Polytechnic of Leiria
School of Technology and Management
Department of Computer Engineering
Master in Computer Engineering - Mobile Computing

IMMUNITY PASSPORT LEDGER

MARCO VERISSIMO OLIVEIRA

Leiria, November 2021

Polytechnic of Leiria
School of Technology and Management
Department of Computer Engineering
Master in Computer Engineering - Mobile Computing

IMMUNITY PASSPORT LEDGER

MARCO VERISSIMO OLIVEIRA
Number: 2192406

Project carried out under the guidance of Professor Catarina I. Reis, PhD (catarina.reis@ipleiria.pt) and Professor Marisa Maximiano, PhD (marisa.maximiano@ipleiria.pt).

Leiria, November 2021

ACKNOWLEDGMENTS

I take this section to express my gratitude to all those who, in some way, supported me during my academic journey and helped me to complete yet another stage of my education.

I would like to thank my teachers and advisors, Catarina I. Reis and Marisa Maximiano, for the technological support, feedback during the prototype's development, and unconditional support, feedback, and motivation during the writing of this document.

This project is also dedicated to my girlfriend Beatriz Baptista, who encouraged me to join this master's degree and supported me throughout this journey, especially during the writing of this work.

I also thank my family for teaching me the values and providing me with the necessary conditions to achieve this goal, especially my parents and sister.

A big thank you to everyone.

INTRODUCTORY NOTE

The work presented in this thesis was carried out in the School of Technology and Management of the Polytechnic of Leiria and resulted in the following publications accepted at the 11th World Congress on Information and Communication Technologies:

Oliveira, M., Honório, T., Reis, C. I., Maximiano, M. (2021). Immunity Passport Ledger: Digital certificates implemented on a permissioned blockchain

Honório, T., Oliveira, M., Reis, C. I., Maximiano, M. (2021). Building trust with a contact tracing application in a COVID-19 epoch: a blockchain approach

All the source code developed under this project can be found at the following GitHub repository: https://github.com/marcovoliveira/Immunity-Passport-Ledger.

ABSTRACT

The global outbreak of Coronavirus (SARS-CoV-2), which in 2020 reached pandemic scale, has been a central topic of debate in our society. Concerns over the ease of transmission of the infection led to the imposition of measures restricting freedom such as curfews, lockdown, general confinement, and trade closure.

Technology was one of the tools used to resist the spread of the disease using applications that, on the one hand, track contacts to warn users that were close to someone infected and, on the other hand, provide immunity digital certification. Despite the relevance of these options, end users have no confidence, transparency, and responsibility that the registration and use of their health data are ethical, secure, anonymous, and available through verifiable credentials. Most importantly, it is being used for its primary purpose.

Consequently, a solution based on distributed ledger technology, such as blockchain, is introduced in this work to assure the trustworthiness and integrity of users' data. Since the proposed work embraced user privacy, we conducted a comparative study between blockchains, focusing on permissioned blockchain that includes an authorization abstraction layer and ensures that identifiable participants can only perform specific actions. This work concluded that Hyperledger Fabric was an option that fulfilled all the requirements to develop a platform for the immunity passport ledger. Its modularity and versatility accommodate the requirements for the development of a prototype. This work describes the architecture used in the developed prototype, as well as the blockchain and applications that compose the system.

Our proof of concept provides an architecture that can be used to further study scaling , load balancing and performance capabilities.

Keywords: Blockchain, Immunity Passport Ledger, Mobile Applications, Web Applications, Hyperledger Fabric

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACL	Access Control Lists.
API	Application Programming Interface.
BFT	Byzantine Fault Tolerance.
CA	Certificate Authority.
dApps	Decentralized Applications.
DeFi	Decentralized Finances.
DLT	Distributed Ledger Technologies.
EVM	Ethereum Virtual Machine.
GDPR	General Data Protection Regulation.
gRPC	Remote Procedure Call.
HTTP	Hyper Text Transfer Protocol.
HTTPS	Hyper Text Transfer Protocol Secure.
IDHP	IBM Digital Health Pass.
JSON	JavaScript Object Notation.
NFT	Non-Fungible Token.
NP	Network Policies.

PCR	Polymerase Chain Reaction.
PDC	Private Date Collection.
PoET	Proof-of-Elapsed-Time.
PoS	Proof-of-Stake.
PoW	Proof-of-Work.
REST	Representational State Transfer.
SDK	Software Development Kit.
SGX	Intel Software Guard Extensions.
SPA	Single-page Application.
TLS	Transport Layer Security.
TPS	Transactions per Second.
WHO	World Health Organization.

1 INTRODUCTION

The Covid-19 pandemic situation raised awareness and the need to address privacy concerns during a public health situation. Namely the question of balancing the boundaries between privacy issues and the impact that this information has directly on a public health issue such as a pandemic.

1.1 MOTIVATION

On December 31, 2019, the first case of a new type of coronavirus was reported in Wuhan, China. On January 30, 2020, World Health Organization (WHO) declared coronavirus outbreak a public health emergency of international concern. On March 11, 2020, the outbreak reached a pandemic scale, with 118 000 cases reported in 114 countries (WHO, 2020). Since then, there has been an ongoing debate about how to best overcome the pandemic. Concerns about the ease of spread of the infection led to the imposition of measures restricting freedom such as lockdowns, curfews, general confinement, and trade closure.

Technology was one of the tools used to resist the transmission of the disease through applications such as Stayaway Covid (INESCTEC, 2020) that tracks contacts and notifies users that they were close to someone infected.

These applications raised concerns about user's data privacy and some countries were considering making these applications mandatory, which harmed their adoption by the population (Miranda Ramos, 2020).

In December 2020 a global scale vaccination was started and a return to normality is expected. One of the options to help in this situation of "back to normal" was the creation of immunity passports that will allow people to prove their health status. This option was questioned and even contested by the World Health Organization (Zhao et al., 2020).

We aim to solve these problems using Distributed Ledger Technologies (DLT) (Frankenfield, 2018) to develop an Immunity Passport Ledger prepared to prove vaccination status. It is imperative to present an Immunity Passport that is trans-

parent and secure and in which end users can trust, where the data is anonymous and used only for its purpose.

1.2 OBJECTIVES AND CONTRIBUTION

The proposed objectives for this work are to carry out a study of current blockchain technologies, understand the current panorama of these technologies and maturity, and provide the reader with an introduction to this technology.

Another objective of the work is to compare and analyze related works currently under development or finalized. This analysis makes it possible to find flaws and improvement points to be addressed by the prototype developed in this work.

The ultimate goal is to develop and present a functional prototype that implements blockchain technology to register and validate immunity certificates.

These objectives can be summarized in the following list:

- O1 - Study blockchain and distributed ledger technologies regarding their technological maturity;
- O2 - Analyze work related to immunity certificates with a special focus on flaws and improvement points;
- O3 - Design an architecture that allows immunity certificate registration and validation;
- O4 - Implement a proof of concept of the proposed architecture.

This project contributions comprise the development of a prototype of an Immunity Passport Ledger, which allows users to present their immunity certificates without providing more data than what is strictly necessary to prove their rightful ownership. The proposed architecture in this project leverages blockchain technology to create a permissioned consortium that allows the interoperability of certificates between various organizations. The architecture proposed in this project is validated by implementing a prototype.

Another academic contribution of this project is the publication of two papers at the 11th World Congress on Information and Communication Technologies. The first one describes the implementation of certificates on a permissioned blockchain. The second paper describes a blockchain approach to contact tracing applications.

This project was also presented at IEEE@Home Blockchain Series 2021: Blockchain and Healthcare Webinar.

1.3 STRUCTURE OF THE DOCUMENT

To fulfill the objectives presented in the previous section, this document describes the work conducted, organized as follows:

Chapter 2 provides a theoretical framework and presents an overview of Distributed Ledger Technologies. This chapter divides into different blockchain technologies, and a technological comparison is presented on the last section.

Chapter 3 presents three sections. The first section reviews and analyzes the related work based on a centralized architecture. The second section does this same review and analyzes projects based on a decentralized architecture. The third and last section compares the related work presented in the last two sections.

Chapter 4 explains the proposed architecture for the Immunity Passport Ledger prototype. The first section provides the requirements elicitation that describes the features and technical aspects that the system should deliver. The second section provides the rationale and the selection of technology. Afterward, a system context overview and the interactions with the actors are presented. Finally, the last section provides an architectural description focused on technical specifications.

Chapter 5 describes the process of development and implementation of the prototype. Section 1 illustrates the steps of implementation of Hyperledger Fabric Blockchain. Section 2 and 3 explain the functionalities of the application and present the interface.

Chapter 6 presents the project's conclusions by explaining how the objectives have been accomplished and proposing considerations for future work.

2

BACKGROUND

This section present the technological background for the document and work developed. It provides a better understanding of the topics discussed in this work.

Distributed Ledger Technology (DLT) can be defined by the technological infrastructure and protocols that support concurrent access, validation, and state updating in an immutable manner across a network that's spread across multiple participants or nodes (Frankenfield, 2018).

A DLT system is characterized as a system with multiple participants which reach a settlement over a set of distributed data and its validity, in the absence of a central authority. What separates a DLT system from a traditional distributed database are the core features capable of transacting data and maintaining data integrity in the presence of malicious actors actively attempting to attack the network (Rauchs et al., 2018).

DLT has great potential to disrupt the way governments, institutions, and corporations work. It can help governments with tax collection, the issuance of passports, recording land registries and licenses, and the outlay of Social Security benefits and voting procedures. Industries such as finance, music and entertainment, diamond and other precious assets, art, and supply chains of various commodities are the early adopters of this technology (McLean and Deane-Johns, 2016).

DLT comes in different architectures such as DAG (Saad and Park, 2019), Hashgraph (Baird et al., 2018), Holochain (Eric Harris-Braun, 2018), Radix (Cäsar et al., 2020), and Blockchain (Satoshi Nakamoto, 2008). Big corporations such as IBM and Microsoft are exploring blockchain, and the most popular protocols include Ethereum, Hyperledger Fabric, R3 Corda, and Quorum (Frankenfield, 2018).

2.1 BLOCKCHAIN

Blockchain is a DLT architecture where transaction records are registered and kept in the ledger as a chain of blocks. This technology is attracting a great deal of attention propelled by the success of Bitcoin, launched in 2009 and triggering a large number

of projects in different industries, with finance being the one that leads the use of this technology due to the success of cryptocurrencies. This technology underlies Bitcoin and has the potential to support a wide variety of business processes (Velde, 2013).

2.1.1 *Permissioned and Permissionless Blockchain*

Permissioned and private Blockchain differ from public Blockchains of Bitcoin and Ethereum. Enterprise Blockchain applications rely on trust relationships between participant organizations, with the need to share data with a greater degree of security. Permissioned blockchains can be used to record promises, trades, transactions, or any data that can't be lost without needing to run a Proof-of-Work mechanism (Androulaki, Barger, et al., 2018).

All organizations have a copy of the ledger identical to others in the network, and nothing can ever be erased or edited and can only be accessible by authenticated participants (Valenta and Sandner, 2017).

2.2 BITCOIN

The Bitcoin paper was written by the unknown author Satoshi Nakamoto, presented in late 2008, and proposes an alternative approach to online payments without dependence on intermediary financial institutions. This approach proposes a solution to the double-spending problem using a peer-to-peer network without a centralized validation system. To solve the double-spending problem, Bitcoin relies on a ledger known as a blockchain, which provides a way for all full nodes to know about all transactions (Satoshi Nakamoto, 2008).

In the Blockchain, all transactions are publicly announced to all nodes, which can then agree in the order in which they were received. If most nodes agree with that order, double-spending attempts will not be accepted. This method uses a timestamp server that receives the hash of a block of transactions and then broadcasts this hash to all the nodes in the network. As each timestamp includes the previous timestamp in its hash, this forms an immutable and publicly verifiable chain of transaction (Satoshi Nakamoto, 2008; Reid and Harrigan, 2013).

Bitcoins are recorded as transactions. There is no concept of withholding but participation in a publicly verifiable transaction event that proves the possession of

a bitcoin. Due to this fact, each bitcoin individually can be easily traced through all the transactions in which it was used until the beginning of its circulation.

Sending and receiving bitcoins requires having access to a pair of public and private keys associated with that bitcoin amount. The public key, directly related to the bitcoin address, is a random sequence of numbers and letters and is comparable to a username or email address, is safe to share, and is needed to receive bitcoins.

As an example of a transaction, we can assume that Marco wants to send some bitcoins to Pedro. To do this, he uses his private key to sign a message with the transaction details for the Blockchain. This message contains an input with Marco's public address, the number of bitcoins that he transferred, and and output with Pedro's public address, as exemplified in Figure 1 (Satoshi Nakamoto, 2008). This message is then broadcast on the Bitcoin network, where they are added to a new block that is later broadcast to all network nodes, that validate and accept it, if valid, and add it to the blockchain.

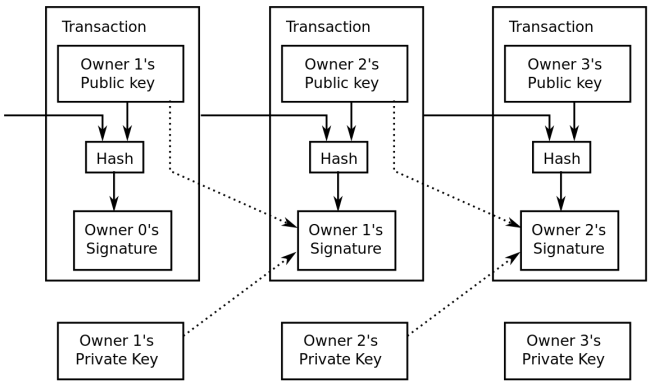

Figure 1: Bitcoin transaction (Satoshi Nakamoto, 2008)

Nodes may deliberately and maliciously distort or contradict information or propagate information to make it impossible to unify the network. This problem is known in the literature as the problem of the Byzantine Generals (Lamport et al., 1982). It occurs specifically in the case of Bitcoin when the entire network needs to agree on the secure and irreversible validity of transactions.

Bitcoin provides the solution to this problem using the consensus algorithm Proof-of-Work (PoW)(Liu and Camp, 2006) which is time-consuming and computationally expensive to produce the new blocks, but which is fast and requires little computational power to verify, if it is correct.

For blocks to be accepted on the network, miner nodes need to perform a computational problem that consists of guessing a random number that corresponds to the right hash for the transaction block.

This process (mining) uses the hash of the previous block, the current transaction block, and a 'nonce' (random integer) that is added to the end of the block. Then, it is necessary to start the calculations using a cryptographic algorithm such as SHA-256. This process consists of changing the nonce until obtaining a sequence that has a given number of zeros at the beginning (that corresponds to the difficulty appointed for the block.

The first node to reach a solution receives a reward and the transaction fees. This system encourages the nodes with high throughput, have a greater reward for working honestly than trying to defraud the network. All nodes trust the longest chain. If anyone wants to tamper with the blockchain, he needs to control more than 50% of the world's hashing power to ensure that it can become the first one to generate the latest block and master the longest chain. The costs from tampering can be much greater than the gains. So the PoW mechanism can effectively guarantee the safety of the blockchain (Du et al., 2017).

The Bitcoin Blockchain structure is composed of several blocks arranged in a linear sequence over time that contains transaction records, and a reference to the previous block.

New transactions are constantly being created and processed for new blocks that are added to the end of the blockchain and after being accepted by the network they become immutable (B. Lee and J. H. Lee, 2017; Vaidya, 2016).

Figure 2 represents the elements that constitute each block:

- Magic number: it is always 0xD9B4BEF9 and serves to identify a message on the Bitcoin network.
- Block Size: is the size in bytes of the block and is added at the end.
- Version: consists of the current version of the blockchain at the time of block creation.
- Previous block hash: refers to the previous block.
- Hash Merkle Root: is the hash of all hashes of all transactions that are part of a block.
- Timestamp: is the block creation date in milliseconds from January 1, 1970.

- Block difficulty: it is a numerical value of the difficulty of the proof of work process stipulated before the creation of the block.
- Nonce: is the numeric value used in *proof of work*.
- Number of transactions is the total number of transactions in the block.
- Transaction list: stores the digital finger-print of all the transactions in that block.

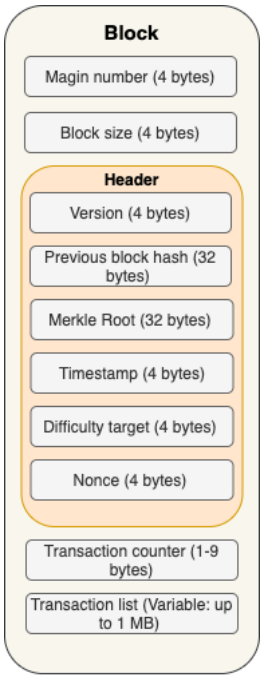

Figure 2: Bitcoin block structure

2.3 ETHEREUM

Ethereum was presented in 2014 and launched in 2015 by Vitalik Buterin (Buterin, 2014).

Ethereum uses Proof-of-Work as a consensus algorithm similar to Bitcoin, and the currency used is named Ether. This platform extends Blockchain concepts from Bitcoin, where transaction information is validated, stored, and replicated by the different nodes of the network. The main difference from Bitcoin is the possibility of executing code in a decentralized way for smart contracts.

A smart contract is simply a program that runs on the Ethereum virtual machine. Smart contracts are a type of account, which means that they have an associated

amount and can carry out transactions on the network. The difference being that a user does not carry out the control.

After the development, smart contracts are stored in the blockchain. The interaction is done through submitted transactions that perform a function defined in the smart contract. The Solidity code, where the smart contract logic is programmed, is compiled by the Ethereum compiler to byte-code and executed in the Ethereum Virtual Machine in an isolated environment defined by blocks (Antonopoulos and Wood, 2018; Idelberger et al., 2016).

A smart contract needs to have 3 characteristics:

- Deterministic
- Terminable
- Isolated

A program is deterministic if it returns the same result for a given input every time.

It must be terminable using a certain amount of resources in a specific time. A lifetime is defined in the form of Gas that refers to the execution fee, and when it reaches the limit of Gas, all operations are terminated (Prates and Gast, 2019).

Smart contracts, by definition, need to expire within a certain time. There are some procedures to ensure that Smart Contracts can be terminated and do not enter an infinite loop that would consume many resources:

- Quasi–Turing-Complete state machine: All execution processes are limited to a finite number of computational steps by the amount of Gas available for any given smart contract execution. Fee meter contracts are executed with a pre-paid fee. The execution of each instruction corresponds to the payment of a fee. If this pre-paid amount is exceeded, the contract is terminated.
- Step and Fee meter: A program can keep track of the number of steps it has already executed and finished, after a number of steps have already been executed.
- Timer: If the contract execution exceeds a predefined time, the execution is terminated.

Smart contracts are executed in an isolated environment in the EVM. When a bug or hack exists in a smart contract, that does not affect the entire Ethereum ecosystem (*Ethereum development documentation | ethereum.org* n.d.).

2.3.1 *Decentralized applications (dApps)*

They are similar to regular applications but are built from smart contracts code that automatically executes the terms of a contract. Smart contracts work like the business logic of a standard application, and the blockchain of Ethereum is used as the application's database. dApps can be run by every single node in the network since they will all execute the same code that resides in the single source of truth (blockchain).

After performing the transaction that registers the smart contract into the blockchain, it becomes immutable by default, making users not dependent on third parties to enforce the rules (Antonopoulos and Wood, 2018).

Some of the advantages of dApps are:

- They have no owners, and once implemented on the chain, they cannot be removed, and everyone can use the same features. Even if the team that created the application ceases to exist, its use is still possible.
- Free from censorship, meaning no one can be blocked in terms of using the application.
- Anonymous login, so it is not necessary to share personal information, just an Ethereum account, and a wallet.
- No downtime. Once the application is on *Ethereum*, the only way the application can go down is for the Ethereum itself to go down, and networks the size of Ethereum are complex to attack.

Some examples of decentralized applications:

- OpenSea (*OpenSea, the largest NFT marketplace* n.d.) - Non-fungible Tokens (NFT) marketplace.
- PeepEth (*Peepeth* n.d.) - Social network.
- Aave (*Aave – Open Source DeFi Protocol* n.d.) - Loans and deposits, Decentralized finance (DeFi).

2.3.2 *Ethereum 2.0*

Ethereum 2.0, also known as Eth2 or Ethereum Serenity, is the next update to the Ethereum platform. This update aims to introduce a new Proof of Stake consensus mechanism.

Beacon Chain is the name given to the first phase of migration to Eth2. This phase introduces the Proof of Stake consensus mechanism. Instead of relying on computational power and electricity to validate new blocks in the blockchain like the Proof of Work, which depends on validators that deposit an escrow of 32 Ether, validators are randomly selected among all validators. They have the opportunity to create the next block that everyone will check. If a validator successfully validates a block or testifies to an already validated block receives a reward. On the other hand, if it tries to compromise the continuation of the blockchain, the collateral is lost in part or whole. This system allows for greater security and scalability.

Another significant improvement in Ethereum 2.0 is the introduction of shard chains. This scalability mechanism will dramatically improve throughput using 64 blockchains in the second phase of the migration process. Currently, the Ethereum blockchain is composed of a single blockchain, so this new improvement aims to split the blockchain and divide the processing responsibility by different nodes, leading to processing transactions in parallel, pointing to a drastic decrease in transaction time (Ethereum, 2021).

In the last transition phase, the docking process occurs. The mainnet network currently used by Ethereum and the network created in the first phase of Eth2 merge into one. Furthermore, the current blockchain becomes one of the 64 shards.

2.4 HYPERLEDGER

Hyperledger (*Hyperledger – Open Source Blockchain Technologies* n.d.), launched in 2016, is an industry-wide open-source initiative to advance blockchain technology, governed by The Linux Foundation. Hyperledger incubates and promotes a range of business blockchain technologies, including distributed ledger frameworks, tools, and libraries for enterprise-grade blockchain deployments (*Hyperledger - Hyperledger - Hyperledger Confluence* n.d.).

2.4.1 *Hyperledger Sawtooth*

Hyperledger Sawtooth is an enterprise solution for building, deploying, and running distributed ledgers, also known as blockchains. Sawtooth provides an extremely modular and flexible platform for implementing transaction-based updates to shared states between untrusted parties through consensus algorithms (Hyperledger Foundation, 2020).

The architecture of a Sawtooth application (see Figure 3), is composed of the following elements (*Architecture Guide — Sawtooth v1.2.6 documentation* n.d.):

- Data Model - Defines valid operations and transaction size.
- Transaction Processor - Defines the application's business logic, validates the transactions, and updates the status based on the rules defined in the application. This component runs on all application nodes.
- Client - Defines the client application logic, generates and sends transactions to the Sawtooth Validator, and presents the information to the user.
- REST API - Communication between the Client and the Transaction Processor. This can be customized, or the existing one can be used.

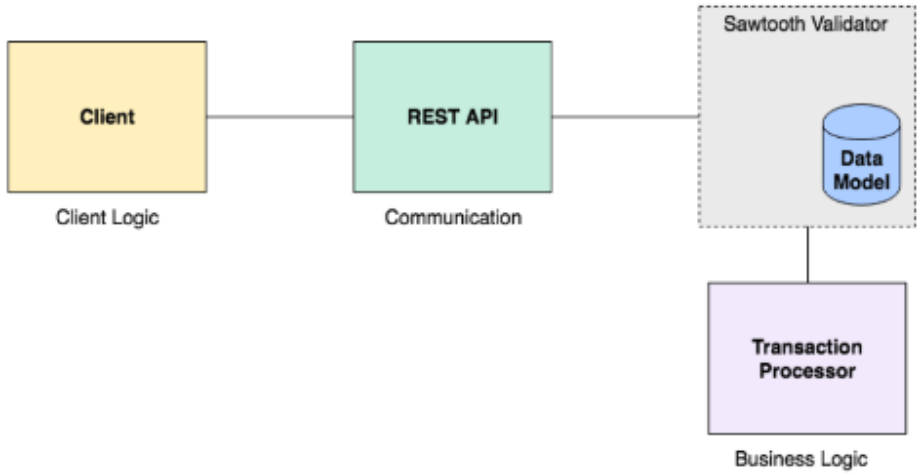

Figure 3: Sawtooth Architecture (*Architecture Guide — Sawtooth v1.2.6 documentation* n.d.)

In Sawtooth, nodes communicate with each other via messages (ØMQ) over TCP connections. These messages contain transaction information, blocks, and peers. All messages structures are serialized by Google's Protocol Buffers. Currently, Sawtooth relies on a gossip protocol to ensure that all messages about transactions and blocks are delivered to all nodes on the network.

In order to meet application needs, Sawtooth supports network permissions at the peer-to-peer layer. Through access control lists, Sawtooth nodes are able to control:

- Who can connect to the network and get the current state of the blockchain.
- Who can receive the consensus messages and participate in their process.
- Who can submit transactions on the network.

One of the fundamental components of a blockchain and state machines is ensuring that all nodes in the network process transactions in the correct order. Sawtooth uses the blockchain data structure to ensure the correct order of blocks linked together, and where each block contains a list of ordered transactions that is agreed by the nodes in the network.

While Bitcoin uses a proof-of-work consensus system, Hyperledger Sawtooth allows you to choose which consensus system you want to use. One of the most common is proof-of-elapsed-time (PoET).

This consensus system is based on the creation of a random number that represents the duration a node is inactive. The node with the lowest inactivity time is the first to validate the new block in the blockchain.

For the correct functioning of the system, it is necessary to guarantee that the participants wait a random amount of time and that the first node to wake up was not dishonest. This is possible due to a special set of CPU instructions called Intel Software Guard Extensions (SGX).

SGX allows applications to run trusted code in a protected environment while maintaining a fair consensus system (Shi et al., 2019).

2.4.2 *Hyperledger Fabric*

Hyperledger Fabric is an enterprise-grade open-source permissioned blockchain framework for developing solutions and applications with a modular architecture. Its modular and versatile design satisfies a broad range of industry use cases, allowing components, such as consensus and membership services, to be plug-and-play (Uddin, 2021; Spengler and Souza, 2021; Pajooh et al., 2021). The unique approach to the consensus mechanism enables performance at scale while preserving privacy that enables the flexibility and scalability needed for enterprise-grade applications (Chacko et al., 2021).

Consensus mechanism has a fundamental role in the transaction flow of Fabric that goes through the process of a transaction proposal, endorsement, ordering, validation, and commitment.

During the lifecycle of a transaction, all the policy rules are checked and enforced to guarantee the members are allowed to perform the action, and payloads are repeatedly signed, verified, and authenticated as a transaction proposal passes through the different architectural components.

The ordering of transactions is a modular component decoupled from the peers that execute the transaction and store the ledger. Implementation of the ordering service can be adapted to meet the needs of a specific solution. Fabric currently offers a crash fault-tolerant ordering service implementation based on the *etcd* library of the Raft protocol (Mohammed et al., 2021).

Fabric has been designed at its core to have a modular architecture and meet the diversity of enterprise use case requirements. In the previous section, we already addressed the flexibility in the consensus mechanism, but besides that, it has pluggable identity management protocols such as LDAP or OpenID Connect, ledger support a variety of DBMSs, and pluggable endorsement and validation policy enforcement (Mohammed et al., 2021).

Performance of Fabric has been further improved, during the release of new versions, with a substantial increase in performance on Fabric 2.0 (Dreyer et al., 2020).

The performance of a blockchain is subjected to many implementation variables such as transaction size, block size, network bandwidth, hardware limitations, consensus algorithm, caching, and parallelism (Nguyen et al., 2021; Xu et al., 2021).

Out of the box with simple configuration, Fabric can support 3000 transactions per second, and with advanced tweaks, a study (Gorenflo et al., 2020), demonstrates that it is possible to scale Fabric performance to 20 000 transactions per second.

2.5 TECHNOLOGY COMPARISON

The above presented technologies were picked because they are already considered mature in the blockchain field. Other options were discarded because they are currently embryonic, and the application requires a more mature technology.

A comparative table was created between the three blockchains, Table 1, to understand specificities of the technology and a future upcoming fit of the needs of the prototype.

Table 1: Blockchains comparative table

	Hyperledger Sawtooth	**Hyperledger Fabric**	**Ethereum**
Permission Level	Permisioned and Permissionless	Permissioned	Public
Consensus Algorithm	PoET, Byzantine Fault Tolerance Raft, Devmode	Kafka, Raft, Solo	PoW
Transaction Speed	∼1300 TPS	2000 to 20.000 TPS	13 to 20 TPS
SmartContracts	Yes	Yes	Yes
Governance	Linux Foundation	Linux Foundation	Ethereum Foundation
Launch	2018	2018	2015
Supported Smart Contract Language	Rust, JavaScript, Go, Python	Go, Java, Javascript, Solidity	Solidity
Currency	None	None	Ether
Contributors	79	294	675
State Storage	Radix Merkle Tree	CouchDB or LevelDB	Account data

3

RELATED WORK

In this chapter, concepts related to immunity passports are introduced. Previous and ongoing applications with related themes are considered, including works whose details help demonstrate the improved and advanced points in the developed proof of concept. Its limitations concerning the related works are also described below.

The recent COVID-19 outbreak leads to several public health concerns that lead to a worldwide lockdown. Meanwhile, a critical and related aspect that has taken as a significant concern was the economic and financial burden that a relatively slow return to work after the lockdown would carry to the global economy.

One of the options available to help in this "back-to-work" situation was the creation of immunity passports that will allow people to prove their health condition. This option was highly questioned and even refuted by the World Health Organization (World Health Organization, 2020).

Despite the relevance of this option, end users lack the trust, transparency, and accountability that the recording and usage of their health data are correct, safe, anonymous, and available through verifiable credentials. Most importantly, that it is being used for its primary purpose.

3.1 IMMUNITY PASSPORT SYSTEMS - CENTRALIZED ARCHITECTURE

3.1.1 *CommonHealth*

CommonHealth, shown in Figure 4 (*Managing Health Data | CommonHealth* n.d.), helps people collect and manage personal health data and share it with the healthcare services, organizations, and applications they trust.

It arose out of the need to offer an alternative to Apple Health for Android users. The later mobile operating system represents 55% of all mobile devices in the U.S.A and 85% worldwide.

This project is an open-source, non-profit public service. It was developed based on the new standards and data interoperability regulations. It was developed jointly with private and public health and technology entities and with the Rockefeller Foundation's support to build the digital age's public infrastructure.

This application offers the following features:

- View health record.
- Export health record.
- View records by date.
- View records by type.
- View new records.
- Encryption of records on the device.
- Share health data with other applications.

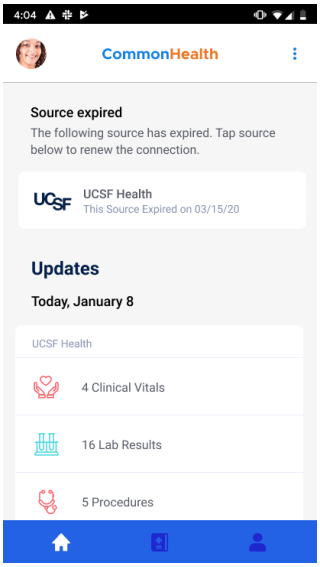

Figure 4: CommonHealth Application

3.1.2 *CommonPass*

The Common Project, the World Economic Forum, and a broad coalition of public and private partners are collaborating to launch CommonPass, shown in Figure 5 (*CommonPass | Digital Health App* n.d.). CommonPass aims to be a trusted and globally interoperable platform for people to document their status COVID-

19, health declarations, PCR tests, and vaccinations to satisfy in-country entry requirements, protecting health data privacy.

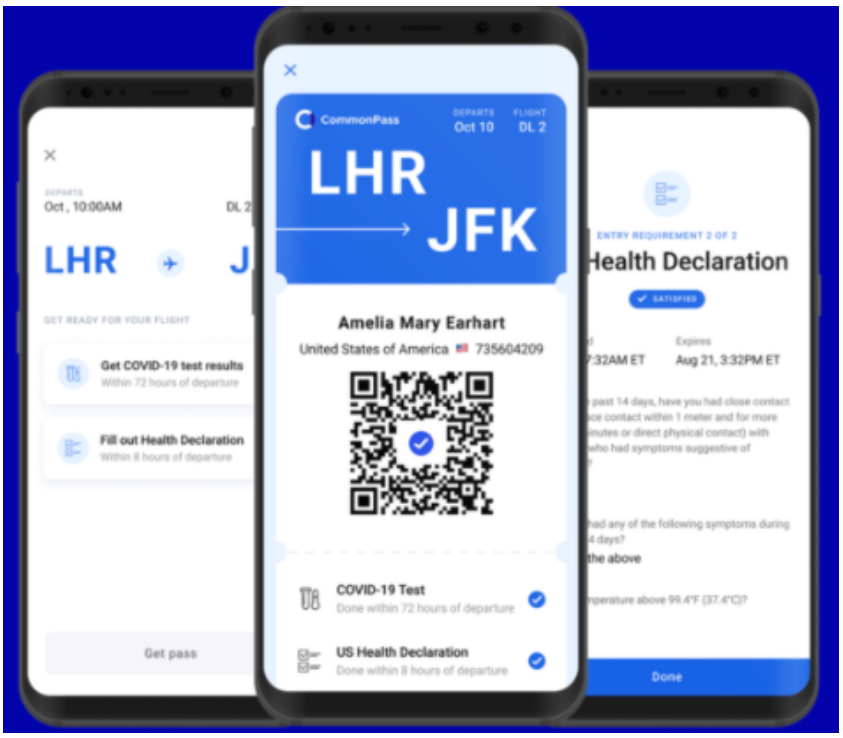

Figure 5: CommonPass Application (*CommonPass | Digital Health App* n.d.)

It allows users to access their laboratory results and vaccination records and consent to this information to validate their COVID status, without revealing any further health data.

Laboratory and vaccine results can be accessed through existing national or local information systems, or digital records such as Apple Health and CommonHealth.

The platform validates that laboratory and vaccine registration data comes from a verified source, with valid credentials and meets the health screening requirements of the country you wish to enter. If all conditions are met, a digital certificate is generated as outlined in Figure 6.

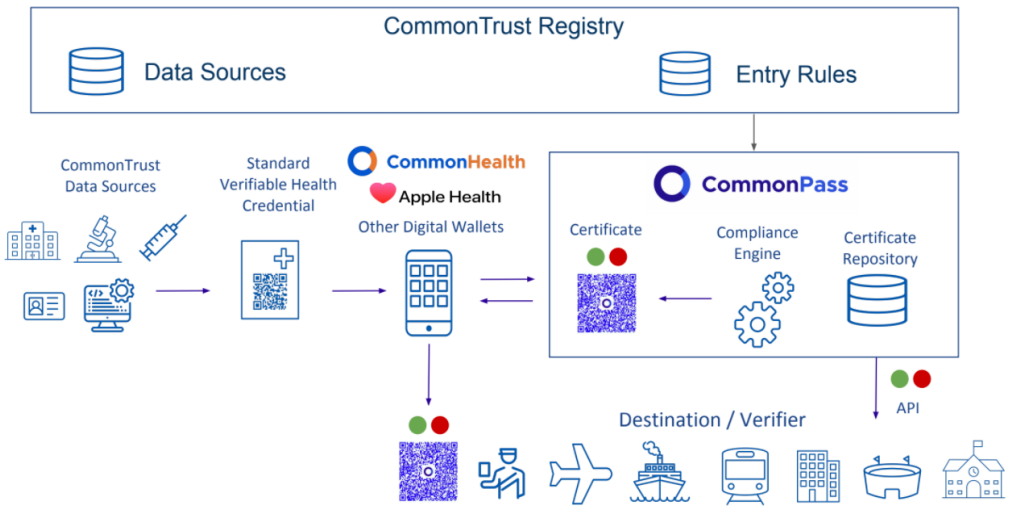

Figure 6: Functioning of CommonPass (*CommonPass | Digital Health App* n.d.)

Returns a yes/no answer if the user meets the current entry criteria, but the underlying health information remains under the user's control.

3.1.3 *Q-Wallet*

Q-Servi is a technology company dedicated to digital transformation in healthcare and develops solutions such as Q-Wallet (*Q-Wallet - Take Control Of Your Personal Records* n.d.).

Q-Wallet allows users to record their health data and medical records securely and privately. Personal information is only stored on the user's device. Any information provided by healthcare professionals is encrypted and references an anonymous identifier of a user.

Allows users to create test certificates that can be shared and authenticated using QR codes (see Figure 7), enables healthcare professionals to securely send information to the users' wallet where it is stored.

The user can select to manually allow the information they wish to share with third parties. This information is sent via a hash and is only valid once.

Figure 7: Q-Wallet application (*Q-Wallet - Take Control Of Your Personal Records* n.d.)

3.1.4 *Health Passport Worldwide*

Health Passport Worldwide is a digital platform developed in Ireland by the ROQU group for storing official vaccinations, PCR tests, and rapid tests (*Home | Health Passport Worldwide* n.d.). This system relies on partnerships with health clinics to enable medical professionals to interact with the systems and update users' test results on health passports and is available in 18 countries.

The application also provides a system to create a travel diary and event journal to trace contracts. Mobile application presents a user interface with a QR code that is verified by a partner application (see Figure 8).

Regarding vaccinations, the system doesn't yet provide integration with current vaccination certifications despite saying that it does.

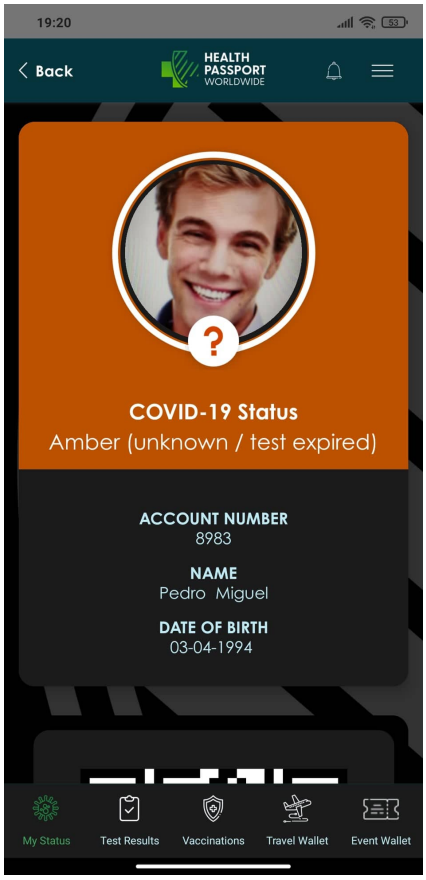

Figure 8: Health Passport Worldwide application (*Home | Health Passport Worldwide* n.d.)

3.1.5 *EU Digital Covid Certificate*

Digital green certificate is a collaborative effort of health care authorities across the EU and consists of a PKI solution with centralized infrastructure (*EU Digital COVID Certificate | European Commission* n.d.).

This digital Covid certificate solution is already a success worldwide, with more than 591 million certificates generated. This solution also set a global standard and is currently used in 43 countries across four continents, and more will join in the following months (*The EU Digital COVID Certificate: EU has set a standard* n.d.).

This system supports three types of certificates:

- Vaccination certificates
- Test certificates (PCR Test or rapid antigen test)
- And Covid recovery certificates

The certificates are issued in a digital format or on paper. Both contain a QR code that contains the information and a digital signature to ensure the certificate is authentic ("eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 1" 2021).

European authorities build a gateway and provide specifications to support all member states in developing software to verify all certificates signatures across the European Union ("eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 2 European Digital Green Certificate Gateway" 2021).

National backend implementations of each member state connect to this gateway to exchange the public keys used in certificate generations (see Figure 9).

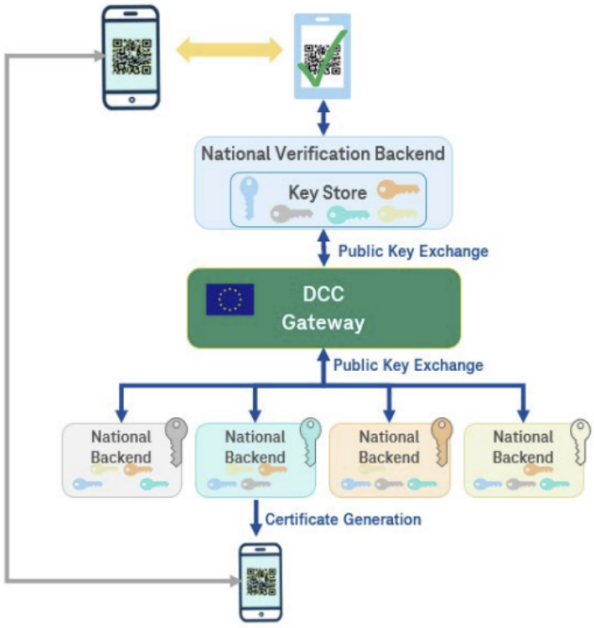

Figure 9: EU Digital Covid Certificate key exchange system ("eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 2 European Digital Green Certificate Gateway" 2021)

The personal data of the certificates don't pass through the gateway, and the QR information is verified locally on the mobile application.

Certificates include data such as: name, birth date, date of insurance, information about the vaccine, test or recovery, and a unique identifier of the certificate ("eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 4 European Digital Green Certificate Applications" 2021).

This solution is not ideal since the key pairs are not issued to single health care professionals, but rather big health organizations. A single breach of the private key will bring down every certificate signed by that authority with that key, as all of the certificates will be marked as untrusted, which potentially risks blocking traveling in Europe. Hackers are also targeting these keys because of their value to generate new certificates and sell them on the black market ("eHealth Network Guidelines on Technical Specifications for Digital COVID Certificates Volume 5 Public Key Certificate Governance" 2021).

Another issue with this implementation is the amount of information shared in QR Codes, such as the brand of vaccine taken. Malicious validator applications can be used to collect this information by reading certificates QR codes ("eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 3 Interoperable 2D Code" 2021).

3.1.6 *CoWIN*

CoWIN is the platform created by the Indian government to manage the appointment of vaccinations and the emission and validation of vaccine certificates. This solution is also based on a PKY system and allows offline verification. The infrastructure is centralized and under a unique central authority (*CoWIN* n.d.; ADB, 2018).

Indian government does not provide further details on the system specifications.

3.1.7 *IATA Travel Pass*

IATA Travel Pass is a mobile app that stores and manages verified certificates for tests or vaccines. It gives the user a digital wallet to store all their travel documentation, including its biometric passports. This application allows travelers to plan their journeys accordingly to the health conditions of the destination country (*IATA - Travel Pass Initiative* n.d.).

3.2 IMMUNITY PASSPORT SYSTEMS - DECENTRALIZED ARCHITECTURE - BLOCKCHAIN

3.2.1 *Covid-19 Health Passport*

SICPA is a Swiss security consultancy company. It provides inks for printing banknotes, documents, passports, lottery tickets, and recently certified digital documents using blockchain technology (*A COVID-19 health passport secured by blockchain to enable deconfinement | SICPA* n.d.).

In consortium with OpenHealth and GuardTime are developing the Covid-19 Health Passport to real-time monitor the population's immunity status.

Through the use of the KSI® Blockchain, data integrity is maintained, and it is possible to validate and present an immunity certificate that cannot be forged and is universally verifiable.

All personal information is anonymized and encrypted, as shown in Figure 10. This application is interoperable and works without a central database. The passport can be changed in real-time (created, expired, renewed, canceled) according to the medical tests based on the rules defined by the competent authorities.

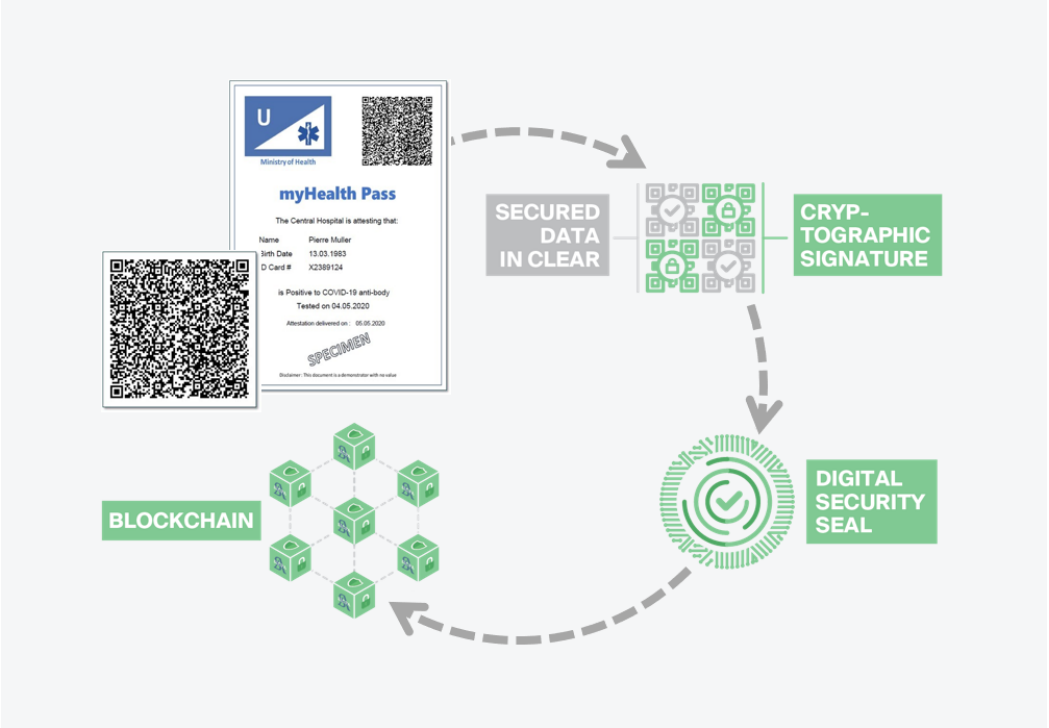

Figure 10: Covid-19 Health Passport system (*A COVID-19 health passport secured by blockchain to enable deconfinement | SICPA* n.d.)

It allows public authorities to control access to services and infrastructure, measure deconfinement's efficiency, and monitor the progress of group immunity in real-time. All data processed in the Sicpa application are under the GDPR.

3.2.2 *IBM Digital Health Pass*

IBM Digital Health Pass (IDHP) is a blockchain-based platform designed to enable organizations to verify health credentials for employees, customers, and visitors, such as test results and vaccination certificates (see Figure 11). It allows individuals to manage their information through an encrypted digital wallet on their smartphone and maintain control of what they share, with whom, and for what purpose (*Digital Health Pass | IBM* n.d.).

Digital Health Pass is part of IBM Watson Health. A suite of applications that uses data, Artificial Intelligence technology, and blockchain to help companies make decisions in three critical areas: workplace re-entry and facilities management, onsite security of work, and tracking of contacts.

IDHP leverages Hyperledger Fabric, an open-source platform whose architecture has been peer-reviewed by top-tier scientific conferences committees and adopted mainly by the enterprise DLT world. On the purely technical level, Hyperledger Fabric offers multiple advantages compared to other enterprises alternatives from scalability, performance, and governance perspective, and nicely integrates traditional public Key Infrastructure hierarchies (Androulaki, Circiumaru, et al., 2021).

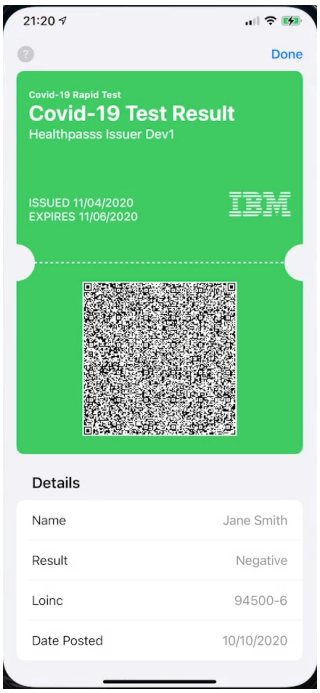

Figure 11: IBM Digital Health Pass application (*Digital Health Pass | IBM* n.d.)

3.2.3 *AOKpass*

AOKpass is a platform and mobile application using blockchain technology, enabling users to verify their health status with third parties while preserving the privacy of their underlying personal health data. Users have exclusive control over their health data, such as health certificates or test results, stored only on the user's mobile device and never on any external database or centralized system. AOKpass saves a hash on the Ethereum public blockchain network for the certificate data of the users (*AOKpass: Secure - Private - Portable* n.d.).

3.2.4 *Immunitee*

Immunitee Health Passport is the Malaysia health passport accepted into Singapore. The Immunitee Health Passport is designed to store personal immunization records and vaccine data with the Unifier platform, providing interoperability to securely share the necessary data with the various national health check systems being put globally (*Malaysia's Immunitee Health Passport gains Singaporean verification* 2021).

Immunitee stores all patient data hashed on a public blockchain system, ensuring that data cannot be adulterated, is protected, and belongs to the user. Validators can only access information by scanning a secure QR Code that contains all the relevant testing and vaccination information and can only be unlocked using a private key that belongs to the user.

3.3 COMPARATIVE ANALYSIS OF THE RELATED WORK

This section performs a comparative analysis of the related work presented in this chapter, considering the most important parameters in these applications.

Some applications on this table are not released yet, and others are not available for public testing.

A comparison is available in Table 5 - Annex A, to fit on a landscape page.

In summary, 4 of the solutions presented refer to the usage of blockchain architecture.

AOKPass uses the Ethereum public blockchain to save hashes of the users.

SCIPA uses a proprietary blockchain KSI®Blockchain to store digital security seals of health passes.

IBM Digital Health Pass uses Hyperledger Fabric blockchain to store information about certificate issuers.

Immunitee stores all patient data on a public blockchain system but does not provide more details about the implementation.

4

IMMUNITY PASSPORT LEDGER - ARCHITECTURE DESIGN

This chapter describes the proposed architecture for the Immunity Passport Ledger prototype.

The first section provides the requirements elicitation that describes the features and technical aspects that should be delivered by the system. The second section provides the rationale and the selection of technology. Afterwards, a system context overview and the interactions with the major actors is presented. Finally, the last section provides an architectural description focused on technical specifications.

4.1 REQUIREMENTS

In this section, it's performed an analysis and discovery of software requirements, functional and non-functional.

Functional requirements describe precisely what tasks the software must perform and help define the scope of the system. As for non-functional requirements, they describe the look and feel of the system, including visual properties, usability, performance, operating environment, and security aspects.

The methodology used for collecting requirements is using multiple index cards for each requirement. These informal and high-level requirements have a user story format and will evolve into a use case or numerous use cases.

Requirements are classified on a scale of priority from Low to High:

1. R01: As a user, I want to have a QR Code in digital/paper format so that I can present my immunity status.

   Priority: High

2. R02: As a user, I want my data to be anonymized so that I can keep my sensitive information safe.

   Priority: High

3. R03: As a user, I want to share only the essential data with the validator so that I keep my sensitive information private.

   Priority: High

4. R04: As a medical professional, I want to register user information so that I can generate immunity certificates.

   Priority: High

5. R05: As a validator, I want to scan QR codes so that I can verify immunity certificate validity.

   Priority: High

6. R06: As a validator, I want to use a mobile application to verify the certificate so that I can easily carry it with me.

   Priority: Medium

7. R07: As a validator, I want to see the unique health identification number after scanning the QR Code so that I can compare it to the user identification

   Priority: Medium

8. R08: As a user, I want my data not to be transmitted during the verification process so that I don't have my information collected.

   Priority: Medium

9. R09: As a statistic organization, I want to collect non-sensitive data so that I can perform metric analysis on community immunity status. Priority: Low

10. R10: As a statistic organization, I want to set alerts and receive updates on defined alerts so that I can follow immunity evolution.

    Priority: Low

These requirements' cards are organized in Trello, an online Kanban board tool where colors were used to visualize priority.

4.1.1 *Product backlog*

Creating the product backlog is the next step in the design process of the prototype. Product backlog integrates the requirements into a comprehensive package that describes the user's interactions with the system.

The product backlog is composed of items that represent a single element of work. These items include specifications, features, bugs, or changes.

Each item has a priority and an estimation of the effort for the implementation. This approach keeps the work organized and smooth to manage the development of all parts of the prototype application.

In Table 2, it is possible to observe the content of the product backlog at the beginning of the prototype development.

Table 2: Product backlog

Product Backlog ID	Description	Estimate	Priority
Item01	Infrastructure and network setup	Large	High
Item02	Setup mobile application for validators	Medium	High
Item03	Setup web application for health professionals	Large	High
Item04	Setup gateway for validators mobile application	Medium	High
Item05	Create certificate web application	Medium	High
Item06	Generate QR Code certificate	Small	High
Item07	Data anonymization in the web application	Medium	High
Item08	QR Code reader in mobile application	Medium	High
Item09	QR Code decoder in mobile application	Medium	High
Item10	Connect mobile application to data gateway	Medium	High
Item11	Handle gateway response on mobile application and present certificate data	Large	High
Item12	Develop gateway endpoint for read certificate	Large	High
Item12	Health professionals authentication web application	Small	Medium
Item13	Show user health number on the application after read QR code	Small	Medium
Item14	Data hashing with salt on connection to gateway	Small	Medium
Item15	Setup web application for statistic organizations	Large	Low
Item16	Send email with QR Code	Small	Low
Item17	Develop a graphical interface to visualize number of generated certificates in a timeline	Large	Low
Item18	Develop a custom query field to allow statistic personal to do custom requests do data source.	Large	Low

4.2 TECHNOLOGY SELECTION

As introduced in section 1.2, the objective of this project is to present an Immunity Passport that is transparent and secure and in which end users can trust, where the data is anonymous and used only for its purpose.

Considering these objectives, the technology choice must go towards a blockchain that provides a degree of permission that only allows verified users to execute write operations on the ledger.

Public Ethereum blockchain does not fit into requirements since it is a blockchain of public access. Anyone can connect, submit new transactions, and inspect the data on the ledger. As explained before, Proof-of-Work consensus mechanism is a computationally expensive task that requires high fees to process new transactions and slows the number of transactions capable of being processed in a time frame.

This factor represents a significant deal-breaker for using Ethereum as our blockchain and smart-contract platform.

The decision stood between the two Hyperledger technologies: Fabric or Sawtooth.

Considering the performance and scalability required for the application, both technologies are similar and allow a high number of transactions per second.

One aspect considered when choosing the technology was the number of people involved in the projects and the number of active contributors. This factor is vital as a project with more traction makes the development process more accessible and quicker to solve problems. In this category, Fabric takes the lead.

One of the requirements defined for the applications is the capacity for statistic organizations to perform metric analysis on ledger data. These requirements raise the need to use a state database that allows complex queries on data. Sawtooth relies on a Merkle radix data structure as the underlying data store. Hyperledger Fabric provides a LevelDB key-value storage, or CouchDB, a NoSQL database that supports complex JSON documents and complex queries.

Taking into account the factors presented, the choice of technology fell to Hyperledger Fabric.

4.3 SYSTEM OVERVIEW

This section presents a high-level system overview architecture of an Immunity Passport solution capable of storing data on a distributed ledger and making that data available that guarantees certificate authenticity and privacy.

The system context diagram allows seeing the big picture of a software system. This approach provides a context of the interaction between the software, its users, and other systems. Details are left to the next section, and the focus is shifted to actors and roles rather than technologies, protocols, and other low-level details. The objective of this section is to provide explained diagrams that could be shown and interpreted by non-technical people.

The proposed general system context overview is shown in Figure 12.

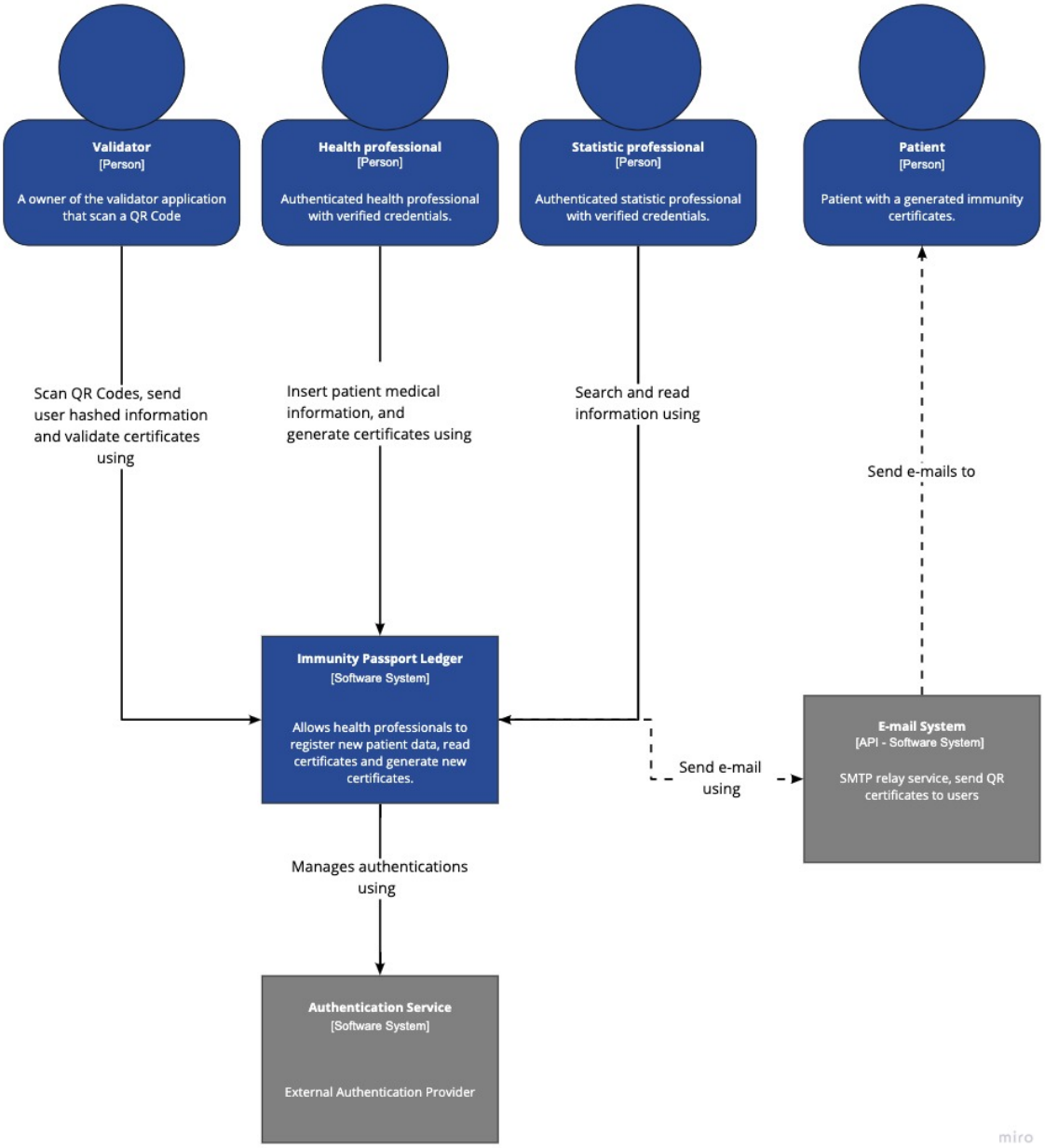

Figure 12: Proposed general system context overview

The diagram in Figure 12, provides a high-level overview of the Immunity Passport Ledger and its context.

Inside the Immunity Passport Ledger, Figure 13, accommodates the architecture of the general system container diagram with the applications that are part of the solution.

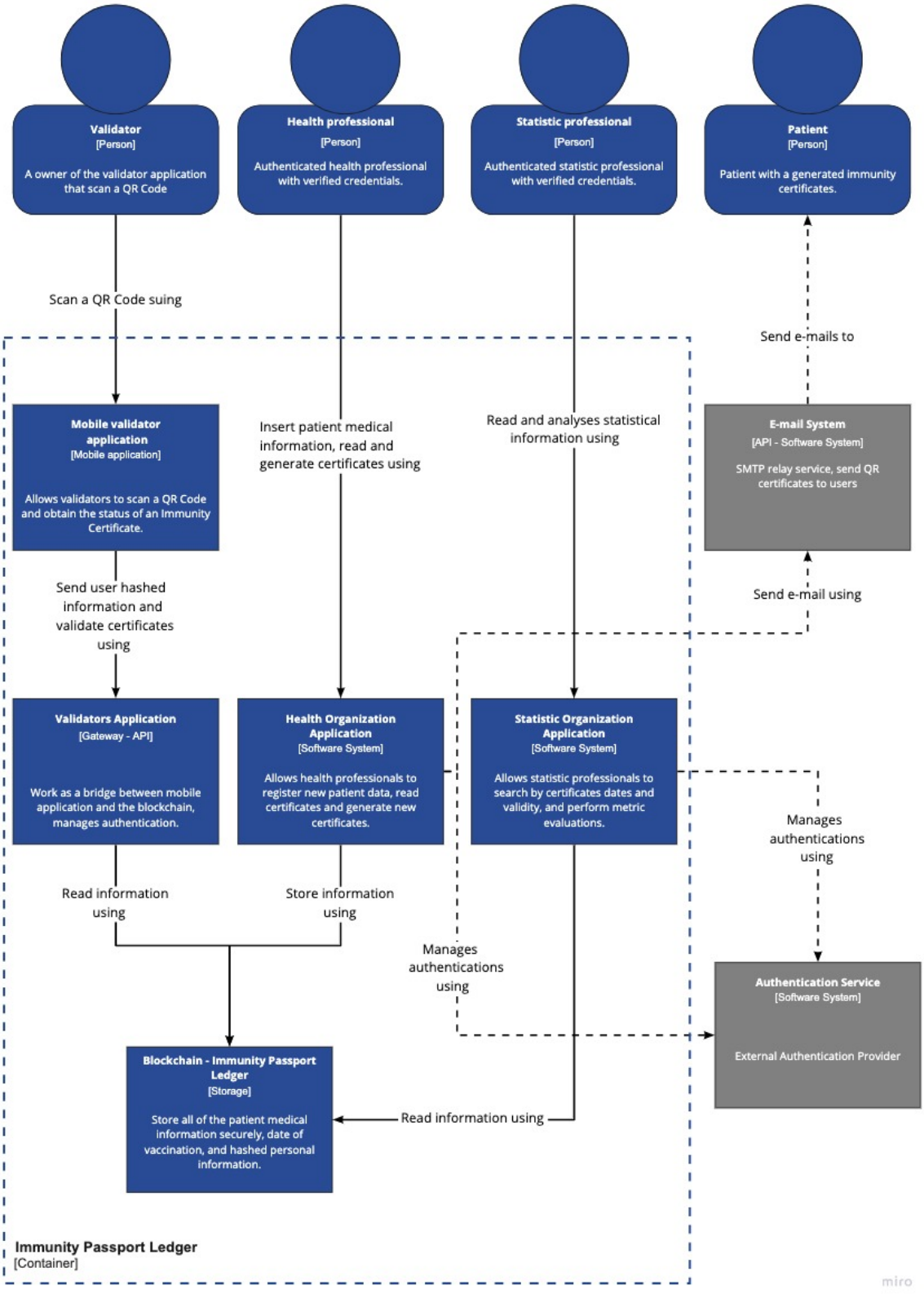

Figure 13: Proposed general system container diagram

As illustrated in Figure 13, the system is composed of four actors: Validator; Health professional; Patient; and Statistic professional.

Three actors interact with the ledger through two software systems and a gateway. Health professionals interact with a web application to generate certificates. Validators interact with the gateway through a mobile application that scans QR Codes. Statistic professionals use a web application to analyze metrics. Patients do not interact directly but receive emails from an external email relay service.

In order to make it easier to explain the contexts and their interactions, this diagram is divided into three diagrams for the three main actors, except for the patients that are part of the health professionals diagram.

The proposed system container diagram for health organizations is presented in Figure 14.

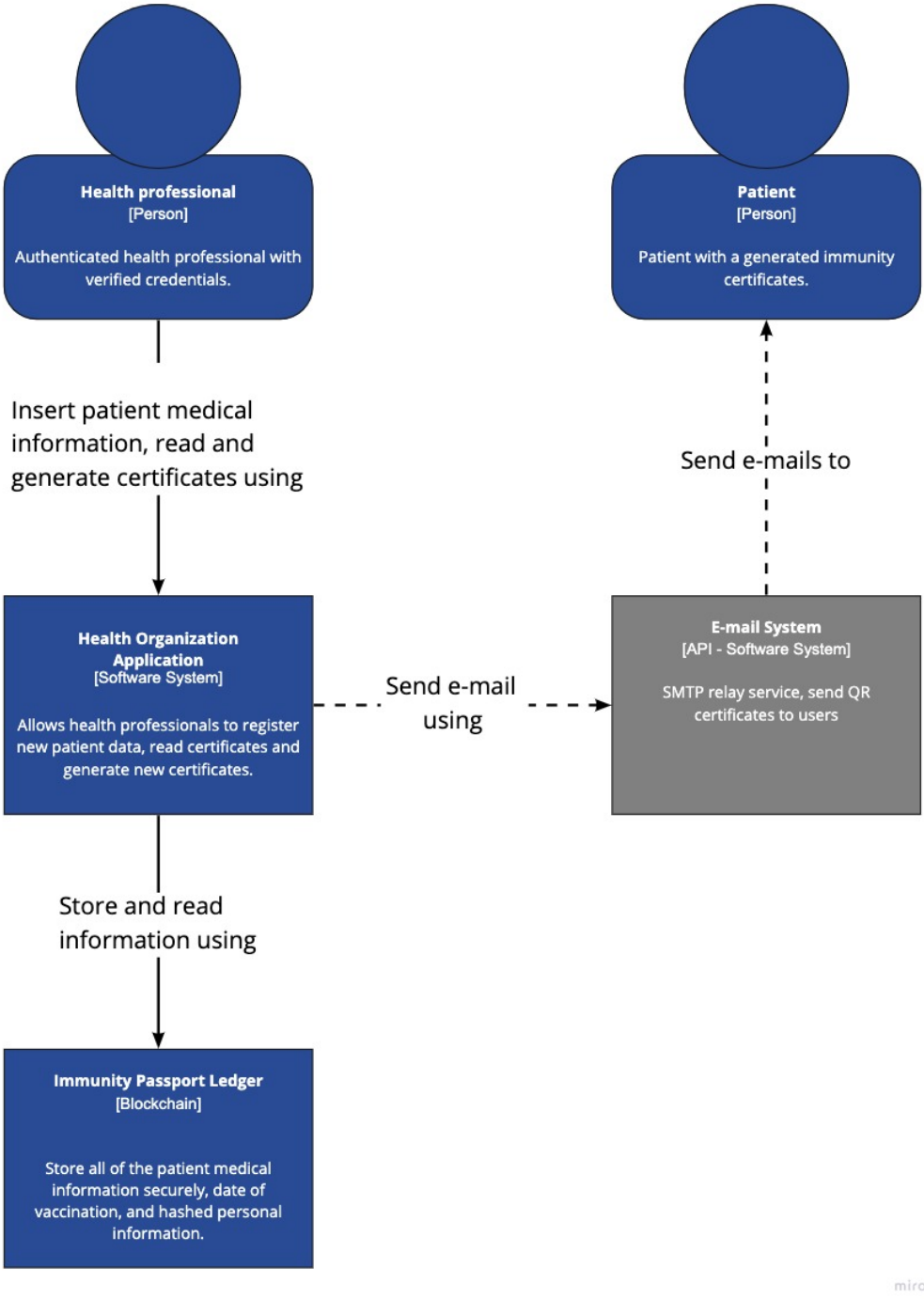

Figure 14: Proposed system container diagram for health organizations

The diagram illustrated in Figure 14 is based on the context of a healthcare organization where healthcare professionals and patients are included. Healthcare professionals need to be pre-authenticated to an existing healthcare system that will access the "Health Organization Application" software. Using this application,

Health professionals at the time of the vaccination insert health information, which will be used to generate the vaccination certificate.

This application will save the information entered by the healthcare professional in the "Immunity Passport Ledger", where it is securely saved in the blockchain.

"Health Organization Application" will then communicate with the external email service, which is grayed out as an external system, to send the QR Code to the patient by email. The healthcare professional must also print this QR code on the spot and deliver the certificate in digital and physical format, promoting accessibility for all.

The Validators organization system container diagram is proposed in Figure 15.

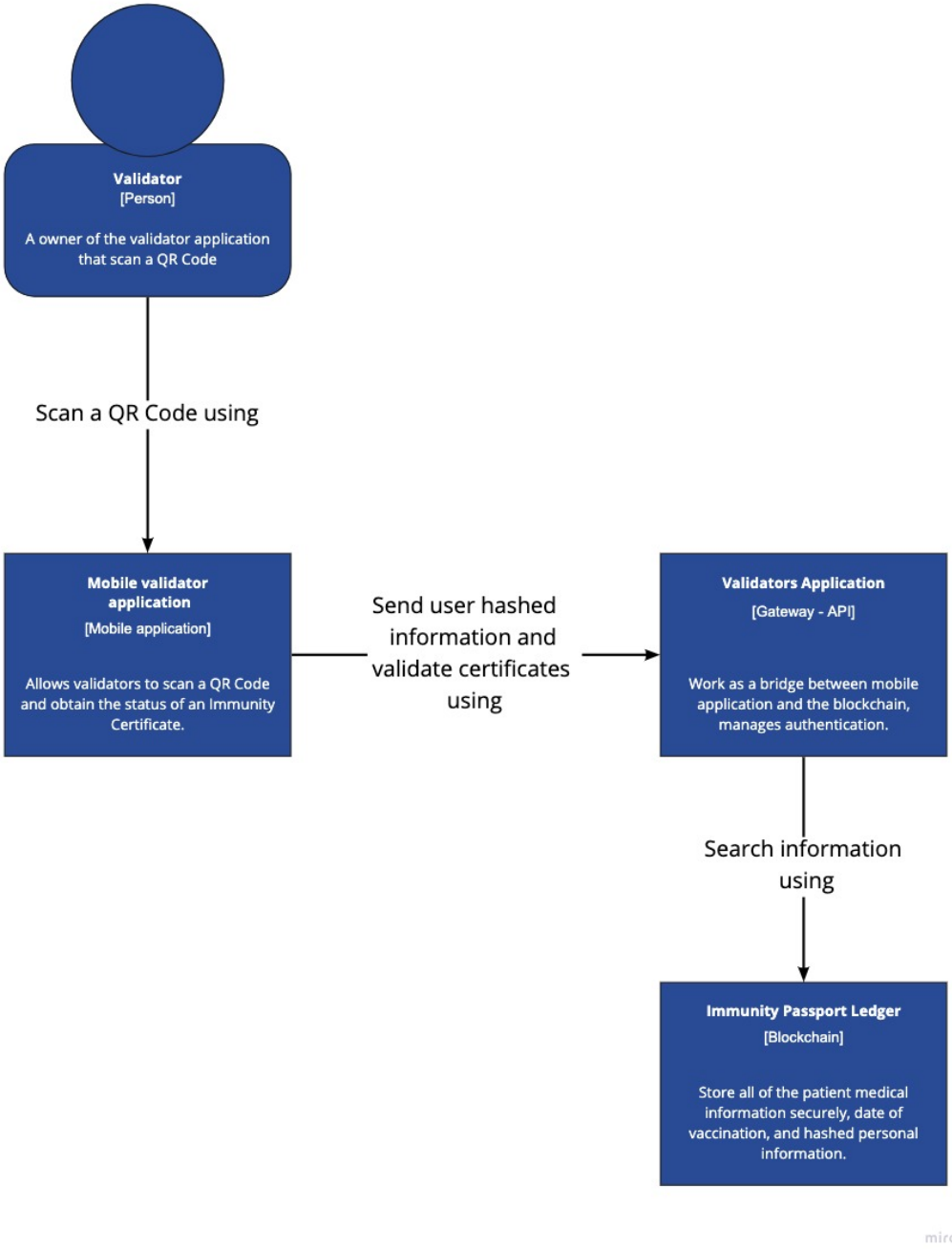

Figure 15: Proposed system container diagram for validators organizations

The diagram illustrated in Figure 15 is based on the context of a validators organization where anyone can engage using a mobile application.

Validators use the mobile application to scan a QR Code. In return, the mobile application will create a hash from the data collected from the QR code and send this hash to the "Validators application", which works as a Gateway. Validators application is necessary to manage blockchain access. API acts as a bridge between

the mobile application and the blockchain, limiting the access that the mobile application can have to the blockchain data and the smart contracts it can invoke. The Gateway, in turn, looks for the certificate in the blockchain, and if it exists, it receives the certificate's expiration date.

This expiration date is validated in the "Validators Application". A valid or invalid certificate message is returned to the mobile application.

This approach allows us to keep the certificate owner's data safe and not send any sensitive data through the web.

Statistic organization system container diagram is proposed in Figure 16.

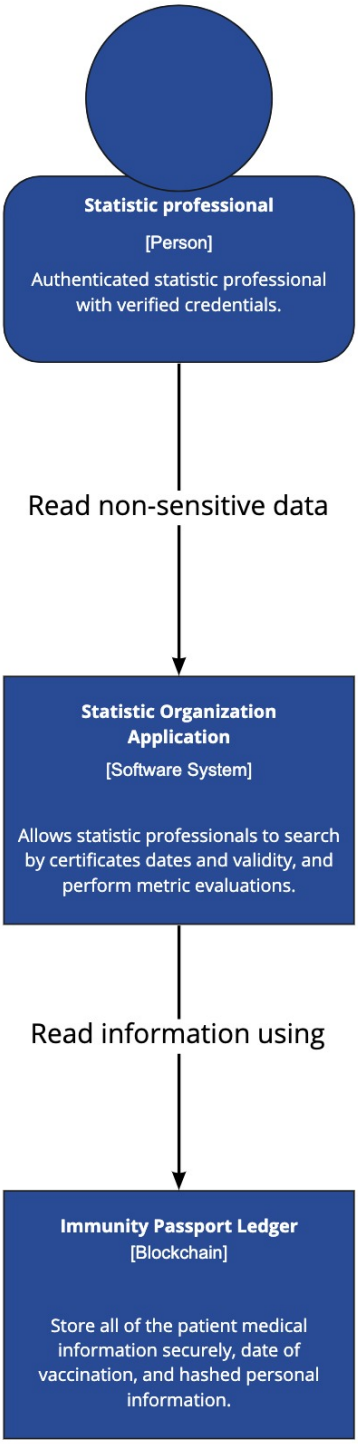

Figure 16: Proposed system container diagram for statistic organizations

The diagram shown in Figure 16 represents the context of a statistic organization.

Statistic professionals need to be pre-authenticated to an existing system that will access the "Statistic Organization Application" software.

This application allows statisticians and mathematicians to see and analyze the number of vaccinated and issued certificates, compare validation dates, and know when these certificates will expire.

Statistical organization applications query the ledger to retrieve information. Since all the data available on the blockchain is non-sensitive and hashed information, there are no privacy issues.

4.4 BLOCKCHAIN ARCHITECTURE DESCRIPTION

The blockchain architecture illustrated in Figure 17 exists within the Hyperledger network, where three organizations represent a consortium. Each of these organizations can use one or more peers to participate in the blockchain. In the illustrated example, each organization uses one peer.

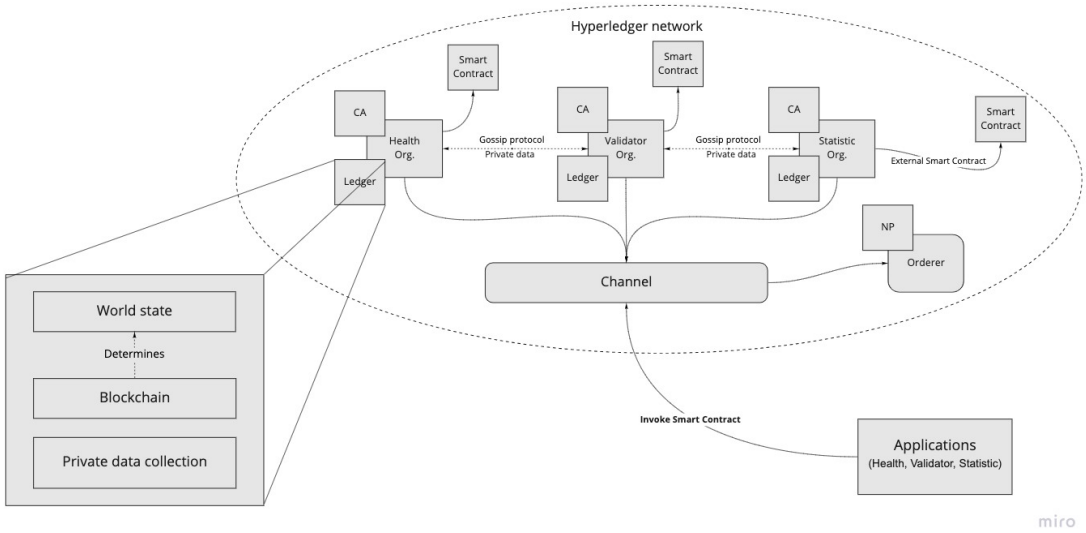

Figure 17: Proposed blockchain architecture

In the proposed architecture, three peers are represented with the name: Health Org.; Org Validator; Statistic Org.; Each of these peers has a version of the ledger distributed among the different peers. Each peer has a Certificate Authority (CA) that provides certificates for these organizations to authenticate their users and access the network with a certain authorization levels. Each peer also has an instance

of the smart contracts that have been approved to run on the blockchain. At runtime, all peers run, and the output between them all must be equal.

Orderer node is one of the blockchain components and is responsible for organizing the order of transactions in blocks and distributing blocks to all peers; it is also responsible for enforcing Network Policies (NP).

A channel in a Hyperledger network is defined as a subnet of communication between one or more peers. Each transaction on the network is performed on a channel, where each organization must be authenticated and authorized to transact on that channel.

The applications are not part of the Hyperledger Network; they use an SDK developed by the Hyperledger Fabric team to interact with the network and invoke Smart Contracts.

4.4.1 *Transaction Flow*

Components that constitute the proposed blockchain architecture are essential for the processing of transactions in the ledger. Figure 18 is an illustration of the transaction flow in the proposed architecture.

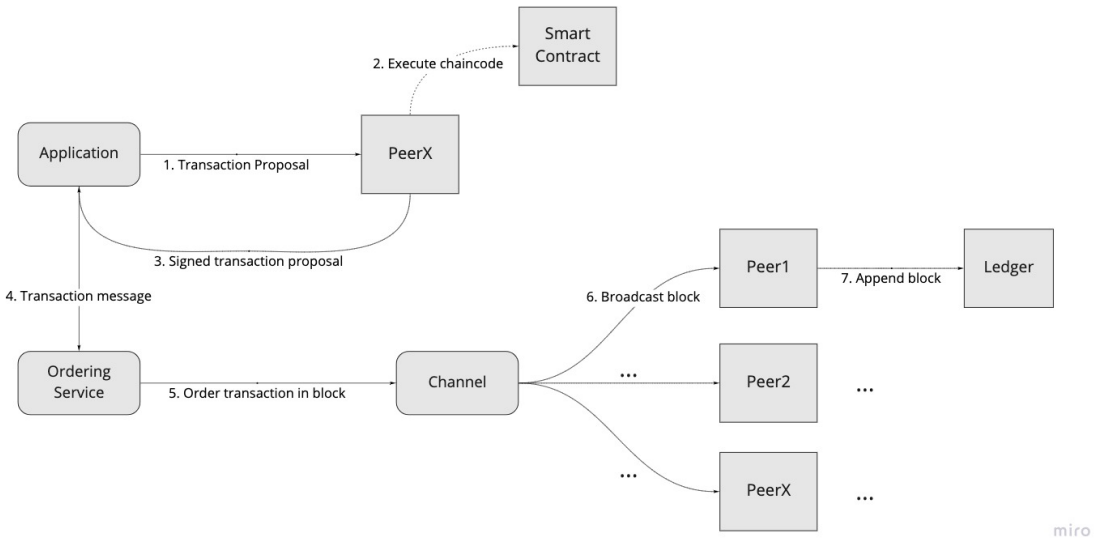

Figure 18: Network transaction flow

As presented in Figure 18, transaction flow shows the process of storing data on the blockchain, from the application level to the block creation and broadcasting among participants.

The transaction is initiated when:

1. Application invokes a Smart Contract to store or read data on the blockchain. A transaction proposal is generated through the SDK API. This proposal is a request to invoke a chaincode function that is part of a Smart Contract with specific input parameters.
2. All the endorsing peers verify if the transaction proposal has the correct format and has not been submitted before. Then the signature is validated to ensure the application has permission to propose a new transaction. The peer uses the submitted parameters to invoke the Smart Contract. Smart contract is executed, but no updates are made to the ledger, and an output is produced. This output includes a response value, read set, and write set.
3. The endorsing peers sign the Smart Contract execution output, which is sent back as a proposal response to the application.
4. The application verifies the signatures on the proposal response and compare the response to determine if they match. If the application is performing a read query to the ledger, the flow ends. Otherwise, it is performing a store operation on the ledger and proceeds to the next step.
5. The application sends the transaction message that contains a channel id read/write sets and the endorsing peers' signatures to the ordering service at the orderer component.
6. Ordering service verifies the signatures, ensures that the policies are met, orders the transaction chronologically, and creates a block of transactions.
7. The block is delivered to all peers on the channel, the transactions within the block are validated to ensure that endorsement policies are met, and that the ledger state did not change since the read set was generated on the execution.
8. Each peer appends the block to the chain, and for each legitimate transaction, the write sets are committed to the world state.

4.5 APPLICATION COMPONENTS ARCHITECTURE

The component diagram focuses on the elements that make up the applications of different organizations and how these components interact with other external components and the blockchain.

These diagrams help understand the building blocks of applications and are part of the application documentation process.

The proposed components architecture for health organization application is shown in Figure 19.

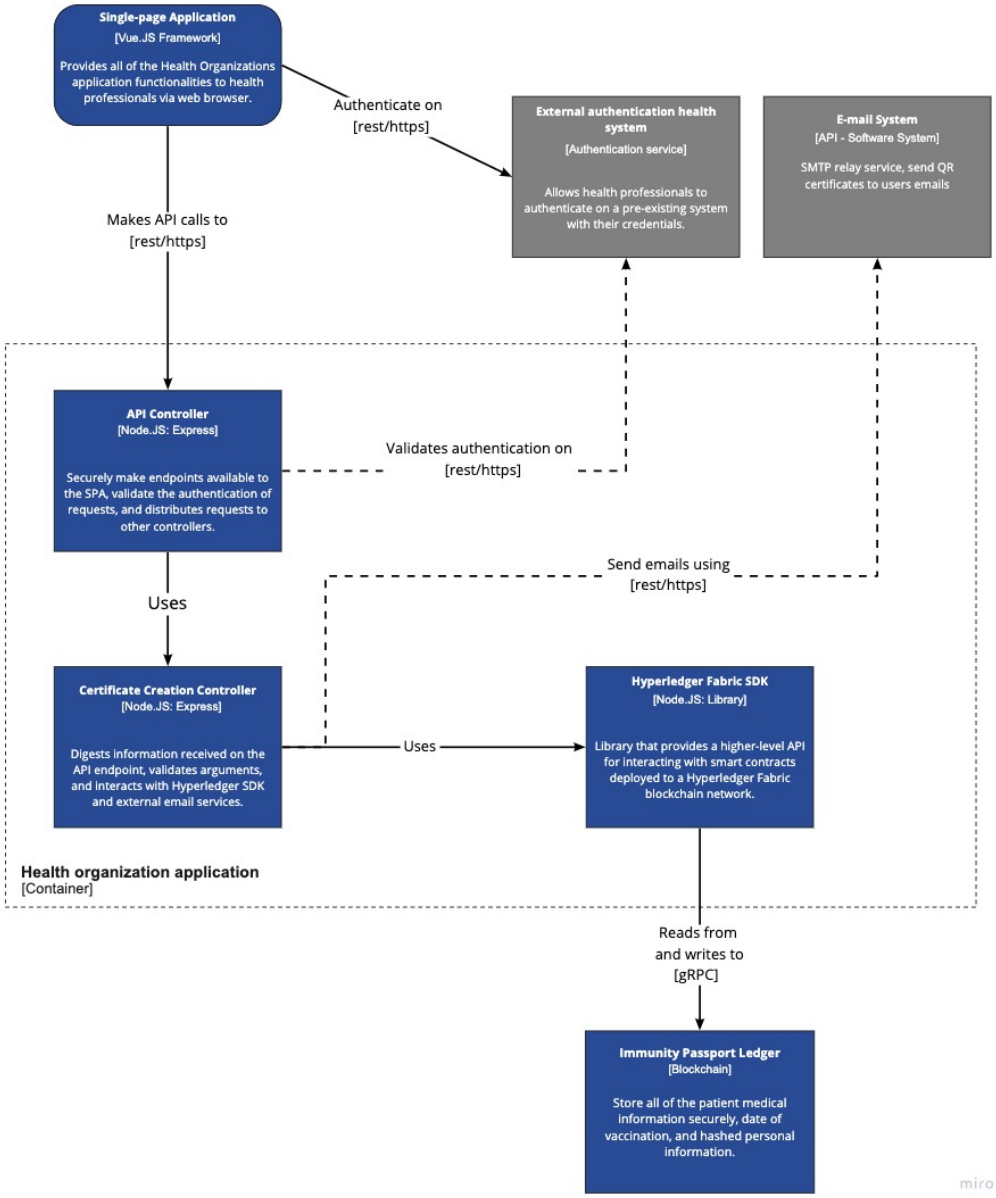

Figure 19: Health application architecture

As shown in Figure 19, the healthcare organization's application architecture is based on an API responsible for connecting single-page application to the blockchain. The single-page application uses a framework: Vue.js. This framework allows the

creation of SPAs and facilitates the creation, use, and reuse of views and data management in the browser.

This application connects to an external authentication system provided by the health institution, which is only accessible in a restricted way and through secure networks. The single-page application sends and receives information in a REST API format via HTTPS requests.

Inside the application container, there is the controller responsible for managing requests to the API. This controller receives requests and distributes them to other controllers. It is responsible for the security of the endpoints, and validates each request's authentication using a request to the external authentication service of the health system.

If the requests are properly authorized, they are forwarded to the certificate creation controller. The information received at the endpoint is analyzed and validated in the certificate creation controller, and a transaction request is generated using the SDK.

Hyperledger Fabric SDK provides a higher-level API for interacting with smart contracts deployed on the blockchain network.

After receiving the arguments and the name of the smart contract from the certificate creation controller, the SDK starts the transaction flow explained in section 4.4.1 using a gRPC connection. After the transaction is completed, an event is emitted, received by the SDK, and sent back to the controller. The controller parses the blockchain response and generates a success or error message displayed in the view.

The QR Code corresponding to the certificate is included and made available for printing along with the success message. The controller also sends an HTTPS request to the email service to email the patient with the certificate.

The proposed components architecture for validators organization application is illustrated in Figure 20.

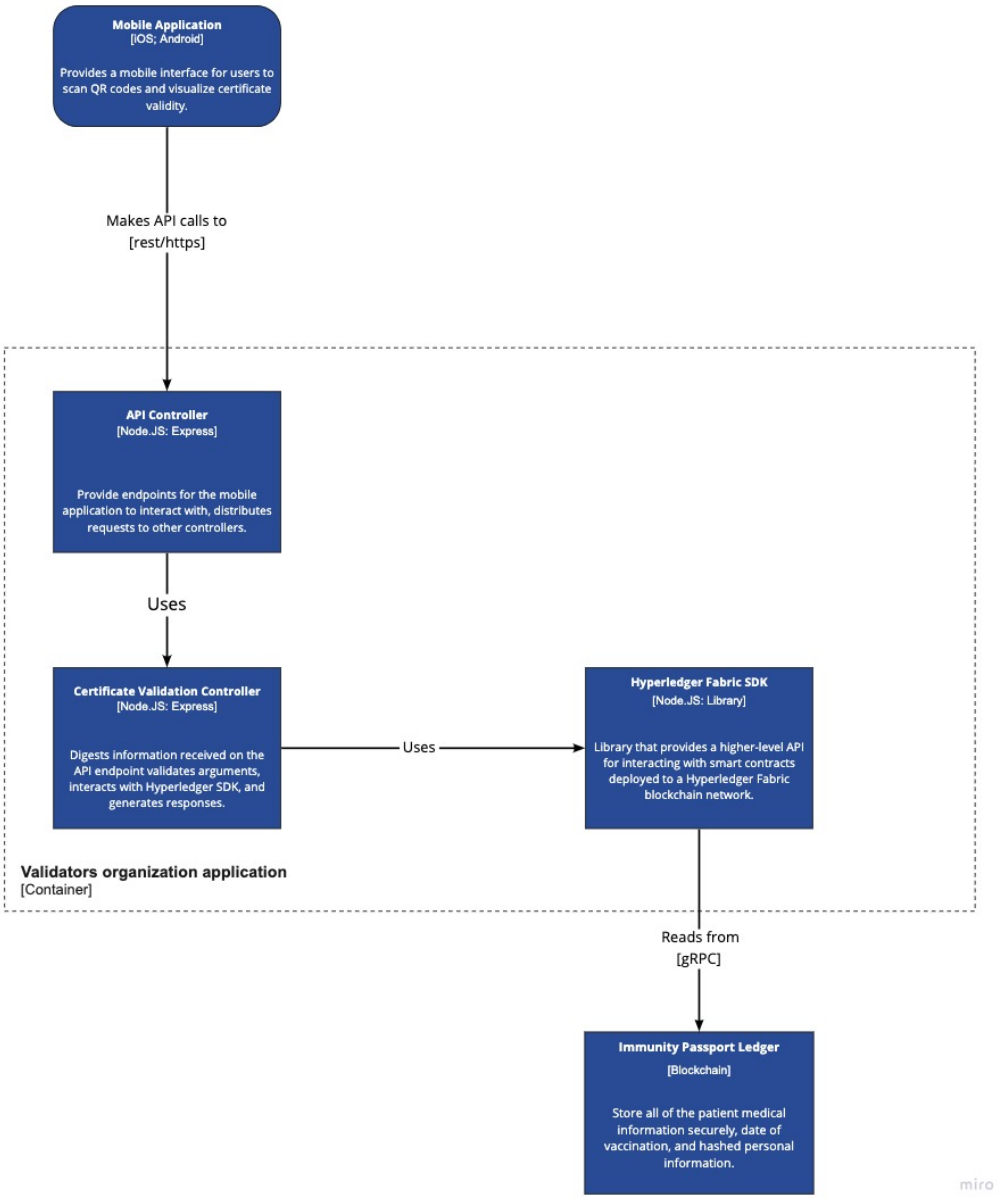

Figure 20: Validators application architecture

Figure 20 illustrates that the validators organization application relies on a mobile application to interact with the users.

The Android and iOS mobile applications allow users to scan the QR Code of certificates and check the certificate's validity. The mobile application makes an HTTPS request to an Endpoint API of the validators application.

The API controller receives these requests and interacts with the certificate validation controller that validates the information sent by the mobile application.

Through the Contract interface of the Hyperledger Fabric SDK library, it is created a evaluate transaction that queries the state from the ledger targeting the Smart Contract to validate the certificates on the blockchain. This interaction occurs under the gRPC protocol secured by TLS.

If a valid certificate is found on the ledger, the validity date is returned to the validators organization application, and it is compared with the server time. If valid, a success message is returned to the mobile application. If this date is invalid or the certificate does not exist on the ledger, an error message is sent to the mobile application.

The proposed components architecture for statistic organization application is presented in Figure 21.

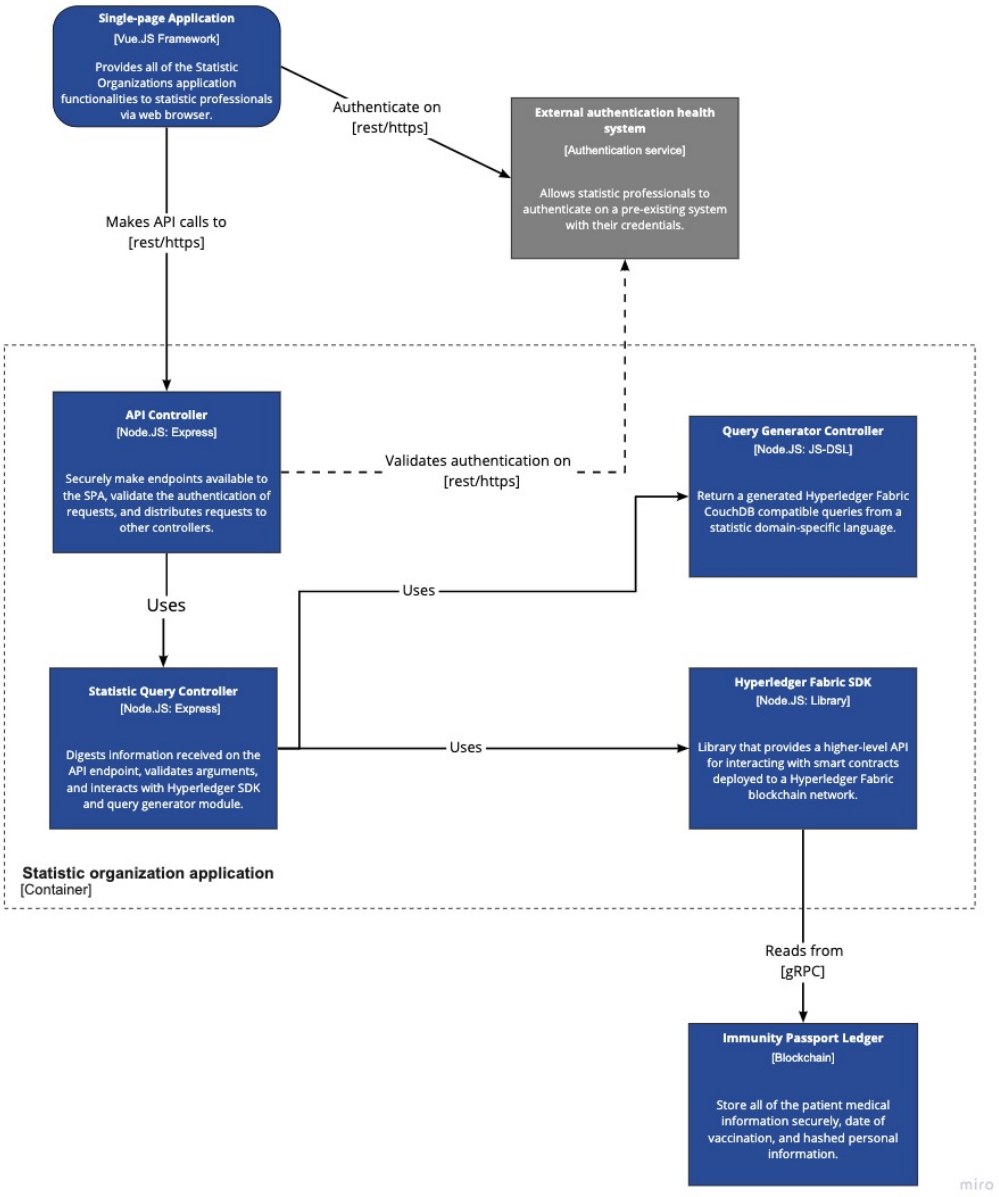

Figure 21: Statistic application architecture

Figure 21 presents the architecture proposed for the application of the statistical organization. Similarly, the application used by healthcare professionals is also a SPA that uses an external authentication system for statisticians.

After authentication, the interface allows a range of statistical queries, custom blockchain queries, and data visualization.

These requests are sent to the application API controller, which securely validates the authentication of requests through a request to the authentication system.

If the request is validated, it proceeds to the statistic query controller, processing and validating the requested information. If the request is a custom blockchain query, the request proceeds to the Query Generator Controller.

In the query generator controller, the domain-specific language used by the stats is converted to a CouchDB-compatible query. In turn, the statistic query controller uses Hyperledger Fabric SDK to query the state from the ledger.

The API returns the data to the single-page application that presents them graphically.

5

PROTOTYPE IMPLEMENTATION

This chapter presents the system's most important functionalities, emphasizing the created relationships between the applications, the blockchain, and the technological challenges exceeded during their development.

This chapter is divided into three parts. In the first part, a description of blockchain framework implementation and functionalities are addressed. Then, in the second part, are presented the health application functionalities and interactions. In the last part, the validator gateway API and mobile application are shown.

5.1 BLOCKCHAIN

As justified in Chapter 4, the blockchain technology chosen to be the source of truth was Hyperledger Fabric. This technology allows the development of permissioned blockchain networks.

Three organizations were created to join and interact with the network, health organization, validators organization, and statistic organizations.

5.1.1 *Network implementation*

The first step in creating the blockchain network is to configure the Certificate Authorities to create the identities and Membership Service Providers for the orderer health organization. For simplicity's sake, a built-in Fabric-CA service is used to manage this overhead in this prototype.

The Network is formed when an orderer node is started, for this prototype is used a single orderer.

This orderer is configured with a set of Network Policies and is the initial administration point for the Network. The health organization is the owner of the orderer peer.

The second step consists in creating a channel through which the peers can communicate privately with each other. This channel has a set of policies that will be discussed in the next section.

Next, the peer for health organization is created and joined to the channel network. At this point, the ledger and the genesis block is generated. The first block stores the channel configuration.

At this point, Figure 22 is a faithful representation of the network.

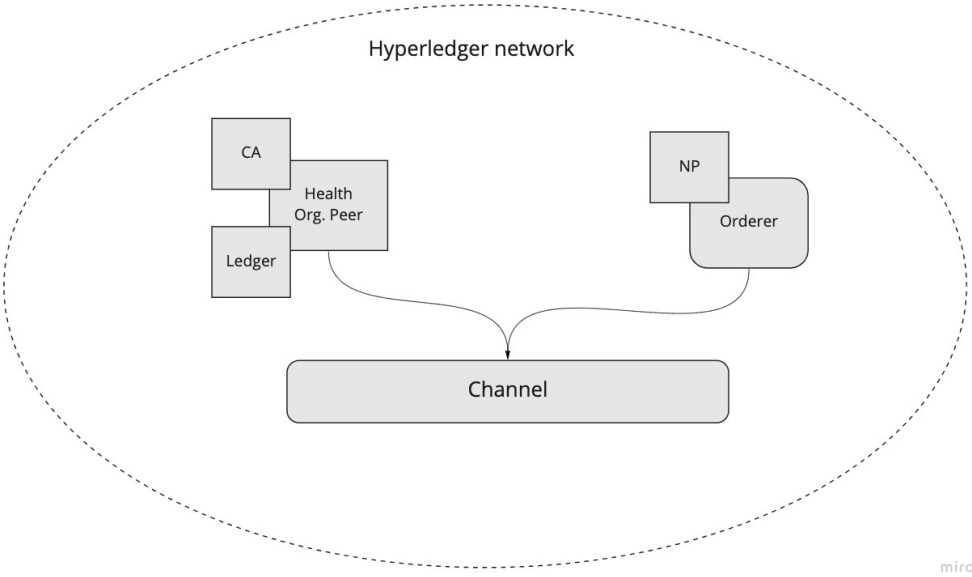

Figure 22: Network, Health Organization peer and Orderer

CA are created for Validators organization peer and for Statistics organization peer, and they can now join the network.

Organizations cannot yet interact with the ledger because it did not have a Smart Contract deployed yet.

The next step is deploying a Smart Contract to the channel so that the end-users can invoke this smart contract and interact with the ledger.

In order to deploy the smart contract, it needs to follow a process known as the Fabric chaincode lifecycle.

The first step consists in installing the chaincode in at least two of the peers. Since this network requires a majority of the channel members to approve a chaincode before it can be used, two endorser peers are needed. After the installation, a package ID is returned and will be used to approve the chaincode. If both of the peers approve the chaincode definition, it can now be committed to the channel.

After this endorsement, it is now possible to invoke the chaincode presented in Appendix B.

5.1.2 *Policies implementation*

Policies are one of the most critical aspects of the Hyperledger Fabric implementation.

These policies define who can invoke smart contracts, read and write on the ledger, endorse transactions and smart contracts, and how many endorses are needed to approve a transaction or a new organization.

The first policies to be put in place are the network configuration policies that describe a set of administrative capabilities, such as the consensus, how new blocks are created, and the list of participant organizations.

Second, we have the channel policies that govern how users interact with the channel, such as organizations that need to approve a chaincode to deploy it to a channel. As described in the previous section, these policies are satisfied if a majority of the organizations agree.

Furthermore, the Access Control Lists (ACL) manages resources by associating a policy with a resource. The invoking chaincode policy was overridden for this prototype to allow only the health organization to write on the ledger.

5.1.3 *Smart Contract Implementation*

Smart Contracts are the core of the Hyperledger Fabric blockchain. It defines the executable logic that generates new data do the ledger and reads the current data of the ledger.

Before organizations can transact or read the ledger, they must agree on a standard set of contracts comprising standard terms, data rules, concept definitions, and processes. This set of contracts defines the business model that governs all of the interactions between the network participants. A Smart Contract is invoked by an application to generate transactions that are recorded on the ledger.

At this point, is defined the structure of the data that will be stored on the blockchain.

Table 3 illustrates the data stored on the ledger.

Table 3: Data stored on the blockchain

Key	Example Value	Description
HealthNumberHash	18e98512ab...	This field stores a hashed health number generated with a salt that only exists on the QR Code data and only belongs to the user.
ImmunityDate	1635787071	This field has the date, hour, minute and second, the certificate was generated in a UNIX timestamp format.
ImmunityValidityInDays	330	This field is introduced by the health professional at the vaccination occurrence and represents the validity the certificate has.
Issuer	HealthOrg1	This field shows who was the issuer of the certificate.

Table 3 shows the four parameters that make up a blockchain certificate. These parameters are initially defined in the Chaincode, which can be seen in Appendix B.

The Chaincode is written in the Go programming language (*Go - Build fast, reliable, and efficient software at scale* n.d.). It has defined a typed collection of fields known as a struct that stores the parameters during the execution of the Smart Contract.

The Chaincode installed on the peers has two Smart Contracts: one to create a new certificate in the "CreateCertificate" blockchain and another to return the certificate "ReadCertificate", if it exists in the blockchain.

5.2 HEALTH APPLICATION

The Health application is used by health professionals to generate new certificates at the time of vaccination.

The installation and management of the application should be the responsibility of each healthcare organization. The structure and application presented in this work serve as a guideline for adaptation by each health organization to a real environment.

5.2.1 *Authentication*

Authentication is a required step to use the application and should be performed by a health professional recurring to an existing authentication service.

As illustrated in Figure 23, Health applications provide a login interface for the authentication of health professionals.

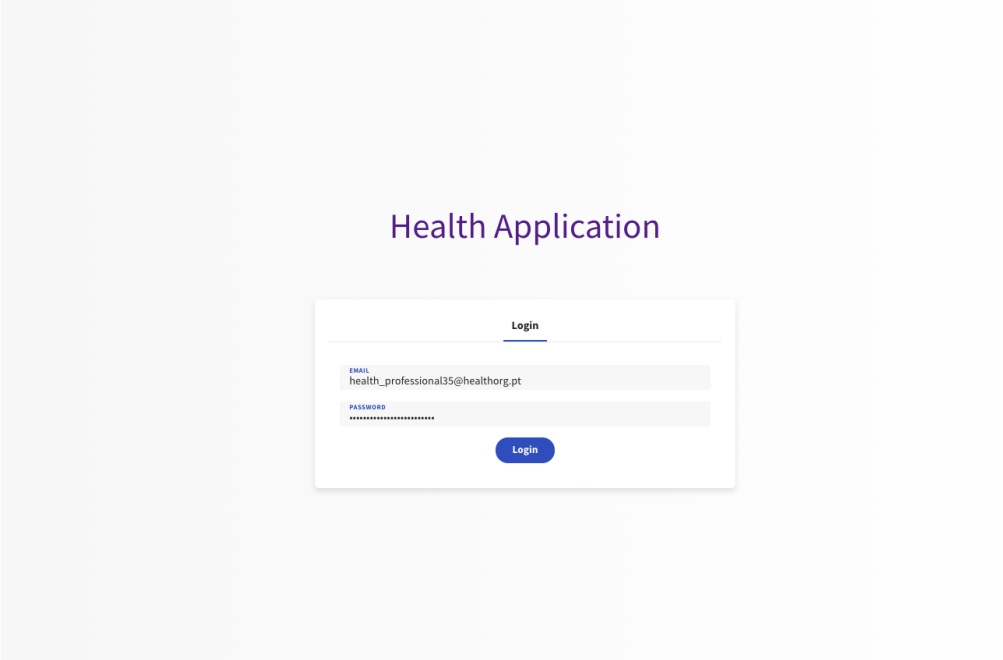

Figure 23: Health application login

5.2.2 *Certificate Generation*

Certificate generation is the core of this project, and healthcare professionals use this feature after vaccination to generate a new certificate for a patient.

This feature is intended not to be intrusive to the patient data, not to record more data than what it is strictly necessary to validate that a particular person is the valid holder of the certificate.

As presented in Code 1 when generating the certificate, a hash is created that corresponds to the user's health number. A random salt with 16 bytes is generated using the Crypto module. The salt is used used to create the hash and is stored in QR Code. This mechanism allows the mobile application to replicate the hash without performing a blockchain request.

Code 1: Certificate creation controller

```
'use strict';
const sdk = require('./app');
const saltedSha256 = require('salted-sha256');
const crypto = require('crypto');


async function createCertificateController(data) {
        let result = {};
        let err;

        let immunityDate = Date.now();
        let validity = data.validity;

        let salt = crypto.randomBytes(16).toString('base64');
        let hashHealthNumber = saltedSha256(data.healthNumber, salt);

        result, err = await sdk.generateCertificate({hashHealthNumber,
        ↪  immunityDate, validity});

        if(err) {
                return {
                        error: err
                };
        }

        let responseToEncode = {
                healthNumber: data.healthNumber,
                salt: salt,
                issuer: result.issuer
        };

        const base64Response =
        ↪  Buffer.from(JSON.stringify(responseToEncode)).toString('base64');
        let response = {
                result: base64Response,
                email: data.email,
                error: null
        };

        return response;
}

module.exports = { createCertificateController };
```

As shown on Code 2 this mechanism allows that looking at the data on the blockchain, it is impossible to know to whom they belong. However, through the QR Code that the patient is the only holder and contains the health number and the salt used to generate the hash, it is possible to perform a rehash and search for a certificate on the blockchain.

Code 2: Certificate creation using SDK

```
'use strict';

const { Gateway, Wallets } = require('fabric-network');
const FabricCAServices = require('fabric-ca-client');
const path = require('path');
const { buildCAClient, registerAndEnrollUser, enrollAdmin } =
↪  require('./CAUtil.js');
const { buildCCPOrg1, buildWallet } = require('./AppUtil.js');

const channelName = 'channel1';
const chaincodeName = 'hpCC';
const healthOrg1 = 'healthOrg1';
const walletPath = path.join(__dirname, 'wallet');
const healthOrgUser1 = 'healthUser1';

async function setupConnection(){
        try {
                const ccp = buildCCPOrg1();
                const caClient = buildCAClient(FabricCAServices, ccp,
                ↪  'ca.org1.example.com');
                const wallet = await buildWallet(Wallets, walletPath);
                await enrollAdmin(caClient, wallet, healthOrg1);
                await registerAndEnrollUser(caClient, wallet, healthOrg1,
                ↪  healthOrgUser1, 'org1.department1');
                return [ccp, wallet];
        } catch (error) {
                return error;
        }
}

async function generateCertificate(data) {
        let [ccp, wallet] = await setupConnection();
        const gateway = new Gateway();
        let result;
        try {
                await gateway.connect(ccp, {
                        wallet,
                        identity: healthOrgUser1,
                        discovery: { enabled: true, asLocalhost: true }
                });
                const network = await gateway.getNetwork(channelName);
                const contract = network.getContract(chaincodeName);
                result = await contract.submitTransaction('CreateCertificate',
                ↪  data.hashHealthNumber, data.immunityDate, data.validity);
        } catch (err) {
                let objErr = err;
                objErr.msg = err.toString();
                return '', objErr;
        } finally {
                gateway.disconnect();
        }
```

```
48          return JSON.parse(result.toString()), null;
49  }
50
51
52  module.exports = { generateCertificate };
```

Through this process, we have a non-intrusive system that respects privacy and is secure.

This system gives total ownership of the data to the holder of the QR Code.

As illustrated in Figure 24, the health professional introduces the health number and email of the patient and the validity of the certificate that should respect standards stipulated by health regulators for attesting immunity.

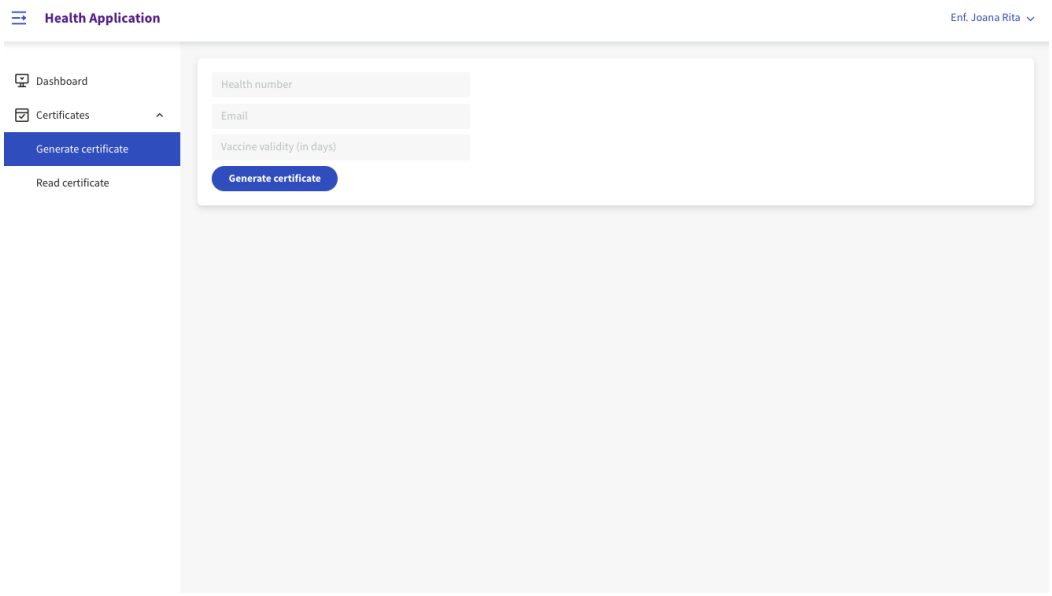

Figure 24: Health application generate certification page

After entering the patient's data and proceeding with the generation of the certificate, a page with the QR Code is displayed, as shown in Figure 25.

Figure 25: Health application generated certificated

This page informs that the QR Code was also sent to the patient's email, and on the right side, it is possible to print the page.

It is important to remember that both the patient's health number, QR Code, and email are not saved in the system, and the only persisted data are shown in Table 3.

Table 4 shows and describes the data in the QR Code, generated by the health application.

Table 4: Data stored on QR Code

Key	Example Value	Description
HealthNumber	332231213	This field corresponds to the patient's health number present in the QR Code.
Salt	4f+7j@FxI#6s	This field is a randomly created string used as an additional input to the one-way function that hashes the health number.
Issuer	HealthOrg1	This field shows who was the issuer of the certificate.

The data has a JSON format and is encoded in Base64 format used to generate the QR Code which is illustrated in Figure 26.

ewogICAiSGVhbHRoT
nVtYmVyIjoiMzMyMjM
xMjEzIiwKICAgIlNhbH
QiOiI0Zis3akBGeEkjN
nMiLAogICAiSXNzdW
VyIjoiSGVhbHRoT3Jn
MSIKfQ==

Figure 26: QR Code certificate and Base64 content

The QR Code shown in Figure 26 is the artifact kept by the patient and it is used to validate their immunity using the mobile application of the validator.

5.3 VALIDATORS APPLICATION

The validators application is a mobile application that allows scanning the QR codes of certificates created by the health application. It aims to check whether these certificates are valid or invalid.

This application has a simple interface and is easy to use. The home page contains a button to open the QR Code scan functionality, as illustrated in Figure 27.

Figure 27: Certificate validator mobile application

The mobile application has the home page in Figure 27 instead of having camera activity because it uses up a lot more battery when the camera is turned on and used continuously.

Whenever a new certificate is validated, the camera's activity is closed, saving battery life.

5.3.1 *QR Code Scanner*

Figure 28 shows that the QR Scan functionality starts a new camera activity and uses the ZXing Barcode Scan library (Owen, n.d.) to read the QR Code content.

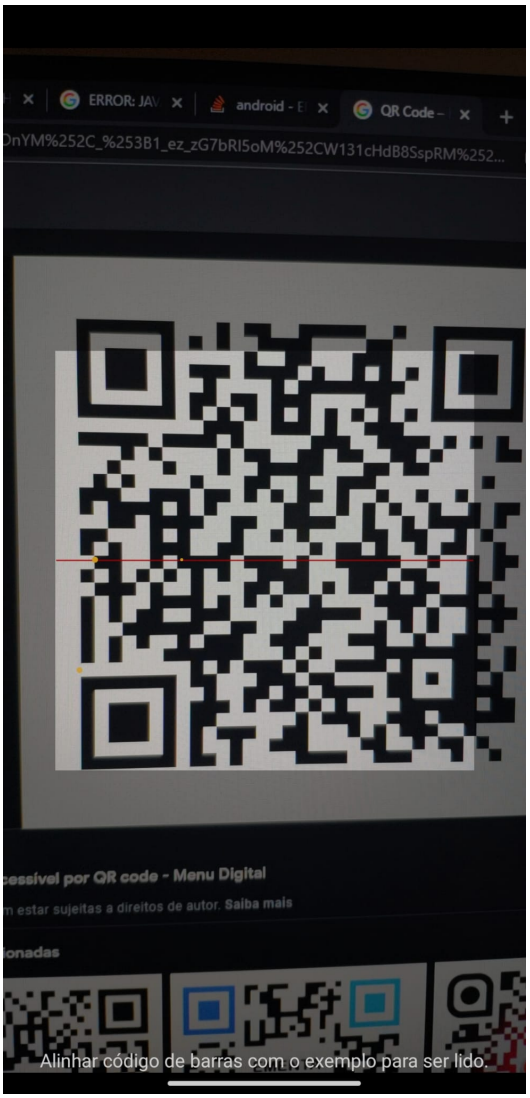

Figure 28: QR Code scan functionality

The QR Code content is decoded to a JSON format, using the health number, and the salt the hash stored in the blockchain is replicated. This hash is sent to the application and it is used to search for a match in the blockchain.

Using this mechanism, no user information is sent over the internet during communication with the validators application.

5.3.2 *Certificate Validation*

In response to sending the hash, the application returns a valid or invalid certificate response.

If the certificate has been validated, the user number, that only exists locally is displayed, together with a valid certificate message, as shown in Figure 29.

This health number allows the validator to compare it with an identification belonging to the person who presented the certificate and proves that he is the certificate's owner.

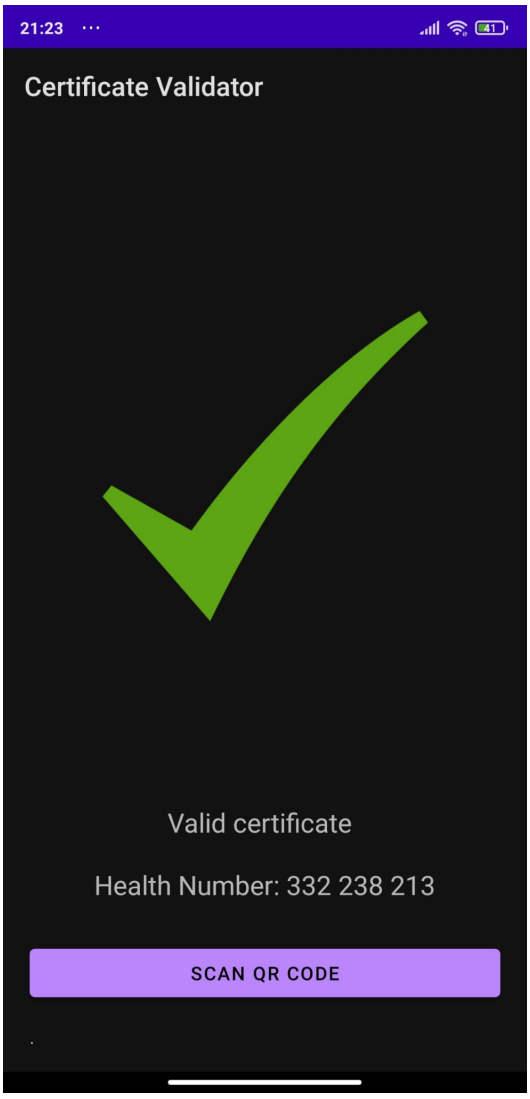

Figure 29: Certificate validator after scan valid certificate

If the certificate is considered invalid for any other reason, such as being out of date or not existing in the blockchain, an invalid certificate message is displayed in the mobile application as illustrated in Figure 30.

Figure 30: Certificate validator after scan invalid certificate

6

CONCLUSIONS AND FUTURE WORK

Returning to work in a post-covid society remains a challenging mission. Mass vaccination remains the best weapon to reduce COVID-19 mortality, and with this, the need to certify that an individual is vaccinated.

The certifications mechanism and system are flawed and share more data than needed for validation. These systems are not tamper-proof and can be exploited, bad actors are not accountable, and certificates are sold on black markets.

This work proposes and describes a prototype of an Immunity Passport Ledger that allows health professionals to register and store Immunity Certificates while respecting user privacy and data ownership. In this project, the proposed Immunity Passport Ledger also allows the certificates to be securely validated without providing more information than what is strictly necessary for this event.

Regarding the proposed goals for this project, it is plausible to consider that they were achieved:

- O1 - A study of blockchain and distributed ledger technologies was conducted in Chapter 2, comparing the most important characteristics of blockchain technologies. This objective set a theoretical framework for a better understanding of the following work.
- O2 - In Chapter 3, previous and ongoing applications related to immunity certificates have been analyzed with a particular focus on flaws and improvement points.
- O3 - A proposed architecture for the Immunity Passport Ledger prototype was presented in Chapter 4. Through a requirements analysis, it was possible to understand the architectural needs of the prototype. An architecture that allows the registration and validation of certificates were proposed. A description focused on the technical specifications of the prototype was made.
- O4 - A proof of concept for the Immunity Passport Ledger was illustrated in Chapter 5. The system's most important functionalities are explained and demonstrated. Is provided a detailed specification of implementation details with source code from the developed proof of concept.

As a result of this work, two publications were accepted at the 11th World Congress on Information and Communication Technologies with the title Immunity Passport Ledger: Digital certificates implemented on a permissioned blockchain in Appendix C and Building trust with a contact tracing application: a blockchain approach in Appendix D. Furthermore, all the developed prototype code is available in a dedicated Github repository as one source.

Regarding the future work that can be developed on the Immunity Passport Ledger, several options can be explored, such as developing the statistical application that already has the architectural base prepared and the existing organization in the blockchain.

It is also essential to conduct a performance and scalability analysis of the blockchain to ensure that it can store the necessary amount of data and analyze the requirements for the number of instances of peer nodes and orderer nodes to ensure the network's performance.

BIBLIOGRAPHY

WHO (2020). *WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020*. Tech. rep. URL: https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020.

INESCTEC (2020). *STAYAWAY COVID*. URL: https://stayawaycovid.pt/.

Miranda Ramos, Luis Felipe (2020). "Digital contact tracing and data protection: assessing the French and Portuguese applications". In: *UNIO – EU Law Journal* 6.2, pp. 35–48. DOI: 10.21814/unio.6.2.2767.

Zhao, Juanjuan et al. (2020). "Antibody Responses to SARS-CoV-2 in Patients with Novel Coronavirus Disease 2019". In: *Clinical Infectious Diseases* 71.16, pp. 2027–2034. ISSN: 15376591. DOI: 10.1093/cid/ciaa344. URL: https://www.who.int/news-room/commentaries/detail/immunity-passports-in-the-context-of-covid-19.

Frankenfield, Jake (2018). "Distributed Ledger Technology Definition". In: *Investopedia* October. URL: https://www.investopedia.com/terms/d/distributed-ledger-technology-dlt.asp.

Rauchs, Michel et al. (2018). "Distributed Ledger Technology Systems: A Conceptual Framework". In: *SSRN Electronic Journal* August. ISSN: 1556-5068. DOI: 10.2139/ssrn.3230013.

McLean, Sue and Simon Deane-Johns (2016). "Demystifying Blockchain and Distributed Ledger Technology – Hype or Hero?" In: *Computer Law Review International* 17.4, pp. 1–8. ISSN: 2194-4164. DOI: 10.9785/cri-2016-0402.

Saad, A. and Soo Young Park (2019). "Decentralized directed acyclic graph based DLT network". In: *PervasiveHealth: Pervasive Computing Technologies for Healthcare* Part F1481, pp. 158–163. ISSN: 21531633. DOI: 10.1145/3312614.3312647.

Baird, Leemon, Mance Harmon, and Paul Madsen (2018). "Hedera: A Public Hashgraph Network & Governing Council". In:

Eric Harris-Braun Nicolas Luck, Arthur Brock (2018). "Holochain: scalable agent-centric distributed computing". In: p. 14. URL: https://github.com/holochain/holochain-proto/blob/whitepaper/holochain.pdf.

Cäsar, Florian et al. (2020). “Cerberus A Parallelized BFT Consensus Protocol for Radix”. In:

Satoshi Nakamoto (2008). “Bitcoin: A Peer-to-Peer Electronic Cash System”.

Velde, François R. (2013). “Bitcoin - A Primer”. In: *Chicago Fed Letter* December, pp. 1–4. ISSN: 08950164. URL: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=92563197&site=ehost-live.

Androulaki, Elli, Artem Barger, et al. (2018). “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains”. In: *Proceedings of the 13th EuroSys Conference, EuroSys 2018*. Vol. 2018-Janua. ACM. ISBN: 9781450355841. DOI: 10.1145/3190508.3190538. URL: https://doi.org/10.1145/3190508.3190538.

Valenta, Martin and Philipp Sandner (2017). “Comparison of Ethereum, Hyperledger Fabric and Corda”. In: *Frankfurt School Blockchain Center* June, p. 8. URL: www.fs-blockchain.decontact@fs-blockchain.dewww.twitter.com/fsblockchainwww.facebook.de/fsblockchain%20www.fs-blockchain.decontact@fs-blockchain.dewww.twitter.com/fsblockchainwww.facebook.de/fsblockchain%0Ahttps://medium.com/@philippsandner/comparison-of-et.

Reid, Fergal and Martin Harrigan (2013). “An Analysis of Anonymity in the Bitcoin System”. In: *Security and Privacy in Social Networks*. New York, NY: Springer New York, pp. 197–223. ISBN: 9781461441397. DOI: 10.1007/978-1-4614-4139-7{_}10. URL: http://link.springer.com/10.1007/978-1-4614-4139-7_10.

Lamport, Leslie, Robert Shostak, and Marshall Pease (1982). “The Byzantine Generals Problem”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4.3, pp. 382–401. ISSN: 15584593. DOI: 10.1145/357172.357176.

Liu, Debin and Jean Camp (2006). “Proof of Work can Work”. In:

Du, Mingxiao et al. (2017). “A review on consensus algorithm of blockchain”. In: *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017* 2017-Janua, pp. 2567–2572. DOI: 10.1109/SMC.2017.8123011.

Lee, Boohyung and Jong Hyouk Lee (2017). “Blockchain-based secure firmware update for embedded devices in an Internet of Things environment”. In: *Journal of Supercomputing* 73.3, pp. 1152–1167. ISSN: 15730484. DOI: 10.1007/s11227-016-1870-0.

Vaidya, Kiran (2016). *Bitcoin’s implementation of Blockchain*. URL: https://medium.com/all-things-ledger/bitcoins-implementation-of-blockchain-2be713f662c2.

Buterin, Vitalik (2014). "A next-generation smart contract and decentralized application platform". In: *Etherum* January, pp. 1–36. URL: http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf.

Antonopoulos, Andreas M. and Gavin Wood (2018). *Mastering Ethereum : building smart contracts and DApps*, p. 384. ISBN: 978-1491971949.

Idelberger, Florian et al. (2016). "Evaluation of logic-based smart contracts for blockchain systems". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9718.July, pp. 167–183. ISSN: 16113349. DOI: 10.1007/978-3-319-42019-6{_}11.

Prates, Silva and Bernardo Gast (2019). "Formal Analysis of Ethereum Virtual Machine Bytecode Patterns". In:

Ethereum development documentation | ethereum.org (n.d.). URL: https://ethereum.org/en/developers/docs/.

OpenSea, the largest NFT marketplace (n.d.). URL: https://opensea.io/.

Peepeth (n.d.). URL: https://peepeth.com/welcome.

Aave – Open Source DeFi Protocol (n.d.). URL: https://aave.com/.

Ethereum (2021). *The Eth2 upgrades | ethereum.org*. URL: https://ethereum.org/en/eth2/.

Hyperledger – Open Source Blockchain Technologies (n.d.). URL: https://www.hyperledger.org/.

Hyperledger - Hyperledger - Hyperledger Confluence (n.d.). URL: https://wiki.hyperledger.org/.

Hyperledger Foundation (2020). *Hyperledger Sawtooth*. URL: https://www.hyperledger.org/use/sawtooth%20https://search.ebscohost.com/login.aspx?direct=true&db=edswdc&AN=edswdc.D02001833&lang=zh-cn&site=eds-live.

Architecture Guide — Sawtooth v1.2.6 documentation (n.d.). URL: https://sawtooth.hyperledger.org/docs/core/releases/latest/architecture.html.

Shi, Zeshun et al. (2019). "Operating permissioned blockchain in clouds: A performance study of hyperledger sawtooth". In: *Proceedings - 2019 18th International Symposium on Parallel and Distributed Computing, ISPDC 2019* June, pp. 50–57. DOI: 10.1109/ISPDC.2019.00010.

Uddin, Mueen (2021). "Blockchain Medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry". In: *International Journal of Pharmaceutics* 597.February, p. 120235. ISSN: 18733476.

DOI: 10.1016/j.ijpharm.2021.120235. URL: https://doi.org/10.1016/j.ijpharm.2021.120235.

Spengler, Ana Caroline Fernandes and Paulo Sérgio Lopes de Souza (2021). "Avaliação de desempenho do Hyperledger Fabric com banco de dados para o armazenamento de grandes volumes de dados médicos". In: pp. 61–72. DOI: 10.5753/wperformance.2021.15723.

Pajooh, Houshyar Honar et al. (2021). "Hyperledger fabric blockchain for securing the edge internet of things". In: *Sensors (Switzerland)* 21.2, pp. 1–29. ISSN: 14248220. DOI: 10.3390/s21020359.

Chacko, Jeeta Ann, Ruben Mayer, and Hans Arno Jacobsen (2021). "Why Do My Blockchain Transactions Fail?: A Study of Hyperledger Fabric". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data* 221, pp. 221–234. ISSN: 07308078. DOI: 10.1145/3448016.3452823.

Mohammed, Alaa Hamid, Alaa Amjed Abdulateef, and Ihsan Amjad Abdulateef (2021). "Hyperledger, Ethereum and Blockchain Technology: A Short Overview". In: June, pp. 1–6. DOI: 10.1109/hora52670.2021.9461294.

Dreyer, Julian, Marten Fischer, and Ralf Tönjes (2020). "Performance analysis of hyperledger fabric 2.0 blockchain platform". In: *CCIoT 2020 - Proceedings of the 2020 Cloud Continuum Services for Smart IoT Systems, Part of SenSys 2020*, pp. 32–38. DOI: 10.1145/3417310.3431398.

Nguyen, Minh Quang et al. (2021). "Understanding the Scalability of Hyperledger Fabric". In: URL: https://github.com/quangtdn/caliper-plus.

Xu, Xiaoqiong et al. (2021). "Latency performance modeling and analysis for hyperledger fabric blockchain network". In: *Information Processing and Management* 58.1, p. 102436. ISSN: 03064573. DOI: 10.1016/j.ipm.2020.102436. URL: https://doi.org/10.1016/j.ipm.2020.102436.

Gorenflo, Christian et al. (Sept. 2020). "FastFabric: Scaling hyperledger fabric to 20 000 transactions per second". In: *International Journal of Network Management* 30.5. DOI: 10.1002/NEM.2099.

World Health Organization (2020). ""Immunity passports" in the context of COVID-19". In: *WHO - Scientific brief* 71.April, pp. 1–2. ISSN: 15376591. URL: https://www.who.int/news-room/commentaries/detail/immunity-passports-in-the-context-of-covid-19.

Managing Health Data | CommonHealth (n.d.). URL: https://www.commonhealth.org/.

CommonPass | Digital Health App (n.d.). URL: https://commonpass.org/.

Q-Wallet - Take Control Of Your Personal Records (n.d.). URL: https://qservi.com/q-wallet/.

Home | Health Passport Worldwide (n.d.). URL: https://www.healthpassportworldwide.com/.

EU Digital COVID Certificate | European Commission (n.d.). URL: https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate_en.

The EU Digital COVID Certificate: EU has set a standard (n.d.). URL: https://ec.europa.eu/commission/presscorner/detail/en/ip_21_5267.

"eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 1" (2021). In:

"eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 2 European Digital Green Certificate Gateway" (2021). In: URL: https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-.

"eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 4 European Digital Green Certificate Applications" (2021). In:

"eHealth Network Guidelines on Technical Specifications for Digital COVID Certificates Volume 5 Public Key Certificate Governance" (2021). In:

"eHealth Network Guidelines on Technical Specifications for Digital Green Certificates Volume 3 Interoperable 2D Code" (2021). In:

CoWIN (n.d.). URL: https://www.cowin.gov.in/.

ADB (2018). "Adb Briefs". In: *ADB Briefs* 7.47, pp. 1–8. URL: https://www.adb.org/sites/default/files/publication/190600/overseas-employment-ban-workers.pdf.

IATA - Travel Pass Initiative (n.d.). URL: https://www.iata.org/en/programs/passenger/travel-pass/.

A COVID-19 health passport secured by blockchain to enable deconfinement | SICPA (n.d.). URL: https://www.sicpa.com/news/covid-19-health-passport-secured-blockchain-enable-deconfinement.

Digital Health Pass | IBM (n.d.). URL: https://www.ibm.com/products/digital-health-pass.

Androulaki, Elli, Ilie Circiumaru, et al. (2021). "IBM Digital Health Pass : A Privacy-Respectful Platform for Proving Health Status Whitepaper". In: i, pp. 1–10.

AOKpass: Secure - Private - Portable (n.d.). URL: https://www.aokpass.com/.

Malaysia's Immunitee Health Passport gains Singaporean verification (2021). URL: https://focusmalaysia.my/malaysias-immunitee-health-passport-gains-singaporean-verification/.

Go - Build fast, reliable, and efficient software at scale (n.d.). URL: https://go.dev/.

Owen, Sean (n.d.). *ZXing ("Zebra Crossing") barcode scanning library for Java, Android*. URL: https://github.com/zxing/zxing.

APPENDICES

APPENDIX A

Table 5: Comparative table of related work

Solutions	Available	Governamental	Developer	Infrastructure	Offline	Tested
CommonHealth	Yes	No	The Commons Project	Local	Yes	No
CommonPass	Yes	No	The Commons Project	Local	Yes	Yes
SICPA Health Passport	No	No	SICPA	Blockchain/PKI	Yes	No
IBM Digital Health Pass	Yes	No	IBM	Blockchain/PKI	Yes	No
Q-Wallet	Yes	No	Q-Servi	Local	Yes	Yes
Health Passport Worldwide	Yes	No	ROQU Group	Local	Yes	Yes
Eu Digital Covid Certificate	Yes	Yes	European Gov.	PKI	Yes	Yes
CoWin	Yes	Yes	Indian Gov.	PKI	Yes	No
IATA Travel Pass	Yes	No	IATA	Local	Yes	No
AOKPass	Yes	No	International SOS	Blockchain	No	No
Immunitee	Yes	Yes	Malaysian Gov.	Blockchain	No	No

B

APPENDIX B

Code 3: SmartContract of the application in Go programing language

```
package chaincode

import (
        "encoding/json"
        "fmt"

        "github.com/hyperledger/fabric-contract-api-go/contractapi"
)

type SmartContract struct {
        contractapi.Contract
}

type Certificate struct {
        HealthNumberHash       string `json:"healthNumberHash"`
        ImmunityDate           int    `json:"immunityDate"`
        ImmunityValidityInDays int    `json:"immunityValidityInDays"`
        Issuer                 string `json:"issuer"`
}

// Generate a new certificate
func (s *SmartContract) CreateCertificate(ctx
↪  contractapi.TransactionContextInterface, healthNumberHash string,
↪  immunityDate string, immunityValidityInDays int, issuer string) error {
        isHealthOrg, org := isHealthOrganization(ctx)
        if !isHealthOrg {
                return fmt.Errorf("Authorization denied! %s does not have
                ↪  permission to invoke this chaincode", org)
        }

        exists, err := s.CertificateExists(ctx, healthNumberHash)
        if err != nil {
                return err
        }
        if exists {
                return fmt.Errorf("the certificate %s already exists",
                ↪  healthNumberHash)
        }

        certificate := Certificate{
                HealthNumberHash:       healthNumberHash,
                ImmunityDate:           immunityDate,
```

```
39                  ImmunityValidityInDays: immunityValidityInDays,
40                  Issuer:                 issuer,
41          }
42          certificateJSON, err := json.Marshal(certificate)
43          if err != nil {
44                  return err
45          }
46
47          return ctx.GetStub().PutState(healthNumberHash, certificateJSON)
48  }
49
50  // ReadCertificate returns the certficate stored in the world state with given
    ↪  hash.
51  func (s *SmartContract) ReadCertificate(ctx
    ↪  contractapi.TransactionContextInterface, healthNumberHash string)
    ↪  (*Certificate, error) {
52          certificateJSON, err := ctx.GetStub().GetState(healthNumberHash)
53          if err != nil {
54                  return nil, fmt.Errorf("failed to read from world state: %v",
                    ↪  err)
55          }
56          if certificateJSON == nil {
57                  return nil, fmt.Errorf("the certificate %s does not exist",
                    ↪  healthNumberHash)
58          }
59
60          var certificate Certificate
61          err = json.Unmarshal(certificateJSON, &certificate)
62          if err != nil {
63                  return nil, err
64          }
65
66          return &certificate, nil
67  }
68
69  // CertificateExists returns true when certificate with given healthNumberHash
    ↪  exists in world state
70  func (s *SmartContract) CertificateExists(ctx
    ↪  contractapi.TransactionContextInterface, healthNumberHash string) (bool,
    ↪  error) {
71          certificateJSON, err := ctx.GetStub().GetState(healthNumberHash)
72          if err != nil {
73                  return false, fmt.Errorf("failed to read from world state: %v",
                    ↪  err)
74          }
75
76          return certificateJSON != nil, nil
77  }
78
79  func isHealthOrganization(ctx contractapi.TransactionContextInterface) (bool,
    ↪  string) {
80          org, err := ctx.GetClientIdentity().GetMSPID()
81          if err != nil {
82                  fmt.Errorf("failed to get client MSPID: %v", err)
```

```
83          }
84          if "HealthOrg" == org {
85                  return true, org
86          }
87          return false, org
88  }
```

C

APPENDIX C

Immunity Passport Ledger

Digital certificates implemented on a permissioned blockchain

Marco Oliveira[1][0000−0003−2189−8860], Tomás Honório[2][0000−0001−9914−3244], Catarina I. Reis[3][0000−0003−1529−629X], and Marisa Maximiano[4][0000−0002−1212−7864]

[1] Polytechnic of Leiria, Portugal 2192406@my.ipleiria.pt
[2] Polytechnic of Leiria, Portugal 2190338@my.ipleiria.pt
[3] ciTechCare - Center for Innovative Care and Health Technology, School of Technology and Management, Polytechnic of Leiria, Portugal
catarina.reis@ipleiria.pt
[4] Computer Science and Communication Research Centre (CIIC), School of Technology and Management, Polytechnic of Leiria, Portugal
marisa.maximiano@ipleiria.pt

Abstract. The global outbreak of Coronavirus (SARS-CoV-2) which in 2020 reached pandemic scale, has been a central topic of debate in our society. Concerns over the ease of transmission of the infection led to the imposition of measures restricting freedom such as curfews, lockdown, general confinement, and closure of trade. Technology was one of the tools used to resist to the spread of the disease using applications that, on one hand, track contacts to warn users that were close to someone infected and, on the other hand, provide immunity digital certification. Despite the relevance of these options, end users have no confidence, transparency, and responsibility that the registration and use of their health data are ethical, secure, anonymous, and available through verifiable credentials and, most importantly, is being used for its main purpose.

Consequently, a solution based on a distributed ledger technology, such as blockchain, is introduced to assure the trustworthiness and integrity of user's data. Since the proposed application embraced user privacy, we conducted a comparative study between permissioned blockchains, that includes an authorization abstraction layer and ensures that certain actions can only be performed by identifiable participants. We concluded that Hyperledger Fabric was an option that fulfilled all the requirements to develop a platform for the immunity passport ledger. Its modularity and versatility accommodates the needs that were initially proposed for the development of a proof of concept. The work leads us to propose that further research be conducted regarding scalability and performance evaluation.

Keywords: immunity passport ledger · blockchain · distributed ledger technology · hyperledger fabric

1 Introduction

On 31 December 2019, the first case of a new type of coronavirus was reported in Wuhan, China. On 30 January 2020, World Health Organization declared coronavirus outbreak a public health emergency of international concern. On 11 March 2020, the outbreak reached a pandemic scale, with 118 000 cases reported in 114 countries [1]. Since then, there has been an ongoing debate about how best to overcome the pandemic. Concerns about the ease of spread of the infection led to the imposition of measures restricting freedom such as lockdowns, curfews, general confinement, and trade closure.

Technology was one of the tools used to resist to the transmission of the disease through applications such as STAYAWAY COVID [2] that tracks contacts and notifies users that they were close to someone infected.

These applications raised concerns about user's data privacy and some countries were considering making these applications mandatory, which harmed their adoption by the population [3].

In December 2020 a global scale vaccination was started and a return to normality is expected. One of the options to help in this situation of "back to normal" was the creation of immunity passports that will allow people to prove their health status. This option was questioned and even contested by the World Health Organization [4].

This is why it is imperative to present an Immunity Passport that is transparent and secure that end users can trust, where the data is anonymous and used only for its purpose.

This paper is organized as follows. Section 2 presents an overview of Distributed Ledger Technologies, review, and analysis of related work. Section 3 gives the specifications for the implemented proof of concept. Section 4 summarizes the concerns we aimed to solve, the problems with similar implementations, and future work on the subject.

2 Related Work

Digital health certification is already a reality worldwide and are used broadly in European flights, through the Digital Green Certificate [5]. Travelling in Asia demands the CoWIN application [6]. They both provide digital and interoperable ways of validating vaccination certificates.

The private sector and other non-government organizations have urged the development of worldwide standardized vaccine certificates such as the CommonPass [7], and the IATA Travel Pass [8].

The use of blockchain technology is already gaining traction with the AOKpass [9] and Immunitee [10] as it is safe against tampered certificates or tests results.

2.1 Distributed Ledger Technologies

Distributed Ledger Technology (DLT) can be defined by the technological infrastructure and protocols that support concurrent access, validation, and state updating in an immutable manner across a network that's spread across multiple participants or nodes [11].

A DLT system has multiple participants which reach a settlement over a set of distributed data and its validity, in the absence of a central authority. What separates a DLT system from a traditional distributed database are the core features capable of transacting data and maintaining data integrity in the presence of malicious actors actively attempting to attack the network [12].

DLT has great potential to disrupt the way governments, institutions, and corporations work. It can help governments with tax collection, the issuance of passports, recording land registries and licenses, and the outlay of Social Security benefits as well as voting procedures [13]. Industries such as finance, music and entertainment, art, and supply chains of various commodities (including diamonds and other precious assets) are the early adopters of this technology [14].

There are multiple types of DLT such as DAG, Hashgraph, Holochain, Radix [15], and one of the most well-known: Blockchain. Big corporations such as IBM, Intel and Microsoft are exploring the technology and some of the distributed ledger protocols. Ethereum, Hyperledger Fabric, R3 Corda, and Quorum are amongst the most popular alternatives [11].

Blockchain is a DLT where transaction records are registered and kept in the ledger as a chain of blocks. This technology is attracting a great deal of attention propelled by the success of Bitcoin [16], launched in 2009 and triggering a large number of projects in different industries, with finance being the one that leads the use of this technology due to the success of cryptocurrencies. This technology underlies Bitcoin and has the potential to support a wide variety of business processes. Ethereum launched in 2015 by Vitalik Buterin [17] extends Blockchain concepts from Bitcoin but introduces smart-contracts that make possible execution of code in a decentralized way [18].

Hyperledger was launched in 2016 and is an industry-wide open-source initiative to advance blockchain technology, governed by The Linux Foundation [19].

Permissioned and private Blockchain differ from public Blockchains of Bitcoin and Ethereum. These Blockchain applications rely on trust relationships between participant organizations, with the need to share data with a greater degree of security. Contract privacy is a mandatory requirement for enterprise applications, and using a permissioned or private blockchain ensures it. Data for these Permissioned blockchains can be used to record promises, trades, transactions, or any data that can't be lost without needing to run a Proof-of-Work mechanism [20].

Hyperledger Fabric is an enterprise-grade open-source permissioned blockchain framework for developing solutions and applications with a modular architecture. Its modular and versatile design satisfies a broad range of industry use

cases [21–23] and allows components, such as consensus and membership services, to be plug-and-play. The unique approach to the consensus mechanism enables performance at scale while preserving privacy [24].

The consensus mechanism has a fundamental role in the transaction flow of Fabric that goes through the process of a transaction proposal, endorsement, ordering, validation, and commitment [19].

Block sequencing and transaction sequencing within blocks are established when the ordering service first creates blocks. Each block contains a sequence of transactions representing a query or update to the world state.

Fabric has been designed at its core to have a modular architecture and meet the diversity of enterprise use case requirements. It allows pluggable identity management protocols such as LDAP or OpenID Connect, and the ledger supports a variety of DBMSs, and pluggable endorsement and validation policy enforcement [25].

The performance of Fabric has been further improved [26], during the release of new versions, with a substantial increase in performance on Fabric 2.0 [27]. The performance of a blockchain is subjected to many implementation variables such as transaction size, block size, network bandwidth, hardware limitations, consensus algorithm, caching, and parallelism, among others [28, 29].

Out of the box with simple configurations, Fabric can support 3000 transactions per second, and with advanced tweaks. A study [30], demonstrates that is possible to scale Fabric performance up to 20000 transactions per second.

2.2 Applications

Digital green certificate is a collaborative effort of health care authorities across the EU and consists of a Private Key Infrastructure (PKI) solution with a single authority.

This solution is not ideal since the key pairs are not issued to single health care professionals but rather by big organizations. A single breach of the private key will bring down every certificate signed by that authority. Then, all the certificates will be mark as untrusted, which risks blocking traveling in Europe. Malicious hackers are targeting these keys because of their value to generate new fake certificates and sell them on the black market [5].

CoWIN is the platform created by the Indian government to manage the appointment of vaccinations and the emission and validation of vaccine certificates. This solution is also based on a PKI system and allows offline verification. The infrastructure is centralized and under a unique central authority [6].

CommonPass is a digital health pass developed by The Commons Project, a non-profit organization with support from the Rockefeller Foundation [7]. This application serves as proof of vaccination for a wide range of airlines. The certificate is stored on the local device.

IATA Travel Pass is a mobile app that stores and manages verified certifications for tests or vaccines [8]. It gives the user a digital wallet to store all their travel documentation, including biometric passports. This application al-

lows travelers to plan their journeys accordingly to the health conditions of the destination country.

AOKpass [9] is a platform and mobile application using blockchain technology, enabling users to verify their health status with third parties while preserving the privacy of their underlying personal health data. Users have exclusive control over their health data, such as health certificates or test results. They are stored only on the user's mobile device and never on any external database or centralized system. AOKpass saves a hash on the Ethereum public blockchain network for the certificate data of the users.

Immunitee Health Passport is the Malaysia health passport accepted into Singapore [10]. It was designed to store personal immunization records and vaccine data with the Unifier platform, allowing interoperability to securely share the necessary data with the various national health check systems being put globally.

Immunitee stores all patient data hashed on a public blockchain system, ensuring that data cannot be tampered, is protected, and belongs to the user. Verifiers can only obtain information by scanning a QR Code that holds all the relevant testing and vaccination data and can only be unlocked using a secret key that belongs to the user.

IBM Digital Health Pass uses a blockchain framework, Hyperledger Fabric. The implementations consist of administration authorities providing X.509 certificates to healthcare organizations. Then they use their private key to sign the user's health certificates. Only the keys supplied to healthcare organizations are stored on the ledger. Verifier uses a QR code reader to extract and decode the issuer identifier. Verifier queries the ledger to obtain the public key associated with issuer identification. Using the latter, the verifier checks the certificate's digital signature and gets assured that the claimed issuer indeed generated the health certificate. [31, 32].

3 Immunity Passport Ledger

The proposed solution for an Immunity Passport Ledger considers the chief requirement of allowing a consortium of government and private agencies to collaborate through a single system to store and validate the information. Most importantly, it requires keeping some information hidden from consortium participants while allowing data to be audited and managing access to the data.

Hyperledger Fabric provides flexibility, modularity, scalability, and performance. These characteristics allow our main use cases:

- Allow health organizations and vaccination centers to issue and verify certificates.
- Allow individuals to choose with whom they share their information with.
- Make information verifiable to a broad range of organizations.
- Allow statistics organizations to get insights on the issued and expired certificates.

3.1 Use case

Application use cases start after the user physically visits an authorized issuer organization, provide a traditional mean of identification with health number and perform a test or vaccination. User data and validity of immunity are submitted to the ledger, and a hash is generated.

Issuance of a QR Code in a digital and/or paper format containing the hash that validates the immunity of the user.

A user presents the QR Code to an authorised verifier organisation member. Then, the verifier scans the QR Code and obtains the information on the validity of the immunity and the name of the person associated with the certificate, and the person presents an identifying document to prove certification ownership.

If, by any chance, a person loses his certificate, a reissue request is available through the health organization platform.

3.2 Data access

Access to data across the blockchain participants ranges accordingly to the needs of the work performed by each organization as shown on Table 1. Health organizations have access to all the data and are the only organization allowed to submit new transactions on the ledger. Verifier organizations will only perform read operations on the ledger and only need access to immunity status and name to validate if the certificate is valid or invalid and user ownership of the certificate. Statistic organizations will perform read operations to get insights on immunity status and validity of the population.

Table 1. Data access

Data	Health org	Verifier org	Statistics org
Immunity date	**X**		**X**
Immunity validity	**X**	**X**	**X**
Name	**X**		
Health number	**X**	**X**	

3.3 Network architecture

The network for the application (Fig. 1) consists of a single channel common on all organizations with the main purpose of conducting transactions, an orderer service, a peer for each organization (that holds a copy of the ledger) with a world state database and the executable chaincode to perform actions on the ledger. The policy rules are specified on the channel configuration represented by the CC1 in the Fig. 1. All organizations in the network endorse these rules.

Health organizations will keep user-sensitive data in a private state database off-chain known as world state database. This Fabric functionality allows all

channel participants to see a transaction while keeping a portion of the data private. Private data will be transmitted peer-to-peer between authorized organizations only, via the gossip protocol. A hash of the data, is endorsed, ordered, and written to the ledgers of every peer that participates on the channel. Hash is used as evidence of the transaction, is used for state validation, and for audit purposes.

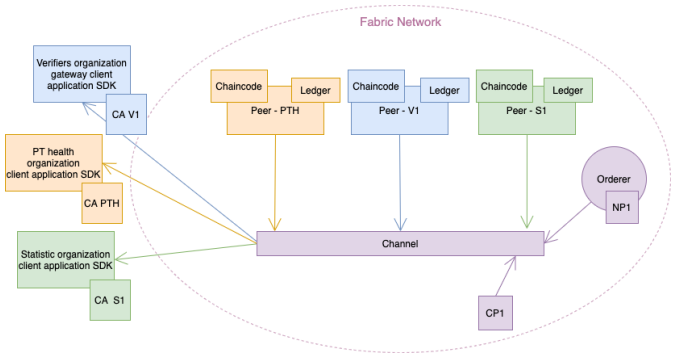

Fig. 1. Proposed Network architecture.

3.4 Data structure

CouchDB holds the world state database and allows ledger states to be structured as JSON documents. This optimization enables JSON queries against stored data values instead of the default approach of using LevelDB, where its primary purpose is to query the keys.

Regarding the data structure, it consists of a JSON object that represents the user certificate. The data ensures the connection between the individual and the immunity validation (certificate status). It also allows statistic organizations to perform metric reports based on dates and health organizations to send certificate expiration notices in advance.

3.5 Application structure

The proposed application has two main components: a web and a mobile application for organizations to manage and issue certificates and an additional component that holds the chaincode installed at the organization's nodes. The two applications invoke the chaincode to update and read the ledger.

Applications use Hyperledger Fabric SDK to interact with the Fabric blockchain network. It provides a gateway module to manage the network interactions and an API to submit transactions or query the ledger. The application stores multiple wallets on the filesystem for each health professional with identities to connect to the network.

Health organizations' platforms allow health care professionals to register user data on the blockchain and generate certificates. The interface provides authentication to health care professionals and a mechanism to generate certificates. After submission to the blockchain, a QR code is generated from the hash and sent to the patient.

Verifiers organizations chaincode search on blockchain by a hash and return valid and health number or not valid. Verifiers have an Android or iOS application that reads QR code and sends the hash to a backend that invokes the smart contract. Then it shows the health number and validity status, or not found in the case certificate doesn't exist on the ledger.

Statistics organizations chaincode receives a date interval and returns the number of certificates expiring between that date also returns the number of certificates issued between that date. This web platform generates reports and allows analysts to visualize data about issued and expired certificates.

4 Conclusion

Providing privacy to a massive number of certificates issued while ensuring scalability can be challenging.

Some of the existing centralized solutions on the cloud can be costly and delegate the data ownership to third parties. Infrastructures with a single point of failure are a significant issue that can suspend global traveling. Private Key Infrastructure solutions aim to solve this problem and verify certifications without an Internet connection, but this raises a more significant issue with a flood of tampered certificates online due to leaks of private keys. Usage of public blockchains can be expensive due to transactions costs and do not assure privacy.

We developed a proof of concept that aims to solve these issues using permissioned blockchain technology with a distributed ledger between multiple organizations that agree on a single source of truth. In future work, we must test the performance and scalability of the proof of concept application simulating a real scenario of use at a global scale.

References

1. WHO. (2020). WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. Retrieved from https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19—11-march-2020
2. STAYAWAY COVID - Fique longe da COVID num clique. (n.d.). Retrieved August 2, 2021, from https://stayawaycovid.pt/
3. Miranda Ramos, L. F. (2020). Digital contact tracing and data protection: assessing the French and Portuguese applications. UNIO – EU Law Journal, 6(2), 35–48. https://doi.org/10.21814/unio.6.2.2767

4. “Immunity passports” in the context of COVID-19. (n.d.). Retrieved October 13, 2021, from https://www.who.int/news-room/commentaries/detail/immunity-passports-in-the-context-of-covid-19
5. EU Digital COVID Certificate — European Commission. (n.d.). Retrieved August 24, 2021, from https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate_en#does-it-matter-which-vaccine-citizens-received
6. CoWIN. (n.d.). Retrieved August 24, 2021, from https://www.cowin.gov.in/
7. CommonPass — Digital Health App. (n.d.). Retrieved August 24, 2021, from https://commonpass.org/
8. IATA - Travel Pass Initiative. (n.d.). Retrieved August 24, 2021, from https://www.iata.org/en/programs/passenger/travel-pass/
9. AOKpass: Secure - Private - Portable. (n.d.). Retrieved August 24, 2021, from https://www.aokpass.com/
10. Malaysia’s Immunitee Health Passport gains Singaporean verification. (n.d.). Retrieved August 24, 2021, from https://focusmalaysia.my/malaysias-immunitee-health-passport-gains-singaporean-verification/
11. Frankenfield, J. (2018). Distributed Ledger Technology Definition. Investopedia, (October). Retrieved from https://www.investopedia.com/terms/d/distributed-ledger-technology-dlt.asp
12. Rauchs, M., Glidden, A., Gordon, B., Pieters, G. C., Recanatini, M., Rostand, F., Vagneur, K., Zhang, B. Z. (2018). Distributed Ledger Technology Systems: A Conceptual Framework. SSRN Electronic Journal, August. https://doi.org/10.2139/ssrn.3230013
13. Hjálmarsson, F. ., & Hreioarsson, G. K. (n.d.). Blockchain-Based E-Voting System.
14. McLean, S., & Deane-Johns, S. (2016). Demystifying Blockchain and Distributed Ledger Technology – Hype or Hero? Computer Law Review International, 17(4), 1–8. https://doi.org/10.9785/cri-2016-0402
15. Different types of dlts and how they work — by TerraGreen — Medium. (n.d.). Retrieved August 25, 2021, from https://medium.com/@support_61820/different-types-of-dlts-and-how-they-work-cfd4eb218431
16. Velde, F. R. (2013). Bitcoin - A Primer. Chicago Fed Letter, (December), 1–4. Retrieved from http://search.ebscohost.com/92563197&site=ehost-live
17. Ethereum Whitepaper — ethereum.org. (n.d.). Retrieved October 13, 2021, from https://ethereum.org/en/whitepaper/
18. Buterin, V. (2014). A next-generation smart contract and decentralized application platform. Etherum, (January), 1–36. Retrieved from http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf
19. Hyperledger – Open Source Blockchain Technologies. (n.d.). Retrieved August 16, 2021, from https://www.hyperledger.org/
20. Androulaki, E., Barger, A., Bortnikov, V., Muralidharan, S., Cachin, C., Christidis, K., ... Yellick, J. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the 13th EuroSys Conference, EuroSys 2018 (Vol. 2018-Janua). ACM. https://doi.org/10.1145/3190508.3190538
21. Uddin, M. (2021). Blockchain Medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry. International Journal of Pharmaceutics, 597(February), 120235. https://doi.org/10.1016/j.ijpharm.2021.120235
22. Spengler, A. C. F., & Souza, P. S. L. de. (2021). Avaliação de desempenho do Hyperledger Fabric com banco de dados para o armazenamento de grandes volumes de dados médicos, 61–72. https://doi.org/10.5753/wperformance.2021.15723

23. Pajooh, H. H., Rashid, M., Alam, F., & Demidenko, S. (2021). Hyperledger fabric blockchain for securing the edge internet of things. Sensors (Switzerland), 21(2), 1–29. https://doi.org/10.3390/s21020359
24. Chacko, J. A., Mayer, R., & Jacobsen, H. A. (2021). Why Do My Blockchain Transactions Fail?: A Study of Hyperledger Fabric. Proceedings of the ACM SIGMOD International Conference on Management of Data, (221), 221–234. https://doi.org/10.1145/3448016.3452823
25. Mohammed, A. H., Abdulateef, A. A., & Abdulateef, I. A. (2021). Hyperledger, Ethereum and Blockchain Technology: A Short Overview, (June), 1–6. https://doi.org/10.1109/hora52670.2021.9461294
26. The Ordering Service — hyperledger-fabricdocs master documentation. (n.d.). Retrieved August 16, 2021, from https://hyperledger-fabric.readthedocs.io/en/release-2.2/orderer/ordering_service.html
27. Dreyer, J., Fischer, M., & Tönjes, R. (2020). Performance analysis of hyperledger fabric 2.0 blockchain platform. CCIoT 2020 - Proceedings of the 2020 Cloud Continuum Services for Smart IoT Systems, Part of SenSys 2020, 32–38. https://doi.org/10.1145/3417310.3431398
28. Nguyen, M. Q., Loghin, D., Tuan, T., & Dinh, A. (n.d.). Understanding the Scalability of Hyperledger Fabric. Retrieved from https://github.com/quangtdn/caliper-plus
29. Xu, X., Sun, G., Luo, L., Cao, H., Yu, H., & Vasilakos, A. V. (2021). Latency performance modeling and analysis for hyperledger fabric blockchain network. Information Processing and Management, 58(1), 102436. https://doi.org/10.1016/j.ipm.2020.102436
30. Gorenflo, C., Lee, S., Golab, L., & Keshav, S. (2020). FastFabric: Scaling hyperledger fabric to 20 000 transactions per second. International Journal of Network Management, 30(5). https://doi.org/10.1002/NEM.2099
31. Digital Health Pass — IBM. (n.d.). Retrieved August 28, 2021, from https://www.ibm.com/products/digital-health-pass
32. Androulaki, E., Circiumaru, I., Vico, J. D., Prada, M., Sorniotti, A., Stoecklin, M., ... Wallace, M. (2021). IBM Digital Health Pass : A Privacy-Respectful Platform for Proving Health Status Whitepaper, (i), 1–10.

5 Apendix A

Table 2. Comparison table of related work

Solutions	Available	Governmental	Infrastructure	Data Privacy	Offline mode
D.G.C.[a]	Yes	Yes	PKI	Yes	Yes
CommonPass	Yes	No	Local	Yes	Yes
IATA T.P.	Yes	No	Local	Yes	Yes
AOKpass	Yes	No	Blockchain	Yes	Yes
Immunitee	Yes	Yes	Blockchain	Yes	No
IBM D.H.P.	Yes	No	Blockchain/PKI	Yes	Yes

[a]Digital Green Certificate

D

APPENDIX D

Building trust with a contact tracing application: a blockchain approach

Tomás Honório[1][0000−0001−9914−3244], Catarina I. Reis[2][0000−0003−1529−629], Marco Oliveira[4][0000−0003−2189−8860], and Marisa Maximiano[3][0000−0002−1212−7864]

Computer Science and Communication Research Centre (CIIC), School of Technology and Management, Polytechnic of Leiria, Portugal

Abstract. On March 11, 2020, the novel coronavirus (COVID-19) was declared a global pandemic. With no treatment or vaccine available at the time, it was necessary to rely on non-pharmaceutical methods for case identification and contact tracing. This kind of approach has good results in detecting and preventing tuberculosis, sexually transmitted infections, and vaccine-preventable diseases. Contact tracing and keeping safe distances are crucial to containing the spread of COVID-19. Nonetheless, contact tracing is a complex intervention, it involves quarantining and investigating close contacts. Manual contact tracing methods are slow, require a large amount of effort, and more often than not rely on the memory or assumptions of individuals. To combat these downsides, contact tracing applications were developed, resulting in quicker and more reliable recognition of infected individuals. However, because of the complex nature of these applications and their lack of transparency, a large portion of the population started doubting the privacy of the data collected. Soon after, many of these applications started to dwindle in the user department, which caused a feedback loop. "If fewer people are using the application, the application itself becomes useless, and there is no longer a reason to use it." Is clear that the main issue behind their downfall was an overwhelming lack of trust. In response, this paper will analyze how the use of blockchain technology can help the development of a more transparent application. And describe how a proof of work based on this concept was implemented. On the same note, it will also approach why was *Hyperledger Sawtooth* chosen, instead of more popular solutions such as *Bitcoin* or *Ethereum*.

Keywords: COVID-19 · Contact Tracing · Bluetooth · Distributed Ledger · Blockchain · Bitcoin · Ethereum · Hyperledger · Hyperledger Sawtooth · Directed Acyclic Graph

1 Introduction

Over the recent COVID-19 epoch, it has become clear that the once acclaimed tracking applications have mostly turned out to be a letdown [1]. That being said, the question is why did it end up like that, and what led this type of application to be abandoned.

The COVID-19 pandemic caused a global health crisis that no one was prepared to face, resulting in millions being infected and the appearance of new, more infectious strains [2]. It is now clear that there is a need for infection tracking, and since this is such a significant process, it needs to be done reliably and in real-time [3]. However, decision-making has proven to be a daunting challenge for both authorities and the public.

To better understand the dynamics of this pandemic and provide effective countermeasures, data with quality is a must. That being said, the search for trustworthy data has led to the appearance of contact-tracing applications, apps that in theory should facilitate the management/tracking of new cases. However, it is apparent that in the vast majority of cases, the result was somewhat of a letdown.

According to [4], the main reason for the low adherence was not technological limitations, but the concern with privacy and the negativity surrounding this type of application. One of the leading factors of this sentiment was the direct connection between these applications and big tech giants such as Google and Apple. A vast portion of contact-tracing applications was based on protocols provided by these companies, giving them the power to change them overnight. These are the same companies that come to blows repeatedly with entities like the *EU Court of Justice* because of privacy concerns.

If possible, a user should be able to identify where and how their data is being used. The proposal is to solve the lack of trust by relying on transparent systems, such as solutions based on *Distributed Ledger*, more precisely Blockchain.

This paper follows the order described below:

- Introduction, describing the problem and the basis for the solution.
- State of the art, describing how it is done and what other implementations exist.
- Discussion, describing what is the reason *Hyperledger Sawtooth* was chosen, and how it competes with existing solutions.
- Contact Tracing using *Hyperledger Sawtooth*, describing why and how a Contact Tracing Application based on *Sawtooth* works and what is their workflow.
- Conclusion, the last section entails an overview of the solution proposed, and why it could be a valid solution for the problems initially raised.

2 State of the Art

To better understand the context behind this development, some key points need to be fleshed out, these are: What is a Distributed Ledger, what is Contact Tracing, what kind of application is currently available, and at last, is there any application that tries to accomplish the same goal.

2.1 Distributed Ledger

The concept of Distributed Ledgers has been gaining attention over the past few years, in part because of the emergence/spread of new blockchain technolo-

gies based on *Bitcoin* [5] and *Ethereum* [6]. That being said, blockchain is just one of the many forms a Distributed Ledger may have, and while *Bitcoin* and *Ethereum* have been pretty popular, blockchain goes beyond that, an example of this, is something like Hyperledger. Hyperledger is an umbrella project focused on developing tools, frameworks, and libraries for enterprise-level blockchain implementations [7]. Some of the most well know projects that resulted from it are Hyperledger Fabric, *Hyperledger Sawtooth*, Hyperledger Grid, and many many more. That being said, *Hyperledger Sawtooth* is of special interest, thanks to its flexibility and ease of use. Being the main advantages, the ability to specify business rules without requiring extensive knowledge of the underlying design, followed by having better parallel transaction scalability when compared with *Bitcoin* and *Ethereum*.

2.2 Contact Tracing

Contact Tracing can be described as using collected data from people diagnosed with an infectious disease, to recognize and provide support to new infectious individuals [8].

By enabling people to know that they may have been infected, it's possible to monitor their health for symptoms. The World Health Organization (WHO) defines three crucial steps for any form of contact tracing [9]:

- Identification - Upon confirmation of an infected individual, all of his contacts must be identified. This is achieved by analyzing the habits and activities of the infected individual.
- Listing - All people who have come into contact with an infected person must be listed as a "Contact" and informed of their status.
- Follow-up - Contacts should be observed regularly to monitor the onset of symptoms/complications.

According to [10], close contacts who spent over 15 minutes in contact with an infected person are of special interest, since they are more likely to be infected. By testing these individuals, it's possible to delay the progress of the transmission chain.

As a benchmark for a successful contact tracing (COVID-19) operation, WHO suggests locating/quarantining 80% of close contacts within 3 days (after the first case is confirmed). According to Christophe Fraser (Oxford University), transmission is extremely fast and the virus can spread before any action is taken [11]. Even if all cases were discovered/isolated within three days, the pandemic will continue to grow. To prevent such an outcome, 70% of cases need to be isolated on the first day, only then, can the outbreak be significantly reduced.

Mukhi et al [12], advocates that traditional methods of surveillance and data collection (employing a paper-based method), pose many challenges, such as data loss, duplication, difficulty in managing databases, and lack of timely access to the data. That being said, the only way to gather such information in such a brief amount of time is by relying on technology. As mentioned by Mukhi et

al [12], "COVID-19 pandemic infection/death rates may slow down in countries with robust vaccination coverage. However, on a global scale, new mutations and new pathogens will continue disrupting society for ages to come, and therefore, it is important to keep advancing in this field".

Contact tracing applications allow for a quicker and more reliable identification of newly infected individuals. Thanks to this, the quality of the data collected is also greatly improved, being this the main purpose for the development of such apps in several countries.

2.3 Conventional Applications

Using the Portuguese application as an example. The development was a collaboration between INESC TEC, ISPUP, Keyruptive, and Ubirider, where companies and research institutions joined efforts to develop the application Stay Away Covid [13]. This application has the purpose of detecting (through the use of a smartphone) if users were near someone infected with COVID-19. It works by announcing its presence to all nearby devices, using random identifiers. By doing this, the application recognizes who has been in contact with the user, for a contact to be of interest, it needs to be within two meters and for at least 15 minutes [13]. In case of infection, the user will receive a code to be inserted into the application, after which they will notify the contacts. One of the key points is that it works without recording personal information, being the sole exception, infected users. However, according to the article "60% have already deleted StayAway Covid" [14], thus the application has been losing users at an alarming rate, being the peak of users during October 2020. Around the same time, a bill was drafted to mandate the use of the application [15], but later on, end up being rejected. Since then, the number of users has been steadily declining and in January 2021 only 39% of the nearly three million people who installed the application continued to use it.

In hindsight, it is possible to point out two main issues: they are lack of trust/concerns with data privacy and a lack of coordination. Lack of trust can be pointed to as the fundamental issue behind the slow adherence and fast decline of the application. There are many reasons for this sentiment, but a glaring one is a reliance on technology developed by Google and Apple. Both of which are known for possessing their fair share of privacy issues [16, 17].

Lack of coordination did also play a big role in the application's failure, quoting [14]: "in the last five months, only 2708 codes were used".

The problem was caused by a lack of knowledge since many of the infected did not know where to get the codes needed for the app, and many health professionals dint know where to find them

Some examples of the overall mistrust are:

- "People are losing confidence in the app because there are no codes";
- "And there are no codes because doctors are poorly informed about how the app works and where the codes are provided";
- "Since the application was launched, doctors have contacted us for help. It shouldn't be like that."

2.4 Contact Tracing using Distributed Ledgers

According to [18], "engineers from the University of Glasgow outline how a trustworthy contact tracing system could be built on the unique properties of distributed ledger technology." In the same article, [19] is mentioned how traditional methods pale in comparison with new, more technologically advanced ways of doing contact tracing. And how an application like the TraceTogether helped the government of Singapore to contain their coronavirus outbreak [20].

Simultaneously, it was proposed a new mobile-based system named BeepTrace [19], which hopes to "harness the unique properties of distributed ledgers to create a decentralized, potentially international system to help break the chains of virus transmission." It works by assigning a randomly generated ID, which changes regularly to prevent tracking or identification. If a test gives a positive for COVID-19, it shares the IDs gathered by the application over the past 14 days with the ledger. Notifying any user that was in contact during the period.

In the same paper, it is also mentioned how the application possesses two different modes, a passive and an active. The passive mode works by relying on GPS information to gather where the user was and who was the user in contact with. The active mode works by relying on Bluetooth to register users in close contact for a prolonged period.

3 Discussion - The Future of Contact Tracing

The Portuguese application is only one of many examples where social factors remain the driving force behind the lack of adherence. To combat the negative sentiment, some key points need to be taken into consideration, they are:

- Data transparency: a user should be able to identify where and how their data is being used;
- Individual control of data: a user should be able to choose what information is being shared;
- Decentralized chain of power: it should not be fully reliant on the goodwill of a company (Google or Apple) to work.

That being said, the proposed way to combat all these issues is by shifting the application from a centralized chain of power to a decentralized system. Where each user has control over the information and can freely interact with it, instead of it being a black box. A more effective system to accomplish this is most likely a *Distributed Ledger* or more accurately, an application that relies on Blockchain technology to store the user information.

Although the use of distributed ledgers has already been approached in studies like the one mentioned in [19], there is still room for improvement. One of the key differences between the developed approach and the one described in the paper mentioned above is the underlying technology. This approach was based on a Chain-based solution, more precisely *Hyperledger Sawtooth* instead of a Directed Acyclic Graph (DAG) based solution.

DAG based solutions - A DAG based solution represents an alternative to the traditional blockchain that aims to improve speed, scalability, and cost. However, it is still a distributed ledger. It relies on a graph structure with directed edges and where no vertex should lead back to himself. Usually, there are no blocks, being the most considerable difference the way they add transactions to the network [21], according to [22] because of the adoption of graph structure, it can do the processing of transactions in parallel. This is in direct contrast to the sequential manner used in chain-based solutions.

Chain-based solutions vs Directed Acyclic Graph based solutions - When comparing Chain-based solutions vs DAG-based solutions, it is possible to recognize some key differences. Chain-based solutions offer transparency and immutability, but lack scalability when it comes to performance since it often relies on consensus algorithms like Proof of Work [23]. These algorithms not only limit the number of transitions that can be processed in a given amount of time but also consume copious amounts of computing power and electricity. Establishing them as not the best-suited solution for high volumes of transactions. DAG offers more efficient scaling and the reduction/avoidance of fees, but as a result, this may make it vulnerable to attacks. That is why many of the DAG based solutions have to rely on centralized features like central co-ordinators, pre-selected validators, 'witness' nodes, or completely private network systems.

4 Contact Tracing using *Hyperledger Sawtooth*

The goal behind this development is to have a contact tracing application capable of leveraging blockchain technology and with this, increase trust and adherence from the end-user. The key requirements for the development are:

- Being able to detect if two users are in close contact;
- In case of a user infection, share this information with the public ledger;
- Fetch information about infections to notify its contacts.

For this implementation, the blockchain technology chosen was *Hyperledger Sawtooth*. It offers freedom when choosing the permission level (either permissioned or permissionless). It gives the choice between a vast amount of different consensus algorithms, making sure that at least one suits the needs. Parallel processing of transactions provides a performance improvement for every workload by reducing overall latency effects, which occur when transaction execution is performed serially. And at last, it provides an SDK in multiple languages, allowing the abstraction of much of the work.

By looking at the diagram 1, it is possible to see the proposed architecture for this development. On the left, there are two users in contact, both with the app installed and the Bluetooth enabled, followed by two arrows repressing the IDs being shared between one another. If at any moment, one of the users is diagnosed with COVID-19 and it decides to share the information, the application will send a request with all IDs gathered in the past 14 days into the Validator Node, being their final destination the *Sawtooth Network*.

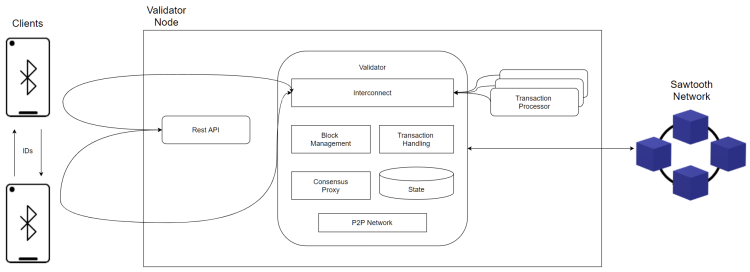

Fig. 1. *Sawtooth* Architecture Diagram [24]

4.1 Detection of close contacts

As the name entails, the primary function behind the application is to contact trace, something that is achieved by making use of proximity tracking. More accurately, it uses the AltBeacon protocol [25] in order to gather the distance between two users. For this to work, the app relies on a beacon advertising in the background, and a tracker to identify other users.

The content of the emitted message allows the receiving device to get the user ID and compute the relative distance between each other [25]. This ID is generated every couple of hours to keep the user's privacy.

For a new contact to be recorded a couple of procedures need to happen: First, the user needs to have done the application setup (Application installed, Bluetooth turned on, etc). Second, the user needs to be in contact with another user that has also completed the setup. Third, for a contract to be deemed of interest, it needs to be for an extended period (around 15 minutes) and within two meters of distance. Afterward, this information is stored locally in a database for at least 14 days. The reason that lead to this choice, was it being the minimum amount of time to ensure no symptoms were present, and also, being the suggested time for a person to be in self-quarantine after exposure to an infected individual [26].

4.2 Identifying and sharing in case of infection

For a user to be deemed infected, he's required to have a positive test result for COVID-19. Then, a code provided by the health authorities allows the user to share its state information with the remaining network. This is achieved by providing the ledger with the IDs collected over time by the user. Something that is done by doing a request to the *Sawtooth Rest API*. After this, the request is validated by the *Sawtooth Validator* and afterward provided to the *Transaction Handler*.

The *Transaction Handler* is what contains the business logic for a particular family of transactions. And on the same note, transaction families can be seen

as the way *Hyperledger Sawtooth* implements smart contracts. That being said, once the *Transaction Handler* gets a hold of the information, it will confirm if everything is alright, and if so, save this new information into the ledger state.

4.3 Notifying about possible infections

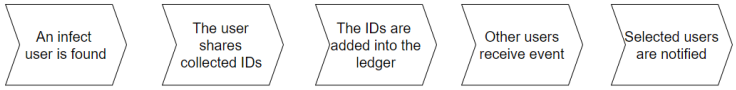

Fig. 2. Notification Process

After everything is set and done, ensuring that everyone that was in contact with the infected user is rightfully notified is of the utmost urgency. To achieve this, the application makes use of a handy feature provided by *Sawtooth* called *Sawtooth Events*. They occur when new blocks are committed and result from the validator broadcasting an event when a commit operation succeeds. By using *Sawtooth Events*, it's possible to notify the user about what exactly was changed inside the ledger, and with this on the same note filter only the relevant information for this use case. That being said, once the information of a new infection is added to the Ledger, everyone can fetch it and see if they are part of the affected users. If so, the app will send a notification to the user, asking for this to go to a testing center, to be tested for COVID-19.

As is shown on Image 2, the steps are, first an infected user is found, then it shares the collection of IDs with one of the Validator Nodes, followed by, them being added to the ledger. Once this is done, all users that need to be notified will be notified to get tested.

5 Conclusion

Although there are multiple examples of contact tracing applications, it is clear that there was a widespread lack of trust in them. This resulted from issues like the heavenly reliance on private companies like Google and Apple, the lack of transparency about how the data is being used, the inability to choose what was shared, and sometimes trying to impose its use without addressing the underlying issues. This lack of trust was the main reason for the lack of adoption by the general population, and the fast decline in users after the first issues became widespread. Resulting in a feedback loop where the fewer users actively using it, the less useful it is, followed by even fewer users using it.

By using blockchain technology, it is possible to solve many of the concerns related to the mistreatment of data, allowing every user to check what information is being used and how is it being handheld.

It also needs to be pointed out that Hyperledger is a significant performance improvement in the handling of parallel transactions when compared with *Bitcoin* and *Ethereum*. With the appearance of new alternatives like for example *Hyperledger Sawtooth*, it is now possible to develop large-scale applications without having to fear performance issues or suffering the shortcomings of alternative technologies like DAG-based solutions. By making use of the *Sawtooth SDK*, it is possible to ease the development cost, since many of the modules needed for such applications are already present.

As a future improvement, the addition of location data would be an interesting challenge, since there are so many ways to go about his issue. He can approach this by gathering GPS data, like how it is done in the BeepTrace project [19], or by creating a location journal, where points of reference (e.g. Restaurants and malls) are stored by making use of QR-codes.

In the end, the development of a contact tracing app allowed for a better understanding of the workflow involved in contact tracing and the challenges related to a distributed solution.

Acknowledgments This publication is funded by FCT-Fundacão para a Ciência e Tecnologia, I.P., under the project UIDB 045242020

References

1. Chan, E.Y., Saqib, N.U.: Privacy concerns can explain unwillingness to download and use contact tracing apps when COVID-19 concerns are high. Comput. Human Behav. 119, 106718 (Jun 2021)
2. CDC: What you need to know about variants. https://www.cdc.gov/coronavirus/2019-ncov/variants/variant.html (Sep 2021), accessed: 2021-10-9
3. Shelby, T., Schenck, C., Weeks, B., Goodwin, J., Hennein, R., Zhou, X., Spiegelman, D., Grau, L.E., Niccolai, L., Bond, M., Davis, J.L.: Lessons learned from COVID-19 contact tracing during a public health emergency: A prospective implementation study. Front Public Health 9, 721952 (Aug 2021)
4. Bambauer, J., Ray, B.: Covid- 19 apps are terrible — they don ' t have to be. https://s3.documentcloud.org/documents/20424830/bambauer-and-ray-final-2.pdf (Nov 2020), accessed: 2021-9-26
5. Nakamoto, S., bitcoin.org, W.: Bitcoin: A Peer-to-Peer electronic cash system
6. Ethereum whitepaper. https://ethereum.org/en/whitepaper/, accessed: 2021-10-9
7. Bhanushali, H., Arthena, A., Bhadra, S., Talukdar, J.: Digital certificates using blockchain: An overview (Apr 2019)
8. COVID19-contact-tracer-508.pdf (Jul 2020)
9. Infection prevention and control: Contact tracing. https://www.who.int/news-room/q-a-detail/contact-tracing (May 2017), accessed: 2021-9-26
10. Lewis, D.: Why many countries failed at COVID contact-tracing — but some got it right. https://www.nature.com/articles/d41586-020-03518-4 (Dec 2020), accessed: 2021-10-5

11. Abueg, M., Hinch, R., Wu, N., Liu, L., Probert, W., Wu, A., Eastham, P., Shafi, Y., Rosencrantz, M., Dikovsky, M., Cheng, Z., Nurtay, A., Abeler-Dörner, L., Bonsall, D., McConnell, M.V., O'Banion, S., Fraser, C.: Modeling the combined effect of digital exposure notification and non-pharmaceutical interventions on the COVID-19 epidemic in washington state (Sep 2020)
12. Mukhi, S., Dhiravani, K., Micholson, B., Yan, L., Hatchard, J., Mubareka, S., Bergeron, C., Beattie, T.: An innovative mobile data collection technology for public health in a field setting (Sep 2018)
13. e ISPUP ..., V.D.I.T.: AIPD_STAYAWAY_v2.0_09_2020.pdf (Aug 2020)
14. 60% já apagaram a StayAway covid: são 1,8 milhões de portugueses — saúde — PÚBLICO. https://www.publico.pt/2021/01/15/tecnologia/noticia/60-ja-apagaram-stayaway-covid-sao-18-milhoes-portugueses-1946366 (Jan 2021), accessed: 2021-10-5
15. Proposta de lei n.º 62/xiv (Oct 2020)
16. Shead, S.: Apple accused of breaching european privacy law by french start-up group. https://www.cnbc.com/2021/03/09/apple-accused-of-breaching-eu-privacy-law-by-french-start-up-group.html, accessed: 2021-10-9
17. Satariano, A.: Google is fined $57 million under europe's data privacy law. The New York Times (Jan 2019)
18. Glasgow-University: University news. https://www.gla.ac.uk/news/headline_-752925_en.html (Sep 2020), accessed: 2021-9-26
19. Xu, H., Zhang, L., Onireti, O., Fang, Y., Buchanan, W.B., Imran, M.A.: BeepTrace: Blockchain-enabled privacy-preserving contact tracing for COVID-19 pandemic and beyond (May 2020)
20. Yuen-C, T.: More than 4.2m people using TraceTogether, token distribution to resume soon: Lawrence wong, politics news & top stories - the straits times. https://www.straitstimes.com/singapore/politics/parliament-more-than-42m-people-using-tracetogether-token-distribution-to-resume (Jan 2021), accessed: 2021-10-5
21. Das, V.K.: Role of directed acyclic graphs in the blockchain landscape. https://www.blockchain-council.org/blockchain/role-of-directed-acyclic-graphs-in-the-blockchain-landscape/ (Sep 2020), accessed: 2021-9-27
22. Yang, W., Dai, X., Xiao, J., Jin, H.: LDV: A lightweight DAG-Based blockchain for vehicular social networks. IEEE Trans. Veh. Technol. 69(6), 5749–5759 (Jun 2020)
23. Nehra, V., Sharma, A.K., Tripathi, R.K.: Blockchain Implementation for Internet of Things Applications, pp. 113–132. unknown (Jan 2020)
24. About sawtooth events — sawtooth latest documentation. https://sawtooth.hyperledger.org/docs/core/nightly/1-2/app_developers_-guide/about_events.html, accessed: 2021-9-27
25. spec: AltBeacon technical specification, accessed: 2021-8-20
26. CDC: Contact tracing for COVID-19. https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/contact-tracing-plan/contact-tracing.html (Aug 2021), accessed: 2021-9-27

DECLARATION

I declare, under commitment of honor, that the work presented in this project, with the title *"Immunity Passport Ledger"*, is original and was carried out by Marco Verissimo Oliveira (2192406) under the guidance of Professor Catarina I. Reis, PhD (catarina.reis@ipleiria.pt) and Professor Marisa Maximiano, PhD (marisa.maximiano@ipleiria.pt).

Leiria, November 2021

Marco Verissimo Oliveira