entity is required to place a stake in order to perform the action. This is used for potentially dangerous actions, in which the entity claims something that could be wrong (either intentionally or unintentionally).

Action	Approved	Rejected
Create traceability data entry	Receives Stake	Looses Stake
Up vote traceability data entry	Receives Stake + Reward	Looses Stake
Down vote traceability data entry	Receives Stake + Reward	Looses Stake

Table 5.2: Possible outcomes for each action of the auditing system

- **Reward** tells whether the action can generate a reward for the entity that performs it.
- **Penalty** tells whether the action can issue a penalty for the entity that performs it.

Table 5.2 provides an overview of the possible outcomes for each action, in order to incentivize correct behavior.

The meaning of each column is as follows:

- **Approved** specifies the action(s) taken by the system in case the action of the entity is approved by network consensus.
- **Rejected** specifies the action(s) taken by the system in case the action of the entity is rejected by network consensus.

Now, we describe the rationale used to architect the concepts described above.

Create traceability data entry Firstly, we have the creation of a traceability data entry that comes already from the basic part of our system, the traceability data decentralized registry, explained in Section 5.3.1 (p. 68). We have further modified the actions taken by the system upon this action to add support for the incentive system. At the moment of creation of an traceability data entry, the requirements are the following:

- **Fee** We decided that this action should require its author paying a fee, since he is requesting the network to audit the traceability data.
- **Stake** We have also made required to place a stake upon creating a traceability, because it is considered a potentially dangerous action, since the entity could be introducing incorrect information on the network.

When the final network consensus decision has been made on whether to approve or reject the traceability data, the outcomes are the following:

In case the traceability data was approved:

- **UnStake** The creator receives back the stake he has placed to create the traceability data entry.

In case the traceability data was rejected:

- **Penalty** The creator receives a penalty which can be higher or equal to the amount he placed as stake in the beginning. In case the amount is equal, the appliance of the penalty can be seen as not receiving back the stake. If the penalty is higher than the amount he placed as stake in the beginning, the appliance of the penalty can be seen as not receiving back the stake and getting its reputation substracted the difference between the penalty and the stake.

Up vote or down vote traceability data entry The requirements and possible outcomes for up or down voting a traceability data entry are very similar, so we describe them together.

A correct vote can be either approving traceability data that is decided by network consensus to be correct or rejecting traceability data that is decided by network consensus to be incorrect. An incorrect vote can be either approving traceability data that is decided by network consensus to be incorrect or rejecting traceability data that is decided by network consensus to be correct.

At the moment of voting for a traceability data entry, the requirements are the following:

- **Stake** A Stake is required upon voting for a traceability data entry, because it is considered a potentially dangerous action, since the entity could be placing a wrong vote (either intentionally or unintentionally). Voting round means up voting something incorrect or down voting something correct.

When the final network consensus decision has been made on whether to approve of reject the vote, the outcomes are the following:

In case the vote is decided to be correct:

- **UnStake** The voter receives back the stake he has placed to create the traceability data entry.
- **Reward** The voter receives a reward since his vote is considered correct by network consensus.

In case the vote is decided to be incorrect:

- **Penalty** The voter receives a penalty which can be higher or equal to the amount he placed as stake in the beginning. In case the amount is equal, the appliance of the penalty can be seen as not receiving back the stake. If the penalty is higher than the amount he placed as stake in the beginning, the appliance of the penalty can be seen as not receiving back the stake and getting its reputation substracted the difference between the penalty and the stake.

The process of auditing the traceability data is based on voting rounds, in an approach similar to the DataProv system, explained in Section 3.2 (p. 29) but with further adaptions and improvements to solve our specific problem. There are two important possible user actions on the system: creating a traceability data entry and voting (either up or down) for a traceability data entry.

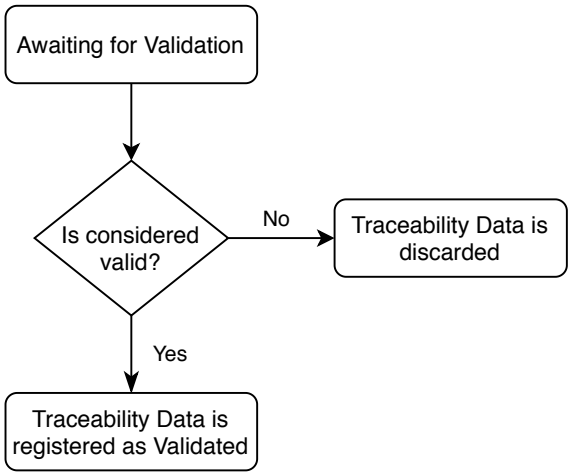

Figure 5.1: Traceability data life cycle

The life cycle of the traceability data can be seen on Figure 5.1. When it is created, it goes to the "Awaiting Validation" state and the voting round begins. At the end of the voting round, if it is considered valid, it is registered as valid. In case it is considered invalid, it is discarded, which means it is made less accessible but still can be seen on the log of transactions of the blockchain, since nothing is really discarded on the blockchain. Therefore, the traceability data possible states are the following:

- Awaiting Validation
- Validated
- Discarded

The Figure 5.2 represents an iteration of the voting round. First, a Voter submits a vote to the java Vote Contract (running on the blockchain). The contract verifies if the round has finished, meaning if there is a considerable ratio between up and down votes and the minimum threshold of votes has been met. For example, if the ratio is 100% but there is only one vote, the round shouldn't be terminated yet since the number of votes is not enough to ensure network consensus. The values of the necessary ratio and the minimum threshold of votes are customizable, in order to better adapt to the size and nature of the network. In case the conditions to terminate the round have been met, the contract checks if the traceability data should be accepted or rejected with each decision involving different outcomes for the entities that have participated on the voting round.

In case the traceability data is accepted, the contract takes the following actions:

Reward Approvers Since it was decided by network consensus that they were telling the truth.

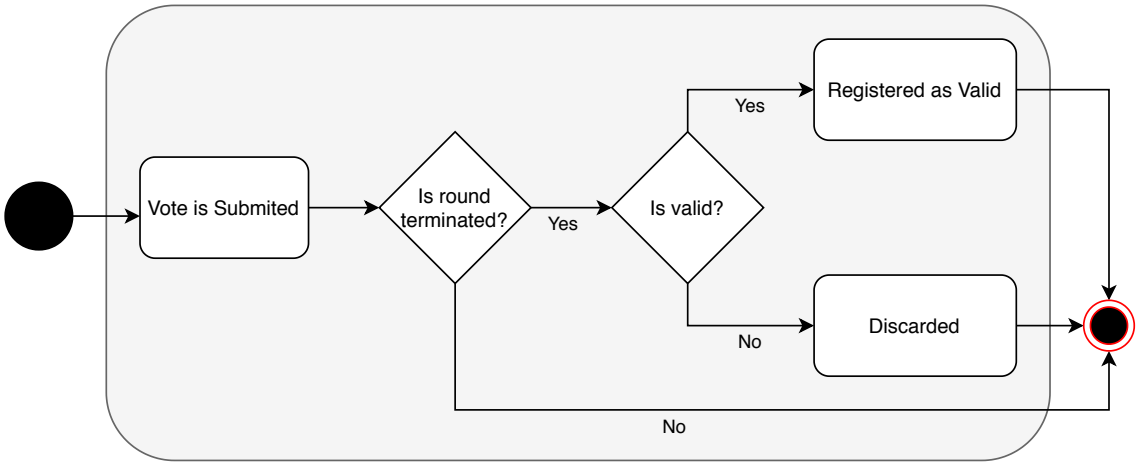

Figure 5.2: Voting round activity diagram

Penalize Rejecters Since it was decided by network consensus that they were not telling the truth (either intentionally or unintentionally).

In case the traceability data is rejected, the contract takes the following actions:

Penalize Creator Since it was decided by network consensus that they were not telling the truth by introducing traceability data that is considered to be incorrect (either intentionally or unintentionally).

Penalize Approvers Since it was decided by network consensus that they were not telling the truth (either intentionally or unintentionally).

Reward Rejecters Since it was decided by network consensus that they were not telling the truth.

It is important to note that the approach is highly customizable to the nature of the network and context of the problem with the values of rewards, stakes and penalties being easily set by the user. There is even support for penalties being different (higher or less) than the stake amount, unlike most proof of stake systems. We assumed that the process of creating the traceability data shouldn't receive a reward but instead pay a fee since it is requesting work from the network to validate it. It also works as a way of spending reputation, ensuring there is a use for it inside the system. However, this can easily be transformed into a reward (and no fee) by the user, if that suits more the context (entities are already incentivized enough to perform validations).

5.4 Implementation

In order to implement the approach described in Section 5.3 (p. 68), we first conducted the frameworks analysis, presented in Section 3.6 (p. 54). In this sense, we have determined that Hyperledger Fabric, reviewed in Section 3.6.2 (p. 56), is the best choice to be used a blockchain framework for creating applications. Furthermore, we have also established that Java should be the language of choice to implement the smart contracts that run in the nodes of the blockchain, under the consensus of the network, as explained in Section 3.6.4 (p. 58).

5.4.1 Workarounds to Fabric's Limitations

The support for smart contracts written general purpose programming languages like Java, is very convenient to the programmer in the sense that it abstracts the P2P network logic as well as some of the blockchain logic. This abstraction allows the programmer to concentrate more on the business logic of the specific domain of the application, enabling us to architect the implementation almost like any other Java application. Yet, there are important limitations to this, enforcing the consideration of the aspects below:

Mandatory use of fully deterministic code Due to the fact that the code will be run on multiple nodes of the network (to ensure decentralization), it is crucial that the whole computation process is deterministic. This is an important restriction, since it denies the use of randomly generated values. It is important to note that experiencies based on pseudo random values can still be run, as long as all the nodes feed the PRNG with the same seed, causing it to generate the same random sequence of numbers. Pseudo random values generated based on time are to be avoided, since different nodes may and most likely will run the computation at different times.

Feature-limited database Like many other applications, blockchain applications also need to store information. In this sense, the Hyperledger Fabric framework provides a database for that same purpose. Since, the application code will not only run on a single machine (as with the Client-Server Architecture, explained in Section 2.1.1 (p. 5)), but instead must be compatible with all the nodes of the network, so needs to be the database. Moreover, and inherently to the definition of network consensus, it is also necessary that all nodes store the same information in their databases, in order to ensure consistency of data upon querying and determinism on running the smart contract code, since the databases provide inputs to the computation.

In this sense, the database offered by the framework is rather limited, providing only storage of strings, organized as key-value pairs. Therefore, it is very useful to use JSON encoding for storing the information in the database, since it allows to convert any object, with a set of properties, and even nested objects to a string. The result of the JSON encoding would be stored as value and a smaller representative of the object would be used as a key, as recommended by the examples

of the Hyperledger Fabric documentation [1]. The objects stored in the database (the value of the key-value pair) are called *assets*, as explained in detail in Section 3.6.2 (p. 56).

Taking into account the second aspect, feature-limited database, we have started by developing some work arounds to enable the abstraction of the database logic from our code. With our abstraction, we made CRUD operations easier to perform. Since the database offered by Hyperledger Fabric lacks the concept of tables, that all database systems encompass, we have created an abstraction that allows our code to be written as if there were database tables. Another difference to common database systems is the need of creating a key for each element stored. We have architected an approach that solves both problems at the same time, by making the desired table name a part of the asset's key. Since one table can contain multiple entries, we have also included in the key an unique identifier for the asset.

In this sense, if, for example, we wanted to store Car1 and Car2 in table Car and Alice and Bob in table Person, and we have defined the ids c1, c2, p1 and p2, the corresponding Hyperledger Fabric key-value pair database structure would be:

- "Car-c1" -> Car1
- "Car-c2" -> Car2
- "Person-p1" -> Alice
- "Person-p2" -> Bob

In order to implement this approach, we have created a Java class, called *HyperledgerFabricBlockhainRepositoryAPI*, that implements this logic. It's constructor receives the name of the table that is intended to be abstracted and implements methods for all CRUD operations. It was possible to create an instance of this class wherever desired, but we decided to create derived classes, one for each table that is intended to be abstracted. The derived classes call the constructor of the main class with the appropriate table name and also implement the CRUD methods since they are inherited (without overriding them). The super class, as well as the derived classes that implement specific table abstractions are bundled into a package called *FabricBlockchainRepositoryAPIs*.

Taking into account the first aspect, mandatory use of fully deterministic code, these unique ids need to be generated by the application that calls the smart contracts, since if they were generated randomly, as UUID's, the network would never reach consensus because all the nodes would, with almost 100% probability, generate different UUID's.

5.4.2 Data Classes

After architecting workarounds for Hyperledger Fabric's main limitations, we started defining the classes that represent the assets that we want to store on the blockchain. These classes will be

[1] Fabric's documentation can be consulted at https://hyperledger-fabric.readthedocs.io/en/release-2.0/whatis.html

serialized to a string, using JSON encoding by the repository API classes, described above. The CRUD methods of these classes receive instances of the objects described in this section, serialize them to strings and then store them in the database. The paragraphs below describe each of the main classes used to define the properties of assets stored on the blockchain.

Entity This class represents an entity and is needed to store information about the entity on the blockchain such as its reputation. The properties of the class are:

id A String, representing an unique identifier of the entity.

reputation A Long, representing the currently available reputation of the entity.

reputationAtStake A Long, he reputation of the entity that is current at stake, due to having performed potentially dangerous actions.

TraceabilityData This class represents an entry of traceability data and contains the necessary information to describe a processing step. The properties of the class are:

inputDatasetHashValue A string, representing the hash value of the input data set, the one that was given to the processing algorithm.

outputDatasetHashValue A string, representing the hash value of the output data set, the result of the processing step.

processingDetails An instance of class *ProcessingDetails*, representing the steps taken to process the input dataset and produce the output dataset as result. The properties of this class are described below.

creatorID A string, representing the ID of the entity that created that requested the creation of the traceability data entry.

approvers An ArrayList, containing the IDs of the entities that have approved the traceability data entry when it was awaiting validation, in a voting round.

rejecters An ArrayList, containing the IDs of the entities that have rejected the traceability data entry when it was awaiting validation, in a voting round.

ProcessingDetails This class represents the steps taken to make the data transformation. This class contains the necessary information to reproduce the data transformation, allowing anyone with the input dataset to produce the output dataset. The properties of the class are:

softwareId A string, representing an unique ID of the software used to perform the data transformation.

softwareVersion A string, representing the version of the software used to perform the data transformation.

softwareBinaryExecutableHashValue A string, representing the hash value of the executable program used to perform the data transformation. This is used for integrity verification, as described in Section 5.3.1 (p. 68).

softwareConfigParams A string, representing the configuration parameters of the software used to perform the data transformation. This should be the necessary command line arguments, that need to be passed to the executable binary file, in order to replicate the data transformation.

5.4.3 Voting Rounds

In order to implement the business logic that manages the voting rounds, we decided to design the solution as a state machine, using the State design pattern. We have bundled the classes to implement the voting rounds in the package *VotingRoundStateMachine*.

Firstly, we created the class *TraceabilityDataStateMachine* that has two methods, one to register an approval (up vote) and the other a rejection (down vote) to a given traceability data entry. This class implements the logic for creating a traceability data entry, checking if the creator has enough reputation and staking the creator's reputation.

Then, we created the abstract class *State* that is called by this class and should be instantiated as one of the derived classes to implement specific behavior for each of the states of traceability data. The derived classes are *AwaitingValidation* and *Validated* which implement the logic for traceability data that is awaiting validation and validated, respectively. The class *AwaitingValidation* implements the more complicated logic in which we now delve.

Both methods of the methods that register a vote, approval and rejection, check if the voter has enough reputation to place the vote, stake his reputation and use the logic implemented in *HyperledgerFabricBlockhainRepositoryAPI* to access the database and register the vote in it. Then, they check if the condition for terminating the voting rounds has been met, which is a combination of enough vote ratio and enough number of votes, as explained in Section 5.3.2 (p. 70). In case the condition to finish the round has been met, the methods delegate the round finishing logic in class *RoundFinisher*.

In this sense, this class has two methods: one to invalidate and the other to validate the traceability data. Both methods start by unstaking the creator's and the voter's reputation, using the following method:

```
1   private void unStakeCreatorAndVotersReputation(TraceabilityData
        traceabilityData, EntityReputationManager manager)
2   {
3       Long unStakeForCreator = ChaincodeConfigs.stakeForCreatingEntry.get();
4       Long unStakeForApprovers = ChaincodeConfigs.stakeForUpVotingEntry.get();
5       Long unStakeForRejecters = ChaincodeConfigs.stakeForDownVotingEntry.get();
6
7       manager.unstakeReputation(traceabilityData.getCreatorID(),
            unStakeForCreator);
8       manager.unstakeReputation(traceabilityData.getApprovers(),
            unStakeForApprovers);
9       manager.unstakeReputation(traceabilityData.getRejecters(),
            unStakeForRejecters);
10  }
```

Listing 5.1: Method to unstake the creator's and the voter's reputation upon voting round termination.

Then, the methods take different actions. If the traceability data is considered correct by network consensus, the method rewards the approvers and penalizes the rejecters, using the following code:

```
1   Long rewardForApprovers = ChaincodeConfigs.rewardForUpVotingCorrectEntry.get();
2   Long penaltyForRejecters =
        ChaincodeConfigs.penaltyForDownVotingCorrectEntry.get();
3
4   manager.rewardEntities(traceabilityData.getApprovers(), rewardForApprovers);
5   manager.penalizeEntities(traceabilityData.getRejecters(), penaltyForRejecters);
```

Listing 5.2: Method to reward approvers and penalize rejecters when traceability data is considered correct.

If the traceability data is considered incorrect by network consensus, the method penalizes the creator as well as the approvers, and rewards the rejecters, using the following code:

```
1   Long penaltyForCreator =
        ChaincodeConfigs.penaltyForCreatingIncorrectEntry.get();
2   Long penaltyForApprovers =
        ChaincodeConfigs.penaltyForUpVotingIncorrectEntry.get();
3   Long rewardForRejecters =
        ChaincodeConfigs.rewardForDownVotingIncorrectEntry.get();
4
5   manager.penalizeEntity(traceabilityData.getCreatorID(), penaltyForCreator);
```

```
6     manager.penalizeEntities(traceabilityData.getApprovers(), penaltyForApprovers);
7     manager.rewardEntities(traceabilityData.getRejecters(), rewardForRejecters);
```

Listing 5.3: Method to penalize creator and approvers, and reward rejecters when traceability data is considered incorrect.

These are the most important details regarding the implementation of the voting rounds.

5.4.4 Transactions

As explained in Section 3.6.2 (p. 56), a *transaction* is defined as a method that contains the code that runs on the nodes of the blockchain upon invocation by external applications, similarly to the endpoints of a REST API. These *transactions* are the entry points for triggering the execution of chaincode in the blockchain, thus being the highest code level which then calls the code defined in other logic packages, described above. We now present some of the most important *transactions*, defined on the main class: *iReceptorChain*.

Create traceability data entry This transactions allows the creation of a traceability data entry. The method receives all the information of the traceability data entry to create as parameters. It uses the class *TraceabilityDataAwaitingValidationRepositoryAPI*, a subclass of *HyperledgerFabricBlockhainRepositoryAPI*, in order to perform the CRUD database operations necessary for the process. The code of the transaction is the following:

```
1     @Transaction()
2     public TraceabilityDataAwaitingValidationReturnType
          createTraceabilityDataEntry(final Context ctx, final String newUUID, final
          String inputDatasetHashValue final String outputDatasetHashValue, final
          String softwareId, final String softwareVersion, final String
          softwareBinaryExecutableHashValue, final String softwareConfigParams)
3     {
4         ChaincodeStub stub = ctx.getStub();
5
6         TraceabilityDataAwaitingValidation traceabilityData = new
              TraceabilityDataAwaitingValidation(inputDatasetHashValue,
              outputDatasetHashValue,
7                 new ProcessingDetails(softwareId, softwareVersion,
                      softwareBinaryExecutableHashValue, softwareConfigParams), new
                      EntityID(ctx.getClientIdentity().getId()));
8
9         TraceabilityDataInfo dataInfo = new TraceabilityDataInfo(newUUID,
              traceabilityData);
10        HyperledgerFabricBlockhainRepositoryAPI api = new
              TraceabilityDataAwaitingValidationRepositoryAPI(ctx);
11
```

```
12          TraceabilityDataStateMachine stateMachine = new
                TraceabilityDataStateMachine(dataInfo, api);
13          stateMachine.initVotingRound(new
                EntityID(ctx.getClientIdentity().getId()));
14
15          TraceabilityDataAwaitingValidationReturnType traceabilityDataInfo = new
                TraceabilityDataAwaitingValidationReturnType(newUUID,
                traceabilityData);
16          return traceabilityDataInfo;
17      }
```

Listing 5.4: Chaincode transaction to create traceability data entry.

Up vote traceability data entry This transactions allows an entity to register an up vote for a traceability data entry. The method receives the information of the traceability data entry to be up-voted, as parameters.

Since the first necessary actions are similar to the ones of the process of down voting, both transactions use method *getTraceabilityDataFromDBAndBuildVotingStateMachine* to perform the actions that are identical. The code of this method is be presented below, after the explanation of the transaction for down voting.

The actions necessary after the common part are rather simple, involving only calling method *upVote* of class *TraceabilityDataStateMachine*. This method performs the appropriate actions, depending on the state of the traceability data, as explained above, in Section 5.4.3 (p. 79).

The code of the transaction is the following:

```
1       @Transaction()
2       public String registerUpVoteForTraceabilityEntryInVotingRound(final Context
            ctx, final String uuid)
3       {
4           TraceabilityDataStateMachine stateMachine =
                getTraceabilityDataFromDBAndBuildVotingStateMachine(ctx, uuid);
5           VotingStateMachineReturn stateMachineReturn =
                stateMachine.upVote(getEntityIdFromContext(ctx));
6
7           return stateMachineReturn.getMessage();
8       }
```

Listing 5.5: Chaincode transaction to register up vote for traceability data

Down vote traceability data entry This transactions allows an entity to register a down vote for a traceability data entry. The method receives the information of the traceability data entry to be down-voted, as parameters.

This transaction also uses method *getTraceabilityDataFromDBAndBuildVotingStateMachine*, since the first necessary actions are similar to the ones of the process of up voting to perform the actions that are identical.

The actions necessary after the common part are rather simple, involving only calling method *downVote* of class *TraceabilityDataStateMachine*. This method performs the appropriate actions, depending on the state of the traceability data, as explained above, in Section 5.4.3 (p. 79).

The code of the transaction is the following:

```
1    @Transaction()
2    public String registerDownVoteForTraceabilityEntryInVotingRound(final Context
         ctx, final String uuid)
3    {
4        TraceabilityDataStateMachine stateMachine =
             getTraceabilityDataFromDBAndBuildVotingStateMachine(ctx, uuid);
5        VotingStateMachineReturn stateMachineReturn =
             stateMachine.downVote(getEntityIdFromContext(ctx));
6
7        return stateMachineReturn.getMessage();
8    }
```

Listing 5.6: Chaincode transaction to register down vote for traceability data

Start of voting transactions Both up and down voting transactions call method *getTraceabilityDataFromDBAndBuildVotingStateMachine* since the first actions for both are identical. This method uses class *TraceabilityDataAwaitingValidationRepositoryAPI*, a subclass of *HyperledgerFabricBlockhainRepositoryAPI*, in order to perform the CRUD database operations necessary for the process. After performing some business logic verifications, it builds an instance of class *TraceabilityDataStateMachine* and returns it so that the transactions for up and down voting can then use it to call the appropriate method. The code for the method is the following:

```
1    private TraceabilityDataStateMachine
         getTraceabilityDataFromDBAndBuildVotingStateMachine(Context ctx, String
         uuid)
2    {
3        ChaincodeStub stub = ctx.getStub();
4
5        TraceabilityDataAwaitingValidationRepositoryAPI api = new
             TraceabilityDataAwaitingValidationRepositoryAPI(ctx);
6        TraceabilityData traceabilityData = (TraceabilityData) api.read(uuid);
7
8        String voterID = ctx.getClientIdentity().getId();
```

```
9         String creatorID = traceabilityData.getCreatorID().getId();
10        if (voterID.equals(creatorID))
11            throw new ChaincodeException("Creator of traceability data cannot vote
                  for it.");
12
13        TraceabilityDataInfo traceabilityDataInfo = new TraceabilityDataInfo(uuid,
              traceabilityData);
14        TraceabilityDataStateMachine traceabilityDataStateMachine = new
              TraceabilityDataStateMachine(traceabilityDataInfo, api);
15
16        return traceabilityDataStateMachine;
17    }
```

Listing 5.7: Method to perform initial actions for registering a vote.

5.4.5 Testing

In order to assure code correctness, we have developed unit tests to all the features implemented. Since the *transactions*' code uses the database as input for processing, mocking the database became necessary to perform entire feature testing. Hyperledger Fabric provides mechanisms that allow us to perform mocking of data, providing database inputs to our code. This requires to setup a testing environment, using the mock classes provided by Hyperledger Fabric, as well as some common methods to initialize the database contents appropriately for the testing methods. We provide coverage statistics for several components below.

Global The global coverage statistics are:

- 86% Class coverage (37/43)
- 87% Method coverage (139/154)
- 89% Line coverage (521/581)

iReceptorChain, main class The coverage statistics for the main class, *iReceptorChain*, which contains the declarations of *transactions*, are the following:

- 100% Method coverage (11/11)
- 91% Line coverage (108/118)

Data classes The coverage statistics for the package of the data classes, described in Section 5.4.2 (p. 77), are the following:

- 100% Class coverage (6/6)
- 100% Method coverage (34/34)

Element	Class, %	Method, %	Line, %
EntityData	100% (1/1)	100% (6/6)	100% (19/19)
EntityID	100% (1/1)	100% (3/3)	100% (8/8)
ProcessingDetails	100% (1/1)	100% (6/6)	100% (17/17)
TraceabilityData	100% (1/1)	100% (11/11)	100% (31/31)
TraceabilityDataAwaitingValidation	100% (1/1)	100% (4/4)	100% (10/10)
TraceabilityDataValidated	100% (1/1)	100% (4/4)	100% (7/7)

Figure 5.3: Coverage for data classes package

- 100% Line coverage (92/92)

The detailed statistics for the coverage of this package, including coverage for every class, can be seen on Figure 5.3.

Voting Rounds The coverage statistics for the package that implements the logic for the voting rounds, described in Section 5.4.3 (p. 79), are the following:

- 66% Class coverage (12/18)
- 79% Method coverage (38/48)
- 79% Line coverage (170/213)

The lack of class coverage is due to the existence of exception classes, which represent errors that are not reproducible by invoking code at the *transaction* level. The exceptions are thrown by verifications on the lower level classes, that are there to ensure that the application does not crash but instead communicates the problem to the upper level code. The upper level code then parses the errors, and converts them into other exceptions that are seen by the *transaction* level. These verifications would only be triggered in case of cascading errors, causing the information on the database to be corrupted, which we don't simulate in our testing environments.

The detailed statistics for the coverage of this package, including coverage for every class, can be seen on Figure 5.4.

Element	Class, %	Method, %	Line, %
Exceptions	54% (6/11)	66% (12/18)	64% (22/34)
Returns	100% (1/1)	66% (2/3)	83% (5/6)
States	75% (3/4)	78% (11/14)	77% (87/112)
EntityReputationManager	100% (1/1)	100% (9/9)	95% (39/41)
TraceabilityDataStateMachine	100% (1/1)	100% (4/4)	85% (17/20)

Figure 5.4: Coverage for voting rounds package

Chapter 6

Expert Opinion

This chapter provides an evaluation of the feasibility and the benefits of our analysis of the fields as well as the approach and respective implementation, presented in Chapter 5 (p. 67). The evaluation is based on the opinion of experts in the fields of blockchain and health data research.

The sections below explain the process of validating our main contributions. Firstly, in Section 6.1 (p. 87) we explain the objectives of these interviews and the contributions being evaluated with them. Then, on Section 6.2 (p. 88), we explain how we have prepared these interviews in order to achieve the desired objectives explained in the previous section. We cover aspects such as the elaboration of questions, the choice of the type of interview and we present the questions that were asked to the experts. Finally, in Section 6.3 (p. 93), we provide an analysis of the results, based on statistics of the answers as well as some important detailed answers that help evaluating the contributions developed. We also present some improvement suggestions made by the experts that could help further making the solution more suitable to our context.

6.1 Objectives

This section describes the objectives of the interviews with experts. The objectives with the interviews are to evaluate our main contributions.

The first objective of this dissertation is to explore the current practices when managing health research data in order to assess the existing issues in the system so that we we can propose new solutions to address them. As a result of this, we produced our first contribution which is an analysis of these issues. The first set of questions, explained below, in Section 6.2.2 (p. 88), aims to evaluate our first our understanding the current issues with the current system of health research data.

Furthermore, it is also our objective to evaluate the current solutions for solving the problems that we have identified and explain the advantages and disadvantages of each, described in Chapter 3 (p. 23). At the end of the analysis of the state of the art, we compare the solutions and establish the best trade-off between the mechanisms of each, in Section 3.5.1 (p. 48). Providing this best trade-off becomes one of our objectives while implementing the solution, described in Chapter 5

(p. 67). Since the contribution produced by the analysis of the state of the art is solution objective, it is assessed together with the evaluation of the feasibility and the benefits of the solution.

Moreover, the next objective of this dissertation is to leverage the properties of blockchain to create a solution to the problems with the current system of health research data that combines the best of the solutions reviewed in the state of the art analysis. This objective produces our second most important contribution which is a high level approach and respective implementation in a prototype, that aims to achieve the desired goal. The interviews with the experts aim to evaluate the success of this contribution in achieving the desired goal. This leads us to last set of questions, where we evaluate the feasibility and the benefits of the approach, as well as the combination of the best between the multiple solutions of the state of the art (which is the idea of the approach that is implemented in the prototype). The questions are described below, in Section 6.2.2 (p. 88).

Lastly, it is also our objective with the interviews to produce our last contribution which is an evaluation of the feasibility and benefits of the solution based on experts' opinions that should provide constructive feedback about the approach and its respective implementation, showing how it could be further improved.

6.2 Preparation

This section describes the preparation of the interviews with the experts. The first step of the preparation phase was picking the experts to be interviewed and is described in Section 6.2.1 (p. 88). The second step of the preparation phase was planning on how and which questions would be asked to the expert, described in Section 6.2.2 (p. 88).

6.2.1 Expert Selection

The experts were selected based on their experience in the fields of health data and blockchain. The expertise of each is listed below:

- Lecturer and researcher on the fields of health data, software engineering and blockchain.
- Technical Manager of iReceptorPlus and researcher on fields of health data.
- Member of the iReceptorPlus, computational biologist at a medical centre and researcher on the field of health data.
- Member of the iReceptorPlus and researcher on the field of health data.

6.2.2 Interview Questions

Since one of our objectives was to explore the fields of health data research in order to understand the current problems in the ecosystem, we prepared an initial set of questions that evaluate our understanding of the current issues. This first set of questions has two subsets that are explained in detail below, each aiming to evaluate different aspects.

- **Conflicting Interests and Competition (CI)** This subset aims to evaluate if there are conflicting interests and competition between the entities in the ecosystem and what are the consequences of it.

- **Process of Building Trust (BT)** This subset aims to evaluate the lack of knowledge about the processing procedures between the entities of the ecosystem. We also evaluate the impact of each of these sets of problems (competition and lack of knowledge of the processing procedures) in cooperation between the entities. And also how the impact in cooperation impacts the progress they make in scientific research. We aim to solve this problems by providing a mechanism where the entities would build trust towards each other.

It was also our objective to investigate the current solutions to solve these problems and identify the advantages and disadvantages of each, establishing the best trade-off between them. Similarly, we have taken that into account in the process of elaborating the interview questions, so that the answers could also provide an evaluation of our state of the art analysis. The questions to evaluate our analysis of the current solutions are bundled together with the set of questions that evaluate the feasibility and the benefits of the approach.

These first considerations on elaborating the questions provide an evaluation for our first contribution which consists of exploring the fields of health data, establishing the best trade-off between the state of the art solutions as well as point out some problems with these solutions, as explained in Section 4.5 (p. 65).

Furthermore, it was also our objective to evaluate the feasibility and the benefits of the solution developed, explained in Chapter 5 (p. 67), so we have prepared a second set of questions in order to provide an evaluation, based on expert opinion. This second set of questions has three subsets that are explained in detail below, each aiming to evaluate different aspects.

- **Feasibility and Benefits of the Approach (FB)** This subset aims to evaluate the feasibility and the benefits of the features provided by the solution. Firstly, it evaluates if the blockchain could provide a feasible mechanism of building trust. Secondly, if the mechanism of the voting rounds further improves solving the problems. And finally, if we were able to establish the best trade-off between the solutions investigated in the state of the art, explained in Section 3.5.1 (p. 48), and develop a solution that joins the best of both worlds in terms of mechanisms while also fixing some problems with each of the mechanisms alone.

- **Benefits of the Solution for Patients (PT)** This subset aims to evaluate the benefits of the solution in patients. Although the solution does not provide them with any feature, they also play a role in the ecosystem since all health data must come from patients. Therefore, they can be indirectly impacted by the changes to the process of research, driven by the features provided by the solution to the research projects.

- **Rewards system (RS)** This subset evaluates the feasibility of the rewarding system. Blockchain requires a consensus algorithm so that the network can reach consensus without the need of trust in a single authority, as explained in Section 2.3 (p. 11). The consensus algorithm

requires a rewards system so that entities are encouraged to perform the validation. The system should also provide penalties for dishonest validators so that entities are discouraged from being dishonest in the sense that they would loose more by being dishonest than if they didn't do anything. The rewarding system should also make sure that the entities prefer to perform the validation then not do anything. This requires establishing a balance that naturally incentivize entities, that are selfish by nature, to perform honest behavior [45]. This subset evaluates our success on establishing this natural balance in order to incentivize honest and dishonest behavior.

Taking into account these considerations, we have elaborated the multiple sets of questions to which the possible answers are:

- *Yes*
- *Most likely Yes*
- *Maybe*
- *Most likely No*
- *No*

These questions are presented below. There are additional questions that are open-ended questions which are not featured in the list below. The answers to those questions won't be analyzed statistically but provide the conclusions based on their answers, in Section 6.3 (p. 93), after the statistical analysis of the questions listed below. The entire list of questions, including also the ones that are open-answered, are listed in Appendix A (p. 121).

Each question belongs to a group that is identified using two characters, as can be seen in the paragraphs and subparagraphs below.

Conflicting Interests and Competition (CI)

CI1 Are there conflicting interests of the entities (data processors and data repositories) in the current system?

CI2 Are the entities in the system competitors of each other?

CI3 Do you think that this competition leads to secrecy of internal procedures?

CI4 Can it incentivise entities to use other entities' data without giving them credit?

CI5 Would entities be more willing to provide data to other entities if they could trust others wouldn't use their data without giving them credit?

CI6 Would entities be more willing to provide data to other entities if they could trust others wouldn't use their data without giving them credit?

CI7 Could they make more progress if they provide the data to other entities and, therefore, are provided data by other entities?

CI8 Would entities provide their data to other entities if they were also provided data by them?

CI9 Would entities provide their data to data processors if they can use their computational resources?

Process of Building Trust (BT)

BT1 Are data processor entities currently unsure about what processing procedures (algorithms, configuration parameters and implementation) other processor entities take?

BT2 Would knowing the processing steps of an entity build trust towards the datasets it produces?

BT3 Does that trust make other entities willing to collaborate with that entity (the one that has built trust)?

BT4 Is sharing processing steps details for public datasets and its output a feasible way of evaluating an entity's processing procedures (in order to build trust on that entity)?

BT5 Are entities interested in sharing processing steps details for public datasets so that other entities are more willing to collaborate with them?

BT6 Would the entities involved be interested in licensing the data?

Feasibility and Benefits of the Approach (FB)

FB1 Does a decentralized database (blockchain, where all nodes agree on the data) that allows entities to register the processing results for public datasets provide a feasible way of evaluating an entity's processing procedures (in order to build trust on that entity)?

FB2 Are entities interested in sharing these results and respective processing steps on the decentralized registry so that other entities are more willing to collaborate with them?

FB3 Does providing a way to quickly validate the entries of that registry (with less decentralization than data storage, but still decentralized, through the voting rounds) further improve solving the problem (without nodes having to run the transformations themselves, which is computationally expensive)?

FB4 Does providing a way to verify the information without trusting validators give more confidence over it (even though it takes a lot of computational effort, but it could also be done only in certain cases)?

FB5 Does providing both worlds (by voting rounds and full validation) make the system more suitable for all use cases?

Benefits of the Solution for Patients (PT)

PT1 Can the traceability of the transformations of health data give patients more confidence over the data sharing system?

PT2 Can the traceability of the transformations help rewarding patients that have contributed to help reach important breakthroughs in disease treatment so that they are more willing to provide their health data?

Feasibility of the Rewards System (RS)

RS1 Is reputation a good incentive for entities to be honest in the process of voting for traceability data?

RS2 Does the loss of reputation provide a disincentive for being dishonest, by thinking that, if the majority is honest, their reputation will be lost?

RS3 Is gaining reputation a good incentive for verifying other's transformations data, given that it is a computationally expensive process?

RS4 Is fear of losing reputation enough to discourage entities from submitting unvalidated votes (without going through the processing)?

Due to specificity of the questions about the feasibility and the benefits of the solution, we anticipated that they would require an explanation of the approach. Therefore, we have decided that those questions should be preceded by an explanation of the solution developed. Since the approach is intended to solve the issues we found on exploring the field of health research data, we decided to place the questions about the context before the questions about the feasibility and

the benefits of the approach. As a result of establishing this flow of questions and explanations as an important aspect of the interview, we decided that a live interview would be the best choice since we could guide the expert through the thought process and even further adapt the questions to their understandings in the field.

6.3 Interview Results Analysis

In this Section, we provide an analysis of the results of the interviews, based on statistics of the answers as well as some important detailed answers that help evaluating the contributions developed. They provide an evaluation of our main contributions based on the opinions of the experts, through statistical analysis of their answers. Furthermore, the evaluation also represents a contribution,

This section is divided into multiple sections, each corresponding to a group of questions. For each group of questions, we first provide a statistical overview of the answers that can be represented into one of the five standard answers, listed above, in Section 6.2.2 (p. 90). We also present some important conclusions resulting from the statistical analysis. Then, we provide the answers to the open-ended questions as well as some important answers to the standard questions that are not representable in one of the five standard answers. This is important since it provides important additional feedback and support to our main contributions as well as constructive criticism to it, with suggestions of improvement.

The Figures 6.1, 6.2, 6.3, 6.4 and 6.5 provide an statistical overview of the answers of the experts. Each figure respects to a group of questions that is analyzed separately.

6.3.1 Conflicting Interests and Competition (CI)

The chart on Figure 6.1 provides an overview of experts' answers to the questions about the conflicting interests and the competition in the system. As can be seen on the graph, almost all answers are "Yes", with the second most common answers being "Most likely Yes". This supports our understanding the current issues in the fields of health research data, presented in Section 4.1 (p. 61), allowing us to reach the following important conclusions:

- All experts confirm that there are conflicting interests between the entities in the system and that these entities are competitors of each other.
- All experts confirm that this competition leads to secrecy of internal procedures and that there is currently an incentive for entities to use other entities' ideas and data without giving them credit.
- All experts think that entities would be more willing to provide data to other entities if they could trust they wouldn't use their data without giving them credit.
- Some of the experts think the entities that this providence of data between entities could lead to more progress, whereas one expert says maybe.

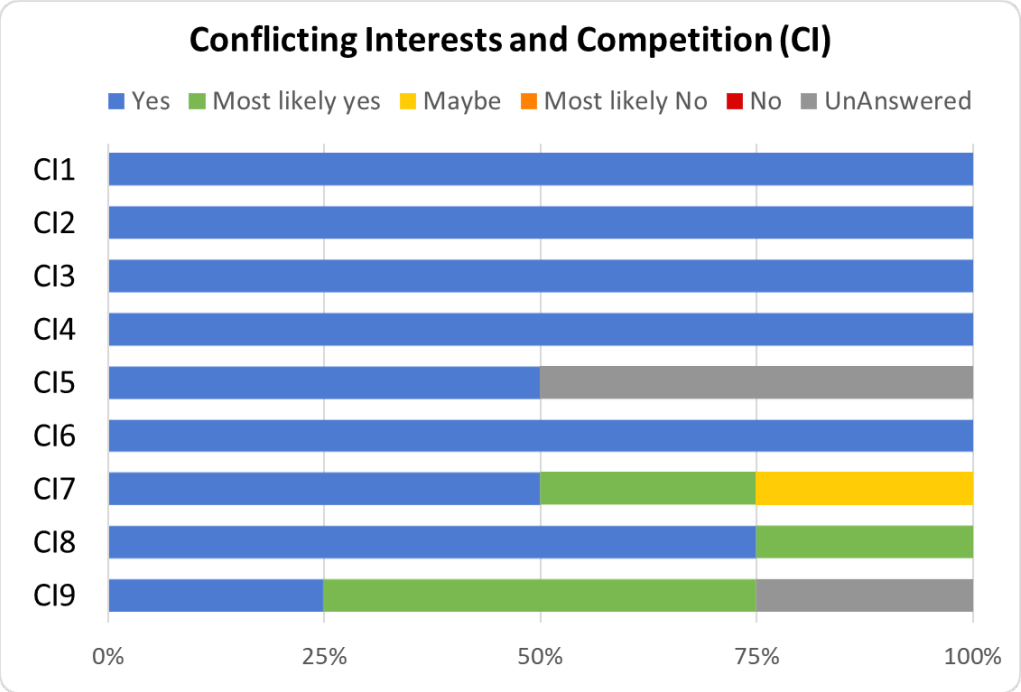

Figure 6.1: Answers to the questions in the Conflicting Interests and Competition (CI) group

- Three experts think that entities would provide data to others if they were also provided data by them and one thinks it would happen most likely. This leads us to the conclusion that cooperation leads to cooperation and that the feedback loop of incentive for cooperation could be created if the problems are fixed.

Now we delve into a more detailed analysis of the answers to the open-ended questions about the Conflicting Interests and Competition (CI). The paragraphs below provide the most interesting and constructive feedback from the experts for each of the questions.

Which conflicting interests do entities have? One of the experts points out competition as the main conflicting interest between research entities. Another expert provides us with a list of the interests of the entities:

- **Patients** want to have as much privacy as possible.
- **Research entities** have conflicting interests due to business issues, since they are competitors.

As well as with a list of current issues:

- Big research entities have advantages over the small ones, leading to a feedback loop of growing inequality in the system.
- Patients should be aware of how the system works and what happens to their data.

What do they want to achieve first? This questions aim to gather experts' opinion about what do the entities want to achieve first (what drives competition in the system). We were able to gather some significant feedback as answers to this question from the multiple experts interviewed, which is listed below.

- The entities want to be the first with publications, preferably in good journals, leading to better grants and funding.
- Different entities are interested in selling different services. The data processors try to derive value from the processing steps they make and bring together as much data as possible, whereas data repositories need to make sure they follow the rules of contracts they make and take responsibility of what happens to the data they are storing.
- The entities want to be the first with recognition in order to guarantee their sustainability in the ecosystem.

These points support the existence of competition and consequent secrecy of procedures, in an attempt to protect intellectual property from being used by other without receiving credit, holding back cooperation in the system.

Could they make more progress if they provide the data to other entities and, therefore, are provided data by other entities? Although this is a question supposed to be answered with on of the 5 possible answers, we did get open feedback from one of the experts that we would like to list. The expert's feedback supports our incentive for collaboration through sharing of the data transformations in the traceability data registry, pointing out that a researcher would be incentivized to share because they would get credit for providing that data as the entities that use it get credit where it is due (in their own research) which could generate recognition for the entity that has shared the data. The interviewee also points out that the entity could get collaborations through that sharing by the others who were provided data, as a reward. This supports that there are currently incentives for cooperation in the system, which can be explored.

Would entities provide their data to data processors if they can use their computational resources? With regards to this questions, we would like to point out that one of the "Most likely yes" answers is justified by the expert under the condition of need of those computational resources. The expert thinks that the entities would provide data in the exchange for computational resource if they need them, meaning, if they don't have them, which happens in case of a very heavy processing procedure or a high amount of data. This supports that there are currently incentives for cooperation in the system, which can be explored.

6.3.2 Process of Building Trust (BT)

The graph on Figure 6.2 provides an overview of experts' answers to the questions about the uncertainty on the processing procedures of each other, how it can impact cooperation and how

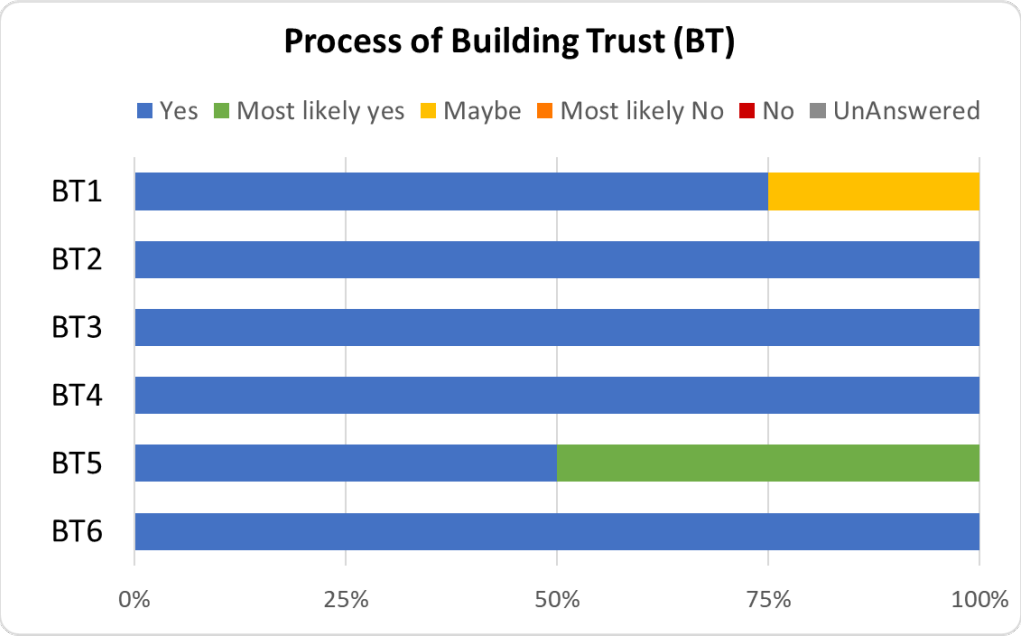

Figure 6.2: Answers to the questions in the Process of Building Trust (BT) group

could this be overcome by a process of building trust. The answers for this group of questions provide an even higher support for our understanding of the current system then the ones of the last group, with more than 95% of the answers for all evaluation questions being "Yes". This further supports our first our understanding the current issues in the fields of health research data, presented in Section 4.1 (p. 61). It also supports the importance of a mechanism of building trust in the ecosystem, as well as the feasibility of our proposed mechanism, which is implemented in the solution, described in Chapter 5 (p. 67). There are some important conclusions to point out, understandable from the statistical analysis that help support our contributions:

- Three experts confirm that there is currently uncertainty in the system about the processing procedures between the entities and one says "Maybe".

- All experts confirm that having confidence over the processing steps of an entity builds trust towards it and that this trust makes other entities more willing to collaborate with the entity. This supports the importance of providing a feasible mechanism for entities to be aware of an entity's processing procedures and how it could incentivize cooperation in the system.

- All experts think that sharing the processing steps and the output for public datasets is a feasible mechanism of evaluating an entity's processing procedures. This supports the feasibility of our proposed mechanism, part of our approach, which was then implemented in the prototype.

- Half of the experts think that entities would be interested in using this mechanism of sharing the processing steps details, and the other half of them think that they most likely would. This supports the mechanism, by providing trust over its adoption by the entities of the system.

Now we delve into a more detailed analysis of the answers to the open-ended questions about the Process of Building Trust (BT). The paragraphs below provide the most interesting and constructive feedback from the experts for each of the questions.

What are the consequences of the uncertainty about the processing procedures between data procedure entities? Two of the experts point out the fact of difficulty or even impossibility of replicating the results, making it impossible to compare them and, therefore, to cooperate. One of the experts also points out the impossibility of further research on top of a certain research, due to the impossibility of running the same processing procedure. This supports our review of the current issues as well as the need for creating a solution for this problem.

What current methods are there in the current system that entities can use to know each others' processing steps? Two of the experts believe there is currently no feasible method. One of the experts mentions the existence of a "Method" section upon publishing but explains that this is an informal and non-standard method. Another expert says there is currently a mechanism of chain of trust where entities review the processing procedures in a chain of reviews, allowing entities to have confidence about the entire chain, if they have confidence about one of the processing procedure of one entity of the chain. None of the methods use a decentralized storage of information, capable of providing confidence over the immutability of the data neither a fully decentralized mechanism of voting for and validating that data.

Is sharing processing steps details for public datasets a feasible way of evaluating an entity's processing procedures (in order to build trust on that entity)? All the experts consider that sharing the processing steps taken for a dataset that both parties have access to and the output of the processing is a feasible way of evaluating an entity's processing procedures. Though, three of the experts consider that public datasets are a feasible material to perform this, one of the experts thinks that it would greatly add value to the process if that data was private and shared between the entities that want to collaborate. This does not go against the principles of our approach and support of the respective implementation in its prototype as it can still be done without loss of privacy, as explained in Chapter 5 (p. 67). However, it provides constructive criticism to our initial prediction on how the system could be used, while also supporting its adaptability to multiple use cases.

6.3.3 Feasibility and Benefits of the Approach (FB)

The graph on Figure 6.2 provides an overview of experts' answers to the questions about the feasibility and the benefits of the approach and respective prototype, presented in Chapter 5 (p.

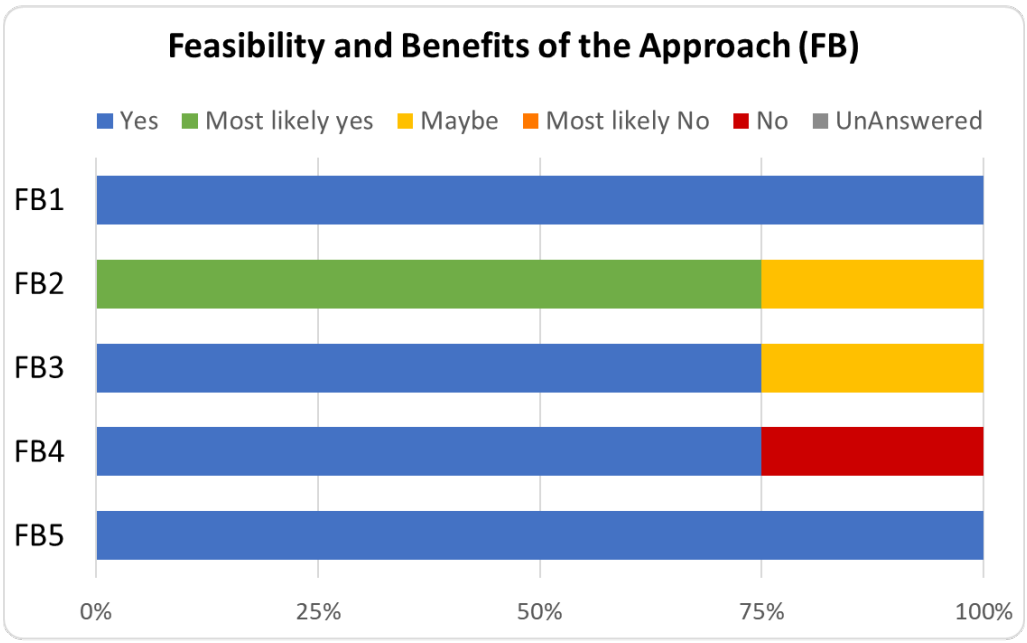

Figure 6.3: Answers to the questions in the Feasibility and Benefits of the Approach (FB) group

67) and how it could achieve the desired goal of improving the process of sharing health research data. The answers for this group of questions provide support for our second contribution which is the solution developed. It also supports the analysis of the current solutions to solve the problem and the establishment of the best of both worlds between these solutions, explained in Section 3.5.2 (p. 52). There some important conclusions to point out, understandable from the statistical analysis that help supporting the feasibility and the benefits of the solution:

- All experts agree that a decentralized registry of traceability data where entities could register the processing steps and output for public datasets provides a feasible mechanism of evaluating and entity's processing procedures. Although this question is similar to the one asked in the last group, here we evaluate if the specific case of blockchain is a good choice for implementing such mechanism since it is decentralized and all entities agree on what is registered there, preventing tampering of information.

- Three of the experts think that entities would most likely be interested in sharing this information on the blockchain so that other entities could be more certain about their processing procedures.

- Three of the experts think that the quick mechanism for validating the blockchain (by trusting the validators of the voting system) further improves solving the current issues, whereas one is not so sure about it.

- Three of the experts think that still providing the possibility of validating the information without trusting the validators (although it is a computationally heavy procedure) would still be useful so that entities could have more trust (since they are trusting more entities rather than just the validators, increasing decentralization). Although three experts approve the feature as important, one of them thinks that it would not provide more trust over the information. This point of view, although unexpected, also proves to be interesting and constructive, as we'll see below, when we delve deeper into the wider opinions for this group of questions.

Now we delve into a more detailed analysis of the answers to the open-ended questions about the Feasibility and Benefits of the Approach (FB). The paragraphs below provide the most interesting and constructive feedback from the experts for each of the questions.

Does a decentralized database (blockchain, where all nodes agree on the data) that allows entities to register the processing results for public datasets provide a feasible way of evaluating an entity's processing procedures (in order to build trust on that entity)? Although all experts think that the answer is "Yes", we would like to present the important feedback of one expert to the question. One of the experts has made clear that the fact that the traceability data is stored in a decentralized registry provides confidence that the record is identical for all nodes of the network in the sense that they reach consensus over it, is important to further increase the confidence over the processing procedure of an entity. This supports the use of blockchain in our solution since its properties are valuable for the context, as well as the success of applying it to the solution.

Are entities interested in sharing these results and respective processing steps on the decentralized registry so that other entities are more willing to collaborate with them? This questions aims to evaluate the possible adoption of the mechanism of the solution by the research entities. Although almost all the experts believe the entities would most likely be interested in sharing the processing steps, one of the expert believes that the action of creating a traceability data entry may not work well in some cases with a fee. He thinks that there could be entities that would be interested in just creating the traceability data and others in validating the data.

Although we aimed to provide a usability to the reputation within our system by placing a fee in the creation action, we have anticipated that there could be the need for customization. Due to this, we implemented the prototype as customizable as possible and, therefore, there is support to turn off this fee and even turn it into a reward, as explained in Chapter 5 (p. 67). This explanation provides constructive criticism to our initial prediction on how the system could be used, while also further supporting its adaptability to multiple use cases.

Does providing a way to quickly validate the entries of that registry (with less decentralization than data storage, but still decentralized, through the voting rounds) further improve solving the problem (without nodes having to run the transformations themselves, which

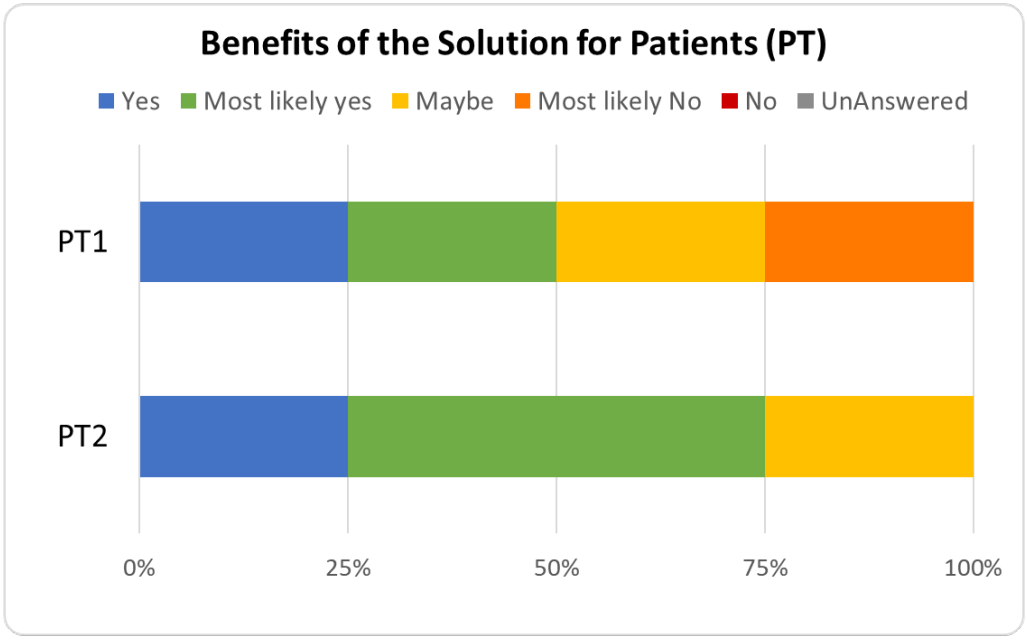

Figure 6.4: Answers to the questions in the Patients (PT) group

is computationally expensive)? Although the majority of the opinions are concentrated in the positive side, we would like to point out the importance given by one of the experts to the fact that the threshold of votes necessary must be reasonable, taking into account the size of the network. This supports the adaptability of our implementation, since the value is easily customizable by the user. It also provides constructive criticism to our implementation in the sense that, since it is an important feature, it could be even more customizable, allowing the system to find the correct value, as the network size increases or even allowing the creator to customize the value, are immediate understandable suggestions of improvement.

6.3.4 Benefits of the Solution for Patients (PT)

The graph on Figure 6.2 provides an overview of experts' answers to the questions about the benefits of the approach for patients. Although they are not the main focus of our approach, since it does not aim to serve them directly, patients are important actors in the ecosystem because they are who supply the health data to research entities in the first place. We did not expect the answers to be as positive as for the other groups of questions, since patients were not the main focus of our contributions. Yet, we still wanted to assess how could the solution benefit the users that provide data to the system. The conclusions understandable from the statistical analysis are the following:

- With regards to the traceability data giving patients more trust over the system, the experts' opinions diverge significantly. We got an answer for each of the options "Yes", "Most likely

Yes", "Maybe", "Most likely No". There are not much conclusions to take from this question, due to the divergence of opinions of the experts.

- With regards to being able to reward patients due to the possibility of determining the provenance of the data (if it was registered on the blockchain to that point), the opinions of the experts are more concentrated on the positive side than the previous question, but they still diverge.

Now we delve into a more detailed analysis of the answers to the open-ended questions about the Benefits of the Solution for Patients (PT). The paragraphs below provide the most interesting and constructive feedback from the experts for each of the questions.

Can the traceability of the transformations of health data give patients more confidence over the data sharing system? Although the opinions of the experts diverge significantly with regards to this question, there is an important open answer from one of the experts. One of the experts thinks that this transparency to the first provider of information to the system is very important. Although not being our main focus, we believe the solution can provide useful information to patients, in the sense that they can be aware of what happens to their data and how it is processed.

Can the traceability of the transformations help rewarding patients that have contributed to help reach important breakthroughs in disease treatment so that they are more willing to provide their health data? Similarly to the previous question, the opinions of the experts are very divergent. The same expert that has provided us with important feedback for the last question has also explained that rewarding patients that contribute to a certain research should be rewarded and has classified it as a very important feature. However, the expert also explained that this would need to be done outside of the system, depending on external adoption of procedures which are not provided by our solution. This supports the feature of being able to determine the provenance of the data, based on the traceability data stored as an important feature, to facilitate the process of rewarding patients. It also points out the limitations of the solution in being able to reach patients, as it was not our main focus (which was more driven to research entities), working as an immediate suggestion of further improvement.

6.3.5 Feasibility of the Rewards System (RS)

The graph on Figure 6.2 provides an overview of experts' answers to the questions about the feasibility of the rewards system, presented in Section 5.3.2 (p. 70) and how it could achieve the desired goal of improving the process of sharing health research data. The answers for this group of questions provide further support for our second contribution which is the solution developed, specifically focusing on the rewards system. Blockchain requires a consensus algorithm so that the network can reach consensus without the need of trust in a single authority, as explained in Section 2.3 (p. 11). The consensus algorithm requires a rewards system so that entities are encouraged to perform the honest behavior and discouraged to perform dishonest behavior. This

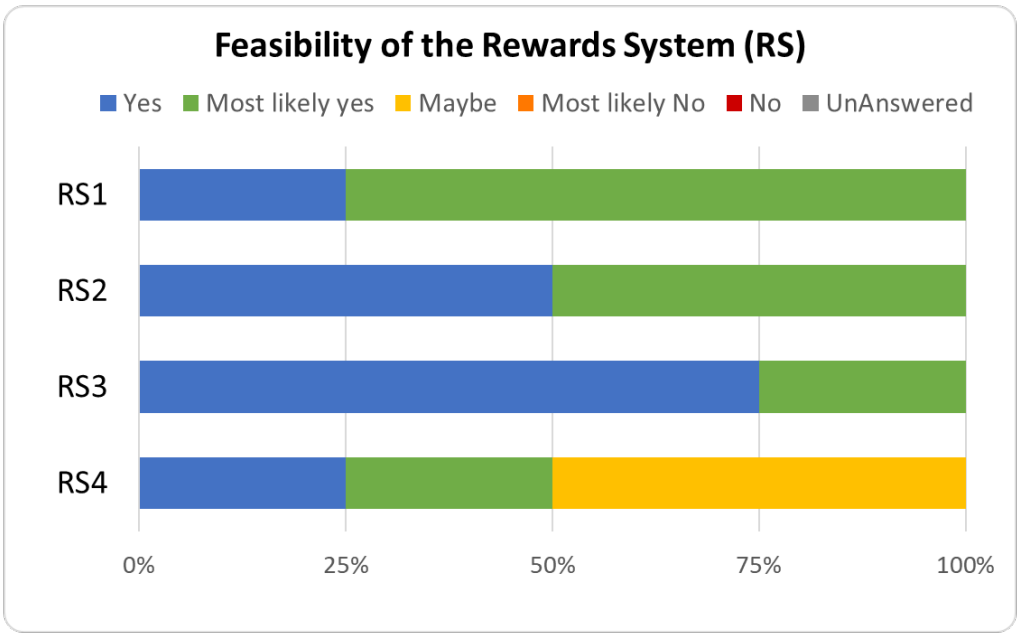

Figure 6.5: Answers to the questions in the Feasibility of the Rewards System (RS) group

requires establishing a balance that naturally incentivize entities, that are selfish by nature, to perform honest behavior. The answers of the experts evaluate our success on establishing this natural balance in order to incentivize honest and dishonest behavior. Given that more than 87% of the answers are positive, we can conclude that, overall, the rewards system was successful to establish this natural balance and incentive for honest behavior. There some important conclusions to point out, understandable from the statistical analysis that help supporting the feasibility of the rewards system:

- Three experts think that reputation is a most likely a good rewards resource (something they would want to have as much as possible) to incentivize entities to be honest in the process of voting for the traceability data. One of the experts thinks that it is a good rewards resource. This supports the use of reputation as an incentive resource, which was an exclusive contribution of this dissertation, not being present in any of the state of the art solutions.

- Half of the experts think that the loss of reputation is a good disincentive for being dishonest and the other half think that it most likely is, but are not so sure.

- Three experts think that gaining reputation is a good incentive for verifying other's traceability data (given that it is a computationally expensive procedure) and one thinks that it most likely is. This supports the feasibility of increasing cooperation in the system through the benefits of solution developed.

- With regards to discouraging entities from submitting unvalidated votes (without going through the computationally heavy process of verifying it), possibly causing anarchy in the voting system, the experts' opinions diverge. One thinks that system discourages entities to do it, one thinks it most likely does and the remaining two think that it maybe does. Due to the divergence of the opinions, we can't be so sure of our conclusions, but it is still positive feedback that supports our rewards system, with regards to avoiding anarchy in the system.

Now we delve into a more detailed analysis of the answers to the open-ended questions about the Feasibility of the Rewards System (RS). On the specific case of this group of questions, we will not analyze each question alone, since the answers of the experts to all the questions depend on the importance of the consensus resource for the entities.

All experts agree that being rewarded with the consensus resource is a good incentive for being honest and to perform computationally expensive tasks. They also agree that being penalized is a good disincentive for being dishonest. However, the most important aspect for the experts is the choice of the consensus resource on which all the answers depend, in the sense that if it is something they value a lot, they will definitely incentivized to behave honestly and disincentivized to behave dishonestly.

With regards to the importance of reputation to the entities: in general, the experts believe it is an important asset to the entities of the system. One of the experts explains he believes it is an important asset for both the academic community and the commercial entities.

With regards to avoiding anarchy in the system, avoiding unvalidated votes by the entities: almost all experts agree that loosing reputation in case the vote is decided to be wrong by the network is enough disincentive for the entities to perform such behavior. One of the experts mentions the importance of the impossibility of playing a Martingale game [43] as one of the key considerations in developing and evaluating the feasibility of the rewards system.

Half of the experts recognize the possibility of customizing the amounts of penalties as well as the possibility of having a penalty higher than the stake amount.

One of the experts suggests an improvement that consists of support for exponential penalties in the sense that as entities make successive mistakes they get increasing penalties. The expert believes it could further help avoiding dishonest behavior and anarchy in the system because there would be a higher penalty for cumulative dishonest behavior.

6.4 Conclusions

In this Chapter, we have performed an evaluation of the feasibility and the benefits of our analysis of the fields as well as the approach and respective implementation, presented in Chapter 5 (p. 67). The evaluation was made based on the opinion of experts in the fields of blockchain and health data research and collected through live interviews with these experts, due to the reasons explained in Section 6.2 (p. 88).

Taking into account the results analysis, in Section 6.3 (p. 93) both statistically and by providing open answers of experts to all questions, we were able to support our first main contribution of

analyzing the fields of health research, pointing out the current issues and taking some assumptions that make the solution both relevant and valid, described in Chapter 4 (p. 61).

We were also able to support the feasibility and benefits of the solution developed, explained in Chapter 5 (p. 67). The expert' opinions support its success in solving the issues identified (and validated in the group of questions before). They also show our limitations on anticipating all the possible use cases of the system and our success on overcoming the consequences of it, by implementing a solution that is highly customizable and, therefore, adaptable to all use cases, highlighting the importance of this important design principle.

Finally, the open answers of the experts also provide important constructive feedback, showing how the solution could be further improved with some of the suggestions listed above, in the Results Analysis, Section 6.3 (p. 93). It is important to note that different experts have provided contrary improvements suggestions, showing the importance of implementing them in a highly customizable approach, showing the importance of a highly customizable solution, that adapts to the multiple scenarios of health research data.

Chapter 7

Conclusions and Future Work

This chapter provides a reflection of all the work developed throughout this dissertation. Firstly, in Section 7.1 (p. 105), we provide a summary of the conclusions and lessons learned on the process of developing our contributions. Then, in Section 7.2 (p. 108), we provide a list of our most important contributions. In Section 7.3 (p. 109), we provide a list of the main challenges encountered throughout the development of this dissertation, as well as the methodology used to overcome them. At the end of the chapter, in Section 7.4 (p. 110), we provide a description of possible future work that can be done to improve the scientific analysis and the solution developed, further supporting the process of sharing health research data.

7.1 Summary

The main objectives of this dissertation are to explore the fields of health research data and leverage the properties of blockchain in order to architect an approach to support the process of sharing and tracing health research data. We wanted to implement the approach in a prototype that should be as customizable as possible in order to adapt to as much use cases as possible.

Therefore, to achieve the desired objectives, we first reviewed the fields of health research data so as to identify the current issues, presented in Chapter 1 (p. 1).

After identifying the issues, we have conducted a state of the art review, in Chapter 3 (p. 23), in order to identify the solutions closer to solving the issues identified in the current system of sharing health research data. At the end of Chapter 3, in Section 3.5 (p. 48), we present an analysis of the main advantages and disadvantages of these solutions, establishing the best trade-off between their mechanisms.

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, in Section 6, (p. 87).

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, in Section 4.3 (p. 63):

"*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, in Section 4.4 (p. 64), which support the proposed solution, our second main contribution.

The solution, described in Chapter 5 (p. 67), inserted in the context of a research project, described in Section 5.1 (p. 67), has its own objectives, described in detail in Section 5.2 (p. 67). Its main objective is to leverage the main conclusions of the analysis presented as our first main contribution and use blockchain to architect an approach that encompasses all the desired aspects for solving the issues presented. The approach is described in detail in Section 5.3 (p. 68), aims to establish the best of both worlds between the current solutions. The approach was implemented in a prototype, with the objectives to be as customizable as possible so it adapts to multiple use cases and open for expansion as possible, allowing further research and improvements to be developed.

Lastly, we provide an evaluation of all the work developed throughout this dissertation, through interviews with experts in the fields of health research and blockchain, as described in Chapter 6 (p. 87). Through the evaluation, we were able to respond to each of the research questions:

RQ1 "*Does providing a decentralized registry where entities can register traceability information for public data sets provide a feasible way of evaluating an entities processing procedures?*" The analysis of the experts' opinion suggests a positive answer to this research question, since all the experts have answered positively to the respective interview question. This supports the feasibility and the benefits of one of the main features of the solution, while also supporting our analysis of the issues of the current system of health research data as it was the basis for establishing this as an important feature of the solution. This feature is described in detail in Section 5.3.1 (p. 68).

RQ2 "*Does providing a mechanism to quickly agree on the validity of the entries of that registry (with less decentralization, through the voting rounds) further improve solving the problem?*" Through the analysis of the experts' opinion, we were able to partially support a positive answer to this research question, since 3 experts have answered positively to the respective interview question and 1 is unsure but believes that it may be the case. This supports the feasibility and the benefits of the extra feature of the solution, the Traceability Data Auditing System, described in Section 5.3.2.2 (p. 71), while also supporting our analysis of the advantages and disadvantages of the current solutions of the state of the art which was the basis for establishing this objective for the solution. This feature is described in detail in Section 5.3.2.2 (p. 71).

RQ3 "*Does providing full validation and voting rounds make the system more suitable for all use cases?*" The analysis of the experts' opinion suggests a positive answer to this research question, since all the experts have answered positively to the respective interview question. This

supports the feasibility and the benefits of the extra feature of the solution, while also supporting our analysis of the current solutions and establishment of the best trade-off between them which our solution was intended to implement. This feature is described in detail in Section 5.3.2.2 (p. 71). The analysis of the current solutions and establishment of the best trade-off are described in detail in Section 3.5.2 (p. 52).

RQ4 "*Is reputation a good incentive resource to improve cooperation on the system?*" The analysis of the experts' opinion suggests a positive answer to this research question, since all the experts provided positive answers to it. The experts approved the mechanisms used to incentivize honest behavior, supporting the structure of the incentive system, explained in Section 5.3.2.1 (p. 70). However, due to the nature of blockchain rewarding systems, as explained in Section 2.3 (p. 11), they did not provide us with any guarantees of success since it could only be fully approved if tested in the context. 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. Yet, three of the experts believe that reputation is very important to these entities, whereas one says it may be important, but is not so sure. This supports the feasibility and the benefits of the incentive system as well as the incentive resource, an important component of the extra improvement to this process that aims to incentivize entities to audit each other's traceability data. This incentive system is described in detail in Section 5.3.2.2 (p. 71).

Based on the information collected from the expert opinion, we supported that providing a decentralized registry of data transformations resulting from research improves the process of sharing health research data, incentivizing cooperation in the ecosystem. In this sense, entities can be more confident of the processing procedures of each other, without the need of trust in a central authority. This supports the interests of the different entities, in a system where there are multiple entities with competitive interests involved.

Furthermore, we also supported that providing a mechanism that allows entities to audit each other's data, further improves solving the current issues in the system, by further incentivizing cooperation. Moreover, the incentive system supporting the traceability data auditing system, also manages to support the interests of the multiple entities, naturally incentiving cooperation by issuing rewards upon correct behavior.

This way, through the analysis of the interview results described above, we were able to validate all the research questions, allowing us to validate also the hypothesis of this dissertation, presented in Section 4.3 (p. 63). Thus, the contribution of our research corresponds to the objectives established.

7.2 Contributions

In Section 4.1 (p. 61), 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 architected an approach which 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:

Review on the process of sharing health research data At the beginning of this dissertation, we started reviewing and exploring the current aspects of the health research system in order to identify the main issues with it. It was through this review, presented in Chapter 1 (p. 1), that we could identify the main issues with the system, presented in Section 4.1 (p. 61), supporting the process of formulating the hypothesis, in Section 4.3 (p. 63) and, therefore, the research questions, in Section 4.4 (p. 64), which are the base for this dissertation.

Literature Review on Health Research Data Taking into account the needs of the current system, resulting from the issues identified, we then started performing a literature review on the state of the art of data traceability. Resulting from this review, we obtained our second contribution, presented in Chapter 3 (p. 23). This allowed us to further formulate issues, also presented in Section 4.1 (p. 61), that consist of the problems with these solutions. At the end of the state of the art chapter, in Section 3.5.2 (p. 52), we provide an important part of this contribution, consisting of a comparison of the solutions closer to solving these problems. The comparison presents the main advantages and disadvantages with each of the mechanisms of the different solutions and establishes the best trade-off between them. Establishing this best trade-off, allowed us to formulate one of our main objectives for the approach, presented in Section 5.3 (p. 68).

iReceptorChain Solution Taking into account all the current issues with the health data research system and the problems with the solutions reviewed in the state of the art analysis, presented in Section 4.1 (p. 61), and consequently, through the formulation of a solution hypothesis, in Section 4.3 (p. 63), we developed a solution, described in Chapter 5 (p. 67). The solution encompasses several aspects, concepts and features:

Traceability data decentralized registry This is the first feature being implemented in the solution. It leverages the decentralized properties of blockchain to create a decentralized registry of traceability data, providing trust over the immutability of the data in a system where there are multiple entities that are competitors of each other. The approach for designing this component is explained in Section 5.3.1 (p. 68) and its implementation is described in Section 5.4 (p. 76).

Traceability data auditing system This is the extra feature implemented in the solution to further improve the feasibility and benefits of it. It aims to provide a way for entities to be able to audit the traceability data (say whether it's valid or not) in order to further support the process

of building trust by keeping all entities engaged on it in a cooperative way while also providing a computationally easier method to verify the information stored on the blockchain. The approach for designing this component is explained in Section 5.3.2.2 (p. 71) and its implementation is described in Section 5.4 (p. 76).

Incentive system This is not a feature, but instead a necessary architectural component for the **Traceability data auditing system**. It is important since the process of verifying the traceability data is computationally hard, requiring a high incentive to be performed. Therefore, there was the need to architect a rewarding 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 Proof of stake. The approach for designing this component is explained in Section 5.3.2.1 (p. 70) and its implementation is described in Section 5.4 (p. 76).

7.3 Challenges

Throughout the development of this dissertation, we had several challenges to overcome related to understanding the aspects and concepts of the field of health research data, that was described in Sections 2.5 (p. 18) and 2.6 (p. 19). It was also a challenge to find feasible current solutions, through the literature review, described in Chapter 3 (p. 23). Finally, architecting the incentive system for the solution, described in Section 5.3.2.1 (p. 70), was also a challenge. This details regarding each of the challenges are the following:

Understanding the health research data system Understanding the system of sharing health research data was the first challenge on the process of developing this dissertation, as didn't have experience on the field. In order to overcome this challenge, we've had several meetups in the context of the iReceptorPlus project, described in Section 2.6 (p. 19), with the intent to learn more about the context with more experienced members.

Literature Review on Health Research Data Finding current solutions for data traceability using blockchain was also a challenge, due to the lack of literature regarding the concept. In order to overcome this challenge, we have increased the extension of our search and also considered solutions that target a different context, in an attempt to extract the valuable mechanisms and techniques from them and adapt them to our context. It was through the comparison of these solutions, described in Section 3.5.2 (p. 52) that we have established the best trade-off between the solutions that most suit our context.

Incentive System Creating an incentive system for blockchain applications requires balancing several aspects and concepts to establish a natural incentive for honest behavior, without the need of policing, as described in Section 2.3 (p. 11). In order to overcome this challenge, we have

invested a several amount of effort in putting together the concepts and aspects learned about the current system of health research data, explained in Sections 2.5 (p. 18) and 2.6 (p. 19), and the current blockchain consensus algorithms, explained in Section 2.3.3 (p. 14) in an attempt to analyze which ones would provide more advantages and further modify them to understand which one would best suit the context. In order to make the consensus algorithm suitable to our context, the most important aspect was to choose the consensus resource that would be rewarded upon honest behavior and removed upon dishonest behavior. Through the analysis of the context, we have established that reputation is an important asset for the entities of the system and, therefore, it would be a feasible consensus resource. After establishing this key aspect of the blockchain consensus algorithm, we have architected a slightly modified proof of stake algorithm, as described in Section 5.3.2.1 (p. 70), and then implemented it, as explained in Section 5.4 (p. 76).

7.4 Future Work

Similarly to most research projects, there are improvements that can be made as future work to this dissertation.

During the live interviews, the experts have provided constructive criticism to our approach, with suggestions of improvement that could be implemented in the future. Some of the improvements that could be made to the solution are:

- Support for exponentially growing penalties for the entities that perform cumulative incorrect behavior. In this sense, entities are incentivized to learn from their mistakes, so that they don't compromise the correction of the consensus of the system. It is important to note that another expert has pointed out his uncertainty about the effects of penalties, in the sense that it could be unfair in a context where it is assumed that entities do not have an incentive to perform dishonest behavior and they would learn from their mistakes even without a penalty. The contrary expert opinions highlight the importance of implementing this future improvement in a customizable way, allowing to be turned off, to make the solution as adaptible as possible.

- Support for a minimum vote threshsold[1] that automatically adapts to the size of the network in the sense that smaller networks require less number of votes to finish the voting round than larger networks.

- Possibility for the creator of a traceability data entry to increase the reward that voters get for verifying his traceability data entry, in order to incentivize higher rewards for entries that are harder to verify.

Another important improvement that could be made is the validation methodology used to support the feasibility and the benefits of the approach. In the future, we would like to test the

[1] 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, as explained in Section 5.3 (p. 68).

prototype, in which the approach was implemented, in a real consortium of research entities and evaluate its performance through a case study. We believe that this would further help improving the approach.

References

[1] Hyperledger Architecture, Volume 1. *Oncology Letters*, 16(4):4129–4136, 2018.

[2] The Ammbr, Public Token, Sale Has, and Been Cancelled. MedicalChain White Paper. pages 1–42, 2017. Available at https://medicalchain.com/Medicalchain-Whitepaper-EN.pdf (accessed at February 2020).

[3] Gregory R. Andrews. *Concurrent Programming: Principles and Practice*. Benjamin-Cummings Publishing Co., Inc., 1 edition, 1991.

[4] Elli Androulaki, Artem Barger, Vita Bortnikov, Srinivasan Muralidharan, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Chet Murthy, Christopher Ferris, Gennady Laventman, Yacov Manevich, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Weed Cocco, and Jason Yellick. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *Proceedings of the 13th EuroSys Conference, EuroSys 2018*, 2018-January, 2018.

[5] Prashant Ankalkoti and Santhosh. A relative study on bitcoin mining. *Imperial Journal of Interdisciplinary Research (IJIR)*, 3, 05 2017.

[6] Hasib Anwar. Consensus algorithms: The root of the blockchain technology, 2008. Available at https://101blockchains.com/consensus-algorithms-blockchain/ (accessed at February 2020).

[7] Dian Abadi Arji, Fandhy Bayu Rukmana, and Riri Fitri Sari. A design of digital signature mechanism in NDN-IP gateway. *2019 International Conference on Information and Communications Technology, ICOIACT 2019*, pages 255–260, 2019.

[8] Stephen Arsenault. Aura Consensus Protocol Audit, 2020. Available at https://github.com/poanetwork/wiki/wiki/Aura-Consensus-Protocol-Audit (accessed at June 2020).

[9] G. Arulkumaran and N. R. Rajalakshmi. Named data networking (NDN), internet architecture design and security attacks. *International Journal of Innovative Technology and Exploring Engineering*, 8(11 Special Issue):1281–1284, 2019.

[10] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. MedRec: Using blockchain for medical data access and permission management. *Proceedings - 2016 2nd International Conference on Open and Big Data, OBD 2016*, pages 25–30, 2016.

[11] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. *Proceedings - IEEE Symposium on Security and Privacy*, pages 459–474, 2014.

[12] BigchainDB GmbH. BigchainDB: The blockchain database. *BigchainDB. The blockchain database.*, (May):1–14, 2018.

[13] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. SoK: Research perspectives and challenges for bitcoin and cryptocurrencies. *Proceedings - IEEE Symposium on Security and Privacy*, 2015, (July):104–121, 2015.

[14] Alexandre A Boschi, Rogério Borin, Julio Cesar Raimundo, and Antonio Batocchio. An exploration of blockchain technology in supply chain management. (October):27–28, 2018.

[15] Stefan Brands. *Rethinking Public Key Infrastructures and Digital Certificates*. Mit Press, 2 edition, 2018.

[16] Vitalik Buterin. A next-generation smart contract and decentralized application platform. *Etherum*, (January):1–36, 2014.

[17] Vitalik Buterin. Ethereum website, 2020. Available at https://ethereum.org/ (accessed at February 2020).

[18] Eliya Buyukkaya, Maha Abdallah, and Romain Cavagna. VoroGame : A hybrid P2P architecture for massively multiplayer games. *2009 6th IEEE Consumer Communications and Networking Conference, CCNC 2009*, (February 2009), 2009.

[19] Christian Cachin, Simon Schubert, and Marko Vukolić. Architecture of the Hyperledger Blockchain Fabric. *Leibniz International Proceedings in Informatics, LIPIcs*, 70:24.1–24.16, 2017.

[20] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Conference on the Theory and Application of Cryptography*, pages 319–327. Springer, New York, NY, 1988.

[21] V. C. Chen, H., Chiang, R. H., & Storey. *Understanding Big Data analytics*. McGraw-Hill Osborne Media, 1 edition, 2011.

[22] Birgit Clark and Ruth Burstall. Blockchain, IP and the pharma industry—how distributed ledger technologies can help secure the pharma supply chain. *Journal of Intellectual Property Law & Practice*, 13(7):531–533, 2018.

[23] Aengus Collins et al. The global risks report 2018. In *Geneva: World Economic Forum*, 2018.

[24] Brian Corrie, Jerome Jaglale, Bojan Zimonja, Nishanth Marthandan, Emily Barr, Laura Gutierrez Funderburk, Scott Christley, Lindsay Cowell, Jamie Scott, and Felix Breden. ireceptor: A case study in the importance of standards for data sharing.

[25] Brian D Corrie, Nishanth Marthandan, Bojan Zimonja, Jerome Jaglale, Yang Zhou, Emily Barr, Nicole Knoetze, Frances MW Breden, Scott Christley, Jamie K Scott, et al. ireceptor: A platform for querying and analyzing antibody/b-cell and t-cell receptor repertoire data across federated repositories. *Immunological reviews*, 284(1):24–41, 2018.

[26] Nicolas T. Courtois and Lear Bahack. On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency. *arXiv preprint arXiv:1402.1718.*, 2014.

[27] M. Creydt and M. Fischer. Blockchain and more - Algorithm driven food traceability. *Food Control*, 105(May):45–51, 2019.

[28] Joan Daemen and Vincent Rijmen. The block cipher rijndael. *Lecture Notes in Computer Science - LNCS*, 1820:277–284, 01 1998.

[29] Ivan Damgård. A design principle for hash functions. pages 416–427, 01 1989.

[30] Paul De Hert and Vagelis Papakonstantinou. The proposed data protection Regulation replacing Directive 95/46/EC: A sound system for the protection of individuals. *Computer Law and Security Review*, 28(2):130–142, 2012.

[31] Kevin Driscoll, Brendan Hall, Håkan Sivencrona, and Phil Zumsteg. Byzantine Fault Tolerance , from Theory to Reality 1 What You Thought Could Never Happen. *Thought A Review Of Culture And Idea*, (2):235–248, 2003.

[32] Abeer Elbahrawy, Laura Alessandretti, Anne Kandler, Romualdo Pastor-Satorras, and Andrea Baronchelli. Evolutionary dynamics of the cryptocurrency market. *Royal Society Open Science*, 4(11), 2017.

[33] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, July 1985.

[34] D. Fett, R. Küsters, and G. Schmitz. The web sso standard openid connect: In-depth formal security analysis and security guidelines. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 189–202, 2017.

[35] Victor Gayoso Martínez, Lorena González-Manzano, and Agustín Martín Muñoz. Secure elliptic curves in cryptography. *Computer and Network Security Essentials*, pages 283–298, 2017.

[36] Peter Gazi, Aggelos Kiayias, and Alexander Russell. Stake-bleeding attacks on proof-of-stake blockchains. *Proceedings - 2018 Crypto Valley Conference on Blockchain Technology, CVCBT 2018*, pages 85–92, 2018.

[37] A. Geissbuhler, C. Safran, I. Buchan, R. Bellazzi, S. Labkoff, K. Eilenberg, A. Leese, C. Richardson, J. Mantas, P. Murray, and G. De Moor. Trustworthy reuse of health data: A transnational perspective. *International Journal of Medical Informatics*, 82(1):1–9, 2013.

[38] Reno Varghese George, Hari Om Harsh, Papri Ray, and Alex K. Babu. Food quality traceability prototype for restaurants using blockchain and food quality data index. *Journal of Cleaner Production*, 240:118021, 2019.

[39] Stephen L George and Marc Buyse. Clinical Trial Perspective part of Data fraud in clinical trials. *Clin. Invest*, 5(2):161–173, 2015.

[40] Shafi Goldwasser. Interactive proof systems. *Computational Complexity Theory*, 18(1):108–128, 1989.

[41] Andres Guadamuz. PayPal: The Legal Status of P2P Payment Systems. *Computer Law & Security Review*, 20(4):293–299, 2008.

[42] Vinay Gupta. A brief history of blockchain, 2017. Available at https://hbr.org/2017/02/a-brief-history-of-blockchain (accessed at February 2020).

[43] P. Hall and C.C. Heyde. *Martingale Limit Theory and Its Application*. Probability and mathematical statistics. Academic Press, 1 edition, 1980.

[44] David J. Hand. Principles of data mining. *Drug Safety*, 30(7):621–622, 2007.

[45] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun. A blockchain based truthful incentive mechanism for distributed p2p applications. *IEEE Access*, 6:27324–27335, 2018.

[46] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on Bitcoin's peer-to-peer network. *Proceedings of the 24th USENIX Security Symposium*, (August):129–144, 2015.

[47] Denise J. Hills, Robert R. Downs, Ruth Duerr, Justin C. Goldstein, Mark A. Parsons, and Hampapuram K. Ramapriyan. The importance of data set provenance for science. *Eos*, 97(3):10–11, 2016.

[48] Hui Huang, Xiaofeng Chen, and Jianfeng Wang. Blockchain-based multiple groups data sharing with anonymity and traceability. *Science China Information Sciences*, 63(3):1–13, 2020.

[49] G. Hurlburt. Might the blockchain outlive bitcoin? *IT Professional*, 18(2):12–16, Mar 2016.

[50] Marco Iansiti and Karim Lakhani. The truth about blockchain, 2017. Available at https://hbr.org/2017/01/the-truth-about-blockchain?referral=03758&cm_vc=rr_item_page.top_right (accessed at February 2020).

[51] Raj Jain. Internet 3.0: Ten problems with current internet architecture and solutions for the next generation. *Proceedings - IEEE Military Communications Conference MILCOM*, 2006.

[52] Bonnie Kaplan. Selling health data: De-identification, privacy, and speech. *Cambridge Quarterly of Healthcare Ethics*, 24(3):256–271, 2014.

[53] Ghassan O. Karame, Elli Androulaki, and Srdjan Čapkun. Double-spending fast payments in Bitcoin. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 906–917, 2012.

[54] Henry M. Kim and Marek Laskowski. Toward an ontology-driven blockchain design for supply-chain provenance. *Intelligent Systems in Accounting, Finance and Management*, 25(1):18–27, 2018.

[55] Kiyun Kim. Modified Merkle Patricia Trie — How Ethereum saves a state, 2018. Available at https://medium.com/codechain/modified-merkle-patricia-trie-how-ethereum-saves-a-state-e6d7555078dd (accessed at June 2020).

[56] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The State of Elliptic Curve Cryptography. *Designs, Codes, and Cryptography*, 19(2-3):173–193, 2000.

[57] Bert Jaap Koops. The trouble with European data protection law. *International Data Privacy Law*, 4(4):250–261, 2014.

[58] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, pages 839–858, 2016.

[59] Patty Kostkova, Helen Brewer, Simon de Lusignan, Edward Fottrell, Ben Goldacre, Graham Hart, Phil Koczan, Peter Knight, Corinne Marsolier, Rachel A. McKendry, Emma Ross, Angela Sasse, Ralph Sullivan, Sarah Chaytor, Olivia Stevenson, Raquel Velho, and John Tooke. Who Owns the Data? Open Data for Healthcare. *Frontiers in Public Health*, 4(February), 2016.

[60] Christopher Kuner. The European Commission's Proposed Data Protection Regulation: A Copernican Revolution in European Data Protection Law. *Privacy {&} Security Law Report*, (February):1–15, 2012.

[61] Protocol Labs. libp2p: A modular network stack, 2020. Available at https://libp2p.io/ (accessed at June 2020).

[62] Yuxin Liao and Ke Xu. Traceability System of Agricultural Product Based on Block-chain and Application in Tea Quality Safety Management. *Journal of Physics: Conference Series*, 1288(1), 2019.

[63] Iuon Chang Lin and Tzu Chun Liao. A survey of blockchain security issues and challenges. *International Journal of Network Security*, 19(5):653–659, 2017.

[64] Laure A Linn and Martha B Koo. Blockchain For Health Data and Its Potential Use in Health IT and Health Care Related Research. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop*, pages 1 – 10, 2016.

[65] C. Mainka, V. Mladenov, J. Schwenk, and T. Wich. Sok: Single sign-on security — an evaluation of openid connect. In *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 251–266, 2017.

[66] J Manyika, M Chui Brown, Bughin B. J., R Dobbs, C Roxburgh, and A Hung Byers. Big data: The next frontier for innovation, competition and productivity. *McKinsey Global Institute*, (June):156, 2011.

[67] Trent Mcconaghy, Rodolphe Marques, Andreas Müller, Dimitri De Jonghe, Troy Mcconaghy, Greg Mcmullen, Ryan Henderson, Sylvain Bellemare, and Alberto Granzotto. BigchainDB: A Scalable Blockchain Database (DRAFT). *BigchainDB*, pages 1–65, 2016.

[68] Matthias Mettler and M A Hsg. Blockchain technology in healthcare: The revolution starts here. *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services, Healthcom 2016*, pages 16–18, 2016.

[69] Sandino Moeniralam. A Blockchain based Data Production Traceability System. (February):1–17, 2018.

[70] M. A. Mohamed. A survey on elliptic curve cryptography. *Applied Mathematical Sciences*, 8(153-156):7665–7691, 2014.

[71] Noman Mohammed, Xiaoqian Jiang, Rui Chen, Benjamin C.M. Fung, and Lucila Ohno-Machado. Privacy-preserving heterogeneous health data sharing. *Journal of the American Medical Informatics Association*, 20(3):462–469, 2013.

[72] James Brennan Nadine Gobron, Mathias Disney, Yves Govaerts, Jean-Luc Widlowski and Corrado Mio. Quality Assurance for Essential Climate Variables. *Quality report of the selected radiative transfer models*, 17(607405):10234, 2015.

[73] Satoshi Nakamoto and A Bitcoin. A peer-to-peer electronic cash system. 2008. Available at https://bitcoin.org/bitcoin (accessed at February 2020).

[74] Nature. Big data, 2008. Available at http://www.nature.com/news/specials/bigdata/index.html (accessed at February 2020).

[75] Benedikt Notheisen, Jacob Benjamin Cholewa, and Arun Prasad Shanmugam. Trading Real-World Assets on Blockchain: An Application of Trust-Free Transaction Systems in the Market for Lemons. *Business and Information Systems Engineering*, 59(6):425–440, 2017.

[76] Tatsuaki Okamoto. An efficient divisible electronic cash scheme. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 963:438–451, 1995.

[77] Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 576 LNCS(d):324–337, 1992.

[78] Christine O'Keefe. Privacy and the use of health data - reducing disclosure risk. *electronic Journal of Health Informatics; Vol 3, No 1 (2008): Special Issue on Privacy and Security; e5*, 3, 01 2008.

[79] Linus Umezuruike Opara. Traceability in agriculture and food supply chain. *Journal of Food Agriculture and Environment*, 1:101–106, 2003.

[80] Aravind Ramachandran and Dr. Murat Kantarcioglu. Using Blockchain and smart contracts for secure data provenance management. *arXiv preprint arXiv:1709.10000*, 2017.

[81] Matei Ripeanu. Peer-to-Peer Architecture Case Study: Gnutella Network. pages 1–11, 2001.

[82] R L Rivest, A Shamir, and L Adleman. A Method for Obtaining Digital Signatures and Public-Key Components. *Communications of the ACM*, 21(2):120–126, 1978.

[83] Meni Rosenfeld. Analysis of Hashrate-Based Double Spending. *arXiv preprint arXiv:1402.2009*, pages 1–13, 2014.

[84] Nicolas Van Saberhagen. CryptoNote v 2.0. *Self-published*, pages 1–20, 2013.

[85] Shazia Sadiq, Maria Orlowska, Wasim Sadiq, and Cameron Foulger. Data Flow and Validation in Workflow Modelling. *Adc'04*, 27(April 2013):207–214, 2004.

[86] Fahad Saleh. Blockchain Without Waste: Proof-of-Stake. *SSRN Electronic Journal*, 2018.

[87] Antonio Fernando Cruz Santos, Ítalo Pereira Teles, Otávio Manoel Pereira Siqueira, and Adicinéia Aparecida de Oliveira. Big data: A systematic review. *Advances in Intelligent Systems and Computing*, 558:501–506, 2018.

[88] Stuart E. Schechter, Rachel A. Greenstadt, and Michael D. Smith. Trusted Computing, Peer-to-Peer Distribution, and The Economics of Pirated Entertainment. *Economics of Information Security*, pages 59–69, 2006.

[89] Jonathan J.M. Seddon and Wendy L. Currie. Cloud computing and trans-border health data: Unpacking U.S. and EU healthcare regulation and compliance. *Health Policy and Technology*, 2(4):229–241, 2013.

[90] Sarah Spiekermann, Alessandro Acquisti, Rainer Böhme, and Kai Lung Hui. The challenges of personal data markets and privacy. *Electronic Markets*, 25(2):161–167, 2015.

[91] William Stallings. *Cryptography and Network Security*. Pearson Education India, 5 edition, 2004.

[92] Maarten Steen and Andrew Tanenbaum. *Distributed systems*, volume 01. Maarten van Steen Leiden, The Netherlands, 3 edition, 1993.

[93] Kensworth Subratie, Saumitra Aditya, Vahid Daneshmand, Kohei Ichikawa, and Renato Figueiredo. On the design and implementation of IP-over-p2p overlay virtual private networks. *IEICE Transactions on Communications*, E103B(1):2–10, 2020.

[94] Shakirat Sulyman. Client-server model. *IOSR Journal of Computer Engineering*, 16:57–71, 01 2014.

[95] Scott Nadal Sunny King. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, 1919(January):1–27, 2017.

[96] Shawn Tabrizi. Parity Substrate: Hello, Substrate!, 2019. Available at https://www.parity.io/hello-substrate/ (accessed at June 2020).

[97] Darren B. Taichman, Joyce Backus, Christopher Baethge, Howard Bauchner, Peter W. de Leeuw, Jeffrey M. Drazen, John Fletcher, Frank Frizelle, Trish Groves, Abraham Haileamlak, Astrid James, Christine Laine, Larry Peiperl, Anja Pinborg, Peush Sahni, and Si Nan Wu. Sharing Clinical Trial Data: A Proposal from the International Committee of Medical Journal Editors. *Chinese medical journal*, 129(2):127–128, 2016.

[98] Bruno Tavares, Filipe Figueiredo Correia, and André Restivo. A survey on blockchain technologies and research. *Journal of Information*, 14(2019):118–128, 2019.

[99] Bruno Tavares, Filipe Figueiredo Correia, and André Restivo. Trusted data transformation with blockchain technology in open data. In *International Symposium on Distributed Computing and Artificial Intelligence*, pages 213–216. Springer, 2019.

[100] Bruno Tavares, Filipe Figueiredo Correia, André Restivo, João Pascoal Faria, and Ademar Aguiar. A survey of blockchain frameworks and applications. In *International Conference on Soft Computing and Pattern Recognition*, pages 308–317. Springer, 2018.

[101] Parity Technologies. Polkadot Consensus, 2020. Available at https://wiki.polkadot.network/docs/en/learn-consensus (accessed at June 2020).

[102] Feng Tian. An agri-food supply chain traceability system for China based on RFID & blockchain technology. *2016 13th International Conference on Service Systems and Service Management, ICSSSM 2016*, 2016.

[103] European Union. *The EU General Data Protection Regulation (GDPR)*. 2018. Available at https://eur-lex.europa.eu/ (accessed at February 2020).

[104] Alexander Uskov, Adam Byerly, and Colleen Heinemann. Advanced encryption standard analysis with multimedia data on Intel® AES-NI architecture. *International Journal of Computer Science and Applications*, 13(2):89–105, 2016.

[105] Jason Anthony Vander Heiden, Susanna Marquez, Nishanth Marthandan, Syed Ahmad Chan Bukhari, Christian E Busse, Brian Corrie, Uri Hershberg, Steven H Kleinstein, IV Matsen, A Frederick, et al. Airr community standardized representations for annotated immune repertoires. *Frontiers in immunology*, 9:2206, 2018.

[106] Varadha Pally Vinay Reddy. Enhancing supply chain management using blockchain technology. *International Journal of Engineering and Advanced Technology*, 8(6):4657–4661, 2019.

[107] Harald Vranken. Sustainability of bitcoin and blockchains. *Current Opinion in Environmental Sustainability*, 28:1–9, 10 2017.

[108] Elissa R. Weitzman, Liljana Kaci, and Kenneth D. Mandl. Sharing medical data for health research: The early personal health record experience. *Journal of Medical Internet Research*, 12(2), 2010.

[109] Erik Westrup and Fredrik Pettersson. *Using the Go Programming Language in Practice*. PhD thesis, 06 2014.

[110] Paul Wicks, Michael Massagli, Jeana Frost, Catherine Brownstein, Sally Okun, Timothy Vaughan, Richard Bradley, and James Heywood. Sharing health data for better outcomes on patientslikeme. *J Med Internet Res*, 12(2):e19, Jun 2010.

[111] Arthur L. Wilson and Elisabeth R. Hayes. From the editors. *Adult Education Quarterly*, 51(1):5–8, 2000.

[112] Sifah E. B. Asamoah K. O. Gao J. Du X. Guizani M. Xia, Q. I. MeDShare : Trust-less Medical Data Sharing Among. *IEEE Access*, 5:1–10, 2017.

[113] Rui Zhang, Rui Xue, and Ling Liu. Security and privacy on blockchain. *ACM Computing Surveys*, 52(3), 2019.

[114] Guy Zyskind, Oz Nathan, and Alex Sandy Pentland. Decentralizing privacy: Using blockchain to protect personal data. *Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015*, pages 180–184, 2015.

Appendix A

Interview Questions

This appendix contains the questions for the interviews with experts to evaluate the feasibility and the benefits of the contributions developed. For more information about the validation and interviews, please refer to Chapter 6 (p. 87).

Since one of our objectives was to explore the field of health data research in order to understand the current problems in the ecosystem, we prepared an initial set of questions that evaluate our understanding of the current issues. Furthermore, we also evaluate the feasibility and the benefits of the solution developed, explained in Chapter 5 (p. 67), so we have prepared a second set of questions in order to provide an evaluation, based on expert opinion.

Below, there are five sections, each one with a different set of questions. The two first sections (A.1 and A.2) present the two first sets of questions, that aim to evaluate our understanding of the current issues. The three last sections (A.3, A.4 and A.5) present the three last sets of questions, to evaluate the feasibility and the benefits of the solution developed.

A.1 Conflicting Interests and Competition (CI)

- Are there conflicting interests of the entities (data processors and data repositories) in the current system?
  - Which conflicting interests do entities have?
- Are the entities in the system competitors of each other?
  - What do they want to achieve first?
- Do you think that this competition leads to secrecy of internal procedures? Could it also lead to dishonesty?
  - Can it incentivise entities to use other entities' data without giving them credit? And without properly licensing it?
- Do you think that this competition leads to secrecy of the produced data?

  - May this happen for fear of not being given credit?
  - Is that a problem? Are entities frustrated with it in the sense that they know it holds back their progress?

- Would entities be more willing to provide data to other entities if they could trust others wouldn't use their data without giving them credit?
  - Could they make more progress if they provide the data to other entities and, therefore, are provided data by other entities?
  - Would entities provide data to other entities if they were also provided data by them?
  - Would entities provide their data to data processors if they can use their computational resources?

A.2 Process of Building Trust (BT)

- Are data processor entities currently unsure about what processing procedures (algorithms, configuration parameters and implementation) other processor entities take?
  - What are the consequences of it?
- Would knowing the processing steps of an entity build trust towards the datasets it produces?
  - How is this trust currently built?
  - Does that trust make other entities willing to collaborate with that entity (the one that has built trust)?
- Is sharing processing steps details for public datasets a feasible mechanism of evaluating an entity's processing procedures (in order to build trust on that entity)?
  - Are entities interested in sharing processing steps details for public datasets so that other entities are more willing to collaborate with them?
- Which other mechanisms of building trust could be used?
- Would the entities involved be interested in licensing the data?
  - Would data owners be interested in licensing the data (e.g., to pharma industries)?
  - Would pharma industries be interested in licensing data from data processor entities?

A.3 Feasibility and the benefits of the approach (FB)

- Does a decentralized database (blockchain, where all nodes agree on the data) that allows entities to register the processing results for public datasets provide a feasible mechanism of evaluating an entity's processing procedures (in order to build trust on that entity)?

  - Are entities interested in sharing these results and respective processing steps on the decentralized registry so that other entities are more willing to collaborate with them?

- Does providing a mechanism to quickly validate the entries of that registry (with less decentralization than data storage, but still decentralized, through the voting rounds) further improve solving the problem (without nodes having to run the transformations themselves, which is computationally expensive)?
- Does providing a mechanism to verify the information without trusting validators give more confidence over it (even though it takes a lot of computational effort, but it could also be done only in certain cases)?
- Does providing both worlds (by voting rounds and full validation) make the system more suitable for all use cases?

A.4 Benefits of the Solution for Patients (PT)

- Can the traceability of the transformations of health data give patients more confidence over the data sharing system?
- Can the traceability of the transformations help rewarding patients that have contributed to help reach important breakthroughs in disease treatment so that they are more willing to provide their health data?

A.5 Feasibility of the Rewards System (RS)

- Is reputation a good incentive for entities to be honest in the process of voting for health data?
- Does the loss of reputation provide a disincentive for being dishonest, by thinking that, if the majority is honest, their reputation will be lost?
- Is gaining reputation a good incentive for verifying other's transformations data, given that it is a computationally expensive process?
- Is fear of losing reputation enough to discourage entities from submitting unvalidated votes (without going through the processing)?
- Is losing reputation enough to discourage entities from submitting unvalidated votes (without going through the processing)?