information

MDPI

Article

Design of an Architecture Contributing to the Protection and Privacy of the Data Associated with the Electronic Health Record

Edwar Andrés Pineda Rincón *,† **and Luis Gabriel Moreno-Sandoval** †

Faculty of Engineering, Design and Innovation, Institución Universitaria Politécnico Grancolombiano, Bogotá 110231, Colombia; lgmoreno@poligran.edu.co

* Correspondence: eapinedar@poligran.edu.co

† These authors contributed equally to this work.

Abstract: The Electronic Health Record (EHR) has brought numerous challenges since its inception that have prevented a unified implementation from being carried out in Colombia. Within these challenges, we find a lack of security, auditability, and interoperability. Moreover, there is no general vision of the patient's history throughout its life since different systems store the information separately. This lack of unified history leads to multiple risks for patients' lives and the leakage of private data because each system has different mechanisms to safeguard and protect the information, and in several cases, these mechanisms do not exist. Many researchers tried to build multiple information systems attempting to solve this problem. However, these systems do not have a formal and rigorous architectural design to analyze and obtain health needs through architectural drivers to construct robust systems to solve these problems. This article describes the process of designing a software architecture that provides security to the information that makes up the Electronic Health Record in Colombia (EHR). Once we obtained the architectural drivers, we proposed Blockchain mainly due to its immutable distributed ledger, consensus algorithms, and smart contracts that securely transport this sensitive information. With this design decision, we carried out the construction of structures and necessary architectural documentation. We also develop a Proof of Concept (POC) using Hyperledger Fabric according to the literature analysis review in order to build a primary health network, in addition to a Smart Contract (Chaincode) using the Go programming language to perform a performance evaluation and do a safety analysis that demonstrates that the proposed design is reliable. The proposed design allows us to conclude that it is possible to build a secure architecture that protects patient health data privacy, facilitating the EHR's construction in Colombia.

Keywords: software architecture; blockchain; electronic health records; hyperledger fabric; security; attribute driven design; architecturally significant requirements; quality attribute; distributed ledger technology

Citation: Pineda Rincon, E.A.; Moreno-Sandoval, L.G. Design of an Architecture Contributing to the Protection and Privacy of the Data Associated with the Electronic Health Record. *Information* **2021**, *12*, 313. https://doi.org/10.3390/info12080313

Academic Editor: Corinna Schmitt and Wade Trappe

Received: 1 April 2021
Accepted: 8 June 2021
Published: 2 August 2021

1. Introduction

The process of designing an information system for the exchange of EHR offers a highly complex domain since it involves multiple actors and processes that must share sensitive information with patients' health data in the Colombian country. Each actor's technological health infrastructures must incorporate the information accumulated, facilitate its exchange, and be compatible with new medical services. All of this has made the medical sector more attractive to attackers and at a higher risk of data theft [1]. Furthermore, as mentioned in [2], healthcare data are notably vulnerable to hacking, and healthcare electronically increases risks. The health sector presented 41.4 million records violated with patient information in 2019, driven by a 49% increase in software piracy.

Medical information is protected by a summary of confidentiality, one of the highest levels of confidentiality. Therefore, security is an essential concept in preserving the privacy,

consistency, and integrity of clinical data since, for example, an unauthorized alteration of a patient's clinical data could put their life at risk. With the approach of a safe and robust architecture, according to the needs found, it will be possible to build multiple digital innovation systems in health and one Single EHR with a security backup, such as telemedicine, electronic prescription, among others.

This research aims to design a secure software architecture that establishes the structure, components, and relationships necessary to exchange information from Electronic Medical Records (EMR) in the Colombian health system. For this purpose, we use Attribute Driven Design (ADD) version 3.0, an architectural design method that offers a guide to satisfy the architectural drivers described above.

According to the review of the architectural design process inputs and needs collected through meetings with hospitals and experts in the health field located in Bogotá city, we choose Blockchain as the reference architecture to satisfy the critical security attributes identified in raising functional requirements and other architectural drivers found. A Blockchain architecture has very robust decentralization and security characteristics. Additionally, its encryption mechanism can be designed to verify the data's content to ensure that it has not been altered [3]. The authors in [4] indicated that the Blockchain architecture is more reliable than other susceptible information exchange mechanisms regarding the exchange of health information.

Subsequently, we carried out a systematic literature review to select the most appropriate Blockchain framework for the design. This review shows that Hyperledger Fabric is the framework that best adapts to the described system's needs. Therefore finally, we build a POC using this framework to consolidate the architectural decision.

Therefore, the contributions of our work are summarized as follows:

- Constructing an architectural design process that solves the problems proposed by creating the EHR in Colombia by obtaining and analyzing the architectural drivers to formalize the essential requirements.
- Proposal of Blockchain's use to solve the most crucial quality attributes (QAs) identified through an in-depth systematic literature review.
- Detailed comparison process of different platforms and reference frameworks to select the most appropriate technology stack to instantiate or implement the proposed architectural model, allowing EHR's exchange in the Colombian context.
- We developed a Poc that demonstrates the resolution of one of the essential attributes of the proposed architecture: Security, by evaluating its robustness against some common attacks within this type of network.

This paper's remainder is organized as follows: The background and related works are introduced in Sections 2 and 3. The design detail of the proposed architecture is described in Section 4. Section 5 introduces the proposed system. Section 6 analyzes proposed architecture through a POC. Section 7 describes the security and evaluation of the system. Finally, Section 8 gives the conclusion of this paper.

2. Background and Related Work

2.1. Electronic Medical Record in Colombia

According to article 34 of Colombian Law 23 of 1981, the medical record is the mandatory record of the patient's health conditions. It is a private document that must be reserved, and third parties can only see it with prior authorization from the patient; each patient's data are an integral part of the national medical history [5].

The Resolution 1995 of 1999 of Colombia defines the Electronic Medical Record as the file made up of the set of documents in which the mandatory registration of the state of health, medical acts, and other procedures performed by the health team involved in the care of a patient is carried out. It also classifies it as a document of vital importance for scientific development. Finally, the EHR is defined as that document to which the Clinical Histories are transferred and have a significant scientific, historical and cultural value that must be permanently preserved [5].

Following Law 1581 of 2012 (Habeas Data Law), only persons authorized by the holder can perform treatment. Therefore, such information will only circulate on the Internet or other mass media if the technical, human, and administrative measures that establish security and access are in place [5].

The National Government issued Law 2015 of 2020, aiming to regulate Electronic Medical History's interoperability to share relevant health data by all Colombians, safeguarding and respecting Habeas Data. The Electronic Medical Record must contain the relevant clinical data of the person in a transparent, complete, and standardized manner with the highest levels of confidentiality [6].

2.2. Software Architecture

Software architecture from a technical point of view is the set of structures necessary to understand a system, which includes software elements, relationships between them, and their properties [7].

The purpose of the Software Architecture is to create documentation (Software Architecture Document) with the analysis made, the QAs described, the technological and business restrictions, the design carried out, and the description of motivators which justify decision-making and that are made in early stages because they will be the pillars for the development and implementation of the information system.

2.2.1. Architecturally Significant Requirements

An Architecturally Significant Requirement (ASR) is a requirement that will have an enormous effect on the architecture, that is, the presence or absence of which radically affects the application's architecture. It includes the most critical functionality of the application, the business and technology system's constraints, and the Quality Attributes . Another name that ASRs are known by is Architectural Drivers [8].

The architectural drivers are in charge of molding an architecture design. These elements generate a need to create a solution to a problem. It is composed of a design purpose, the QAs, the main functionality, architectural concerns, and the constraints or Constraints of the system [8].

2.2.2. Quality Attribute (QA)

A QA is a measurable or testable property of a system that is used to indicate how well a system meets the needs of its stakeholders [7].

2.2.3. Quality Attribute Scenarios

To document the QAs, Quality Scenarios are used, which describe the response that a system should have to a specific stimulus, always thinking in numerical values with an emphasis on the limit cases, with this numerical value, we establish and document the desired response to the received stimulus, which is also known as **response measure**. There are other parts necessary to complete a scenario and therefore make it fully descriptive, the first is the source of the stimulus, the affected artifact, and the conditions or the environment under which the scenario is executed (Environment) [8].

2.2.4. Attribute Driven Design—ADD

ADD is a well-established and mature software architecture design method. It establishes a step-by-step guide in order to perform software architecture design iteratively. Moreover, it was the first method to focus specifically on the system QAs and their achievement by creating architectural structures and their representation through views [8].

2.3. Blockchain

Blockchain or Distributed Ledger Technology (DLT) [9], is a technology that was initially proposed in a cryptocurrency system known as Bitcoin, which allows a P2P (Peer to Peer) transfer of digital assets without intermediaries or third parties [10]. It is

a decentralized ledger of immutable transactions where each node maintains a copy of the ledger by applying transactions that a consensus protocol has validated, and these transactions are grouped into blocks that include a hash that links each block to the previous one, as shown in Figure 1.

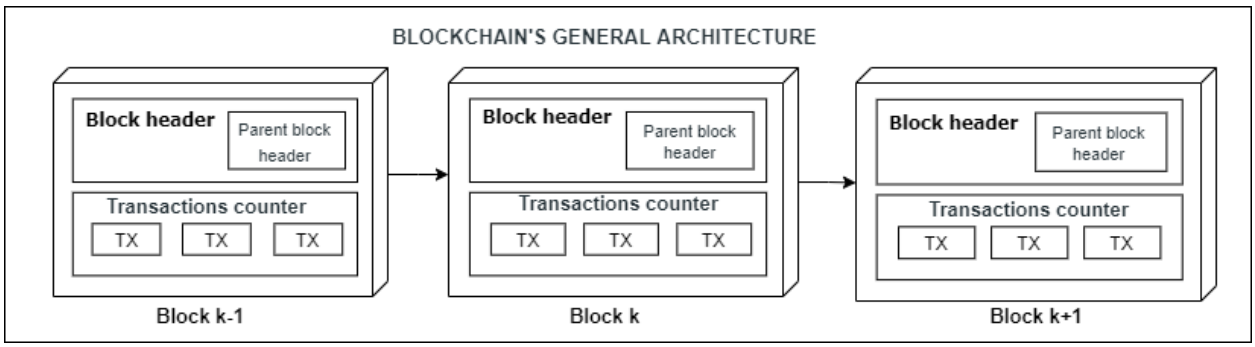

Figure 1. General blockchain architecture.

2.4. Blockchain Structure

A transaction is the fundamental element of Blockchain, a set of transactions is validated and issued. Many transactions form a block. Many blocks form a chain through a digital data link. Each of the blocks goes through a consensus process to select the following block added to the chain. The chosen block is verified and added to the current chain.

The consensus and validation processes are carried out by particular peer nodes called miners. The general functioning of the Blockchain architecture is shown in Figure 2.

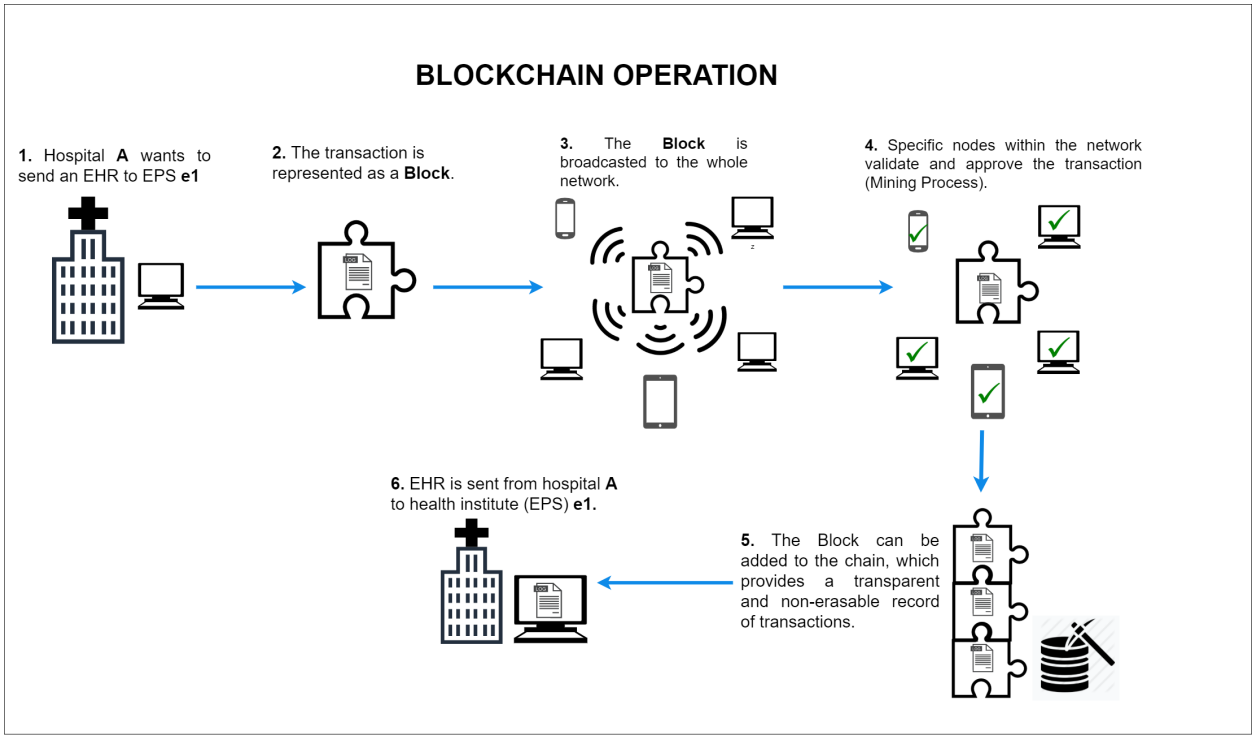

Figure 2. Blockchain operation.

2.5. Public vs. Private Blockchain

In Table 1 we find a general overview of the comparison between Private and Public Blockchain types, according to the use cases implemented in healthcare.

Table 1. Private vs. Public Blockchain Features.

Public	Private
Pseudo-anonymous nodes	Identified nodes
Permissions to view all data	Restricted data.
Focused on Cryptocurrencies such as finance, supply chain, etc.	Focused on enterprise solutions in various industries
Miners are required	Miners are not needed.
Use of PoW as consensus protocol in most cases	Use of different consensus protocols.
High power consumption	Low power consumption.

In Table 2 we find a comparison associated with the QAs that according to the literature review, may be needed in the Healthcare area.

Table 2. Private vs. Public Blockchain Quality Attributes.

Attribute	Public	Private
Performance	Low	High
Latency	Slow	Medium.
Number of readers	High	High
Number of writers	High	Medium
Number of untrusted writers	High	Low.
Consensus Mechanism	Mainly PoW, and PoS	Many, BFT protocols.
Centrally managed	No	Consortium (Set of participants).

2.6. Hyperledger Fabric

Hyperledger Fabric's objective is to provide a decentralized platform that creates specific functionalities for particular business rules using Smart Contracts called Chaincode within its architecture. It was also designed to have privacy as one of its main QAs [11].

One of Fabric's best qualities is that it has a pluggable modular architecture that allows it to be used with different consensus algorithms such as PBFT, SIEVE, or NOOBS. Furthermore, we can program Chaincode in some of the most widely used general-purpose languages in the industry, such as Java, Node Js, and Go. This characteristic gives Fabric a significant advantage over other architectures that must be programmed on Domain-Specific Restricted Languages, for its acronym in English (DSL) [12] which allows Fabric to achieve a lower learning curve of the language and more outstanding documentation at the time of implementation, key aspects to take into account when choosing a Software Architecture [8].

Finally, it is worth mentioning that a crucial capability of this framework that makes it different from other DLTs is to maintain multiple ledgers within its ecosystem, i.e., within its architecture, there is a single ledger for each domain [11].

2.6.1. Hyperledger Fabric Architecture

Fabric employs a layered architecture that helps fulfill the privacy of the ledger. The first layer is the identity layer, where a trusted service provider called Membership Service Provider (MSP) manages the identities of all nodes participating in the ledger (enrollment process). This layer allows the creation of security policies that specify which entities can execute which actions.

The second layer is the channel layer, where Fabric allows the creation of Fabric's channels for specific sets of entities and segregates their relationships from other entities. Fabric allows associating a Ledger to a specific channel and exercising adequate privacy controls over it, isolating transactions of entities that do not have to know them (different domains).

The core of its architecture is the Bockchain Service which is composed of [9]:

- Consensus manager, responsible for providing the interface to the consensus protocol and receiving transactions and executing them according to the chosen consensus algorithm.
- Distributed ledger, a database used by Smart Contracts to store relevant state information during transaction execution.
- A peer-to-peer protocol.
- Ledger storage

2.6.2. Consensus in Hyperledger Fabric

The consensus in Fabric is divided into three phases: Endorsement (Endorsement), Ordering (Ordering), and Validation (Validation).

- Endorsement: It is oriented by a policy where only certain participants take the role of endorsing a transaction.
- Ordering: The ordering phase is in charge of accepting transactions that were previously accepted in the endorsement step. Additionally, an order is agreed upon to confirm transactions in the distributed ledger and delivers the resulting blocks to the committing peers.
- Validation: Obtains a block of ordered transactions and validates the certainty of the results, including verification of the approval policy and ensuring that there is no double-spending.

2.6.3. Nodes in the Fabric Network and Their Roles

- Clients: They are applications in charge of executing transactions in the system by sending them to the endorsing peers or computers.
- Committing peers: This phase is in charge of persisting the chains transmitted in the channels (committing them) in addition to maintaining the general ledger and its status. May have a smart contract (Chaincode).
- Endorsing peers: This phase is responsible for collecting transactions (transaction proposal) from customers and analyzing them using Smart Contracts to enforce their associated rules (endorsing them), e.g., verifying whether an entity is allowed to execute a specific action on the ledger and then signing them.
- Peers or Computing Services: This phase is responsible for approving the inclusion of transaction blocks in the ledger and communicating with peer nodes that commit and endorse to ensure that the ledger is consistent.
- Authentication Authorities: Responsible for ensuring the veracity or authenticity of the identity of users or components, in addition to authenticating users so that they can execute transactions.

These components communicate with each other through *Channels*, which are structures created to allow transactions privately.

2.6.4. Types of Ordering Services

- Solo: Type of service suitable for testing as it does not have fail-safe functionalities such as that of Byzantine Generals failures. It is recommended for moderately reliable environments.
- Kafka-based: Ordering service recommended for productive releases in Fabric versions before 1.4.2 uses Apache Kafka components.
- RAFT Emerged from version 1.4.2. It has Byzantine Fault Tolerance (PBFT). It is recommended for unreliable environments.

2.7. Related Works

2.7.1. Benefits of Blockchain in Healthcare Software Architecture

The author of [4] demonstrates that, concerning data exchange in healthcare and to preserve the integrity and confidentiality of such data, Blockchain Architecture is the

most reliable Architecture to exchange this kind of information with a degree of reserve summary. Ref. [13] highlights that Medicine is one of the most important and promising areas for Blockchain adoption; it also mentions four main categories in which Blockchain has been adopted in healthcare.

- Improved management in medical records, within which are solutions for secure storage of patient medical records, secure and scalable clinical data exchange, privacy risk management assistance, and EHR's security or privacy.
- Insurance claims process improvement applications
- Applications that accelerate biomedical research
- Applications that aid in healthcare through Ledgers

The decentralized management of Blockchain, in addition to robustness, availability, scalability, improved security and privacy of some of its components such as immutable records, are clear benefits of Blockchain, where health research institutions and care providers prefer not to cede control to a single central authority, but instead want to work together [13].

Blockchain has a strong fit to provide a suitable solution for addressing problems arising from sharing of medical data through its immutability and decentralization features [14].

2.7.2. Best Suited Blockchain Type in Healthcare

As mentioned in [13], to start designing and building a Blockchain project in healthcare, one of the critical steps is to select the most suitable underlying Blockchain platform. Comparative analysis at a qualitative and quantitative level of some of the blockchain frameworks, both public and private, was performed in [11]. They mentioned that it is vital to understand the underlying architecture and characteristics of a specific Blockchain, i.e., whether its category is public or private, or whether its architecture is general-purpose or designed for specific business rules (Finance, Banking, IoT), in order to determine which platform to use.

2.7.3. Public or Private Blockchain?

The authors of [3] highlighted that health data should be built in a secure environment, such as a private blockchain instead of a public one, in order to increase the efficiency and stability of the transmission and exchange of this type of data.

The authors of [15] mentioned that the private blockchain allows access only to invited and verified users, which ensures greater nodes accountability since, in the first place, all nodes are already known users, and besides, Blockchain keeps a record of their actions. Regarding data immutability, the authors of [11] mentioned that private blockchain networks cannot provide the same amount of security compared to public ones, but however, a good security level is achieved since the validator nodes are trusted participants in the network.

The authors of [12] mentioned that many business use cases require performance features that public blockchain networks are unable to provide. In a public blockchain, the state of the ledger, the transactions and the stored information is transparent and accessible to all, leading to privacy issues for particular scenarios where the data requires it to be preserved. Additionally, private blockchain neteorks contain features that make them suitable for scenarios that need to deal with highly sensitive data [11].

The authors of [16] decided to implement a Permissioned or authorized instead of a Permissionless: public Blockchain system mainly because:

- In a public blockchain, users' anonymity and inability to verify account owners' identity could cause data misuse, whereas in a private blockchain, a user or entity must be authenticated or authorized before joining the system.
- Privacy of patient data.
- Fast system response (Throughput).

The authors of [17] mentioned that Blockchain lacks scalability given a large amount of information that must be transmitted and stored in the system. If each patient visit is considered a transaction on the Blockchain, the amount of space and the architecture to support it must be considered to apply Blockchain in this domain successfully.

This fact indicates that between a public and private Blockchain, the choice of a private blockchain would exponentially improve scalability and storage as only specific nodes would be in charge of handling the consensus and addition of blocks to the chain.

Table 3 shows a summary that can be used as a reference to choice a blockchain technology.

Table 3. Guide for selecting a healthcare-focused Blockchain platform.

Element	Options and Features
Permission type in the network	Public (Permissionless): Public participation is required, e.g.,: Applications that rely on patient-managed data. Private (Permissioned): It is required to include only authorized participants, e.g.,: Information exchange networks between hospitals.
Consensus Protocol or Algorithm	Proof of Stake: Suitable for healthcare applications. Kafka Based on election process. Finalizes consensus quickly due to a small number of members.
Programming language	Bitcoin Script: Smart contracts with simple programming languages. Not useful for healthcare applications, . Solidity: Ethereum smart contracts language. It is one of the most popular languages for writing Smart Contracts. Go, Node.js and Java: Hyperledger Fabric Chaincode languages. Widely known and used programming languages. An interesting paper to choose Hyperledger's Chaincode language can be found in [18].
Software License Type	MIT: Open source, and non-copyleft license. Allows healthcare applications to reuse the source code of the Blockchain platform. GPL: Copyleft license that allows source code use. Its derivative works in some cases must also be open-source. Apache License v2.0: Type of license used by Fabric, it is not a copyleft license but requires a copyright notice retention.

2.7.4. Best Suited Blockchain Architectures for Clinical Informatics

There are many Blockchain platform options available today. According to [13] there are just over 35 Blockchain platforms within the industry that can be used in healthcare. A Blockchain platform should be general-purpose (not limited to financial applications), as well as being technically mature and widely supported by the community for easing the construction and future maintenance of the system [13].

It is essential to perform a detailed analysis on the Blockchain architecture or technology to be used since each one was designed for a specific purpose. Additionally, in some cases, we find that more than one Blockchain platform can help an existing problem, so it is necessary to make a trade-off taking into account the most critical QAs and find the one that satisfies them the most.

The authors of [13] showed that Ethereum, Hyperledger, and MultiChain are the most likely platforms to create applications in the healthcare area because they combine the most relevant features for these kind of systems. Considering this statement and according to the literature review performed, some of the blockchain architectures such as the Hyperledger suite (Fabric, Sawtooth), Monero, Multichain, and Ethereum are within the most indicated architectures according to the guidelines oriented to the healthcare domain.

Hyperledger provides a set of quite robust architectures that can be used to build a Secure Architecture in EHRs. Hyperledger Burrow has some exciting features but does not yet have complete documentation and is still in the process of migration to the Hyperledger project [11]. According to the literature review, Hyperledger Fabric poses to be one of

the most suitable architectures as it has strong support in privacy using a multi-channel architecture. Additionally, it provides flexible support for writing complex Smart Contracts using general-purpose programming languages, highly adopted in the industry such as Java, Go, and Node.js [12].

Hyperledger Sawtooth could provide higher security than Fabric due to its SGX component and its secure execution environment called enclave [19]. However, it relies on the SGX component which constraints to integrate the architectural solution into legacy systems of healthcare entities that do not possess this technology [11]. Monero has some exciting security features that could be applied in multiple Health Record exchange processes. However it has some limitations due to its high power consumption since it only uses the PoW consensus protocol [13]. On the other hand, Multichain has been used in multiple healthcare use cases, such as connected health [11], but it does not provide extra privacy to the users like other platforms such as Fabric and Ethreum.

We proceed, therefore, to make a comparison between Ethereum and Hyperledger Fabric's architectures, mainly because of their consolidated Distributed Ledger architectures [11], and additionally because some architectural designs and implementations have been made in the field of EHRs and Health using both frameworks. Below, we describe some of these implementations and the QAs taken into account to shape such architectural constructions.

Table 4 details some papers where Hyperledger Fabric technology was chosen as the Blockchain Framework to implement a healthcare security system and the respective design rationale for its choice.

Table 4. Blockchain in healthcare systems using Hyperledger Fabric.

Quality Attributes	Design Rationale
Throughput, Reliability	Based on healthcare context architectural constraints. The need to pay for transaction execution (As in Ethereum) limits the system's usability and throughput [16]. Hyperledger Fabric aims to develop distributed ledgers with a particular focus on improving the performance and reliability of such systems [9].
Scalability	Hyperledger Fabric would exponentially improve scalability and storage size in healthcare domain as only specific nodes would be in charge of handling the consensus and addition of blocks to the chain [17]. Having a ledger for each channel allows distributing transactions among several committer nodes, increasing the amount of data and the requests that a node can store and fulfill [17]. Fabric solves some performance, scalability, and privacy issues that some permissionless blockchain architectures such as Ethereum possess thanks to its permissioned blockchain mode of operation, use of BFT consensus algorithm and fine-grained access control [20].
Security: Privacy, Confidentiality, and Reliability	Hyperledger Fabric has a multi-channel architecture where participants cannot access a chain if they do not have access to the channel the chain belongs to [9,17]. Additionally, it is reliable since it will not allow the information to be modified. Fabric has a security infrastructure that includes transaction enrollment and authorization through a public key certificate [9] and achieves a level of confidentiality through in-band encryption (occurs while data are in transit) as described in [21]. In [22] a Hyperledger Fabric-based solution is proposed which stores patient records in an immutable distributed ledger with anonymity and data privacy. Additionally, the use of Idemix cryptographic protocol suite provides patient's data unlinkability.
Auditability	Regulatory compliance and access for regulators to investigate transaction records is required in health context [22]. Fabric's ledger provides this QA, as it provides authorized entities with the means to link user transactions according to their roles and access a particular user's activity in the system [23]. The authors of [22] created a system that supports secure auditing, and preservation of privacy using Hyperledger Fabric.

Table 4. *Cont.*

Quality Attributes	Design Rationale
Modularity	Authors of [12] mentioned that one of the differentiating qualities of Hyperledger Fabric over other DLTs is that it has a highly modular and configurable architecture, which allows it to be used in a wide range of particular use cases, such as EHRs. This modularity is achieved through different types of Pluggable Consensus Protocols, which allows it to be adjusted to particular cases and specific trust models.
Flexibility (Modifiability), Resilience	The modular architecture of Hyperledger Fabric delivers a high degree of resilience, flexibility, and confidentiality. This flexibility enables other QAs such as Scalability and Privacy to be achieved [9].

Some projects that built a private Blockchain architecture based on Ethereum to create healthcare data management systems and their respective design rationale can be seen in Table 5.

Table 5. Healthcare-oriented Blockchain systems using Ethereum.

Quality Attributes	Design Rationale
Confidentiality, Availability	An Ethereum Blockchain Architecture for clinical records exchange platform integrated with the international standard for health data exchange HL7—FHIR (High Level Seven—Fast Healthcare Interoperability Resource) was built in [3], it manages clinical data exchange authorization to create a private chain.
Integrity	Ethereum private chain architecture was used to store the Hash values of patients' health records (PHRs) in the Southeast Asian health network to ensure their integrity in [3].
Fault Tolerance	The core of Ethereum is its EVM (Ethereum Virtual Machine), which maintains consensus across the entire Blockchain network when executed on each network node. This decentralization of consensus ratifies an extreme level of fault tolerance [9].
Performance	Ethereum Architecture has been used together with the PoA consensus mechanism, which compared to other proof mechanisms, allows faster creation of the blocks since the verification of the new blocks will be done by some verifier nodes using their real identities [3] , so the waiting time for information exchange is reduced considerably.
Security, Privacy	Ethereum achieves Immutability and Security in part through SHA-256, the default hash algorithm used by the Ethereum architecture to create blocks composed of the block's content, the hash value of the previous block, and a timestamp. The authors of [14] proposed a data sharing framework to ensure access to sensitive electronic health data records using a blockchain-based data-sharing scheme and the use of smart contracts built upon Solidity to monitor the behavior of data when it is outside the custodians care facilities.
Auditability	Authors of [14] implemented a blockchain-based system to provide data provenance, auditing and traceability for medical data among health entities (cloud service providers) mainly through tagging the smart contracts to the data.

In Tables 4 and 5, we can find exciting design justifications detailing QAs that show why researchers and Architectures instantiated a Software Architecture using Hyperledger Fabric or Ethereum Private Chain. We show a comparative analysis of these two Blockchain Architectures below.

Regarding Smart Contracts, Ethereum lacks update functions, making it difficult to update when adding new features or fixing an existing bug. In [9] there is a description of some types of attacks to which Ethereum has been vulnerable, such as the 51% Attack [24], which occurs when an attacker possesses more than half of the mining power of the network since they can process blocks faster than the rest, allowing them to create their

chains. Furthermore, in Ethereum, Smart Contracts are visible to all Blockchain users, which increases the likelihood of a user seeing existing security holes or bugs and is not suitable for handling sensitive data [11].

The authors of [25] described another famous attack suffered by Ethereum in 2016: the famous DAO attack, in which hackers stole millions of USD in cryptocurrencies a blockchain hard fork was necessary. This branch nullified the effects of the transactions involved in the attack. Some DLTs such as Fabric and Sawtooth support the ability to update Smart Contracts, a vital feature when a bug or error needs to be fixed, or a new feature needs to be incorporated.

There is a significant difference between Hyperledger Fabric and Ethereum: the way they were designed, and their target audience. For example, EVM runs smart contracts publicly to any user without permission, ideal for distributed or DAPPS applications. On the other hand, Fabric has a modular architecture that ensures flexibility and scalability in a permission mode.

The authors of [20] made an excellent comparison between some Blockchain frameworks, such as Ethereum and Hyperledger Fabric. They also present a helpful guide of the use cases where each one excels. First, they mention that Hyperledger Fabric has a modular and extensible architecture used in multiple industries such as Banking, Healthcare, and supply chains. Ethereum, on the other hand, provides a generic platform for all types of transactions and applications but has less rigor than Fabric for having modularity as the main QA when building its architecture.

Related to the security of data storage in the ledger in Ethereum, ref. [20] highlights that, although the logs are anonymized, these are accessible to all participants, which is a severe problem to take into account for applications that require a high degree of privacy. Additionally, ref. [11] highlights that in this type of Blockchain, all nodes are identified through cryptographic pseudonyms, which hinders the Auditability of the system (QA associated with Security) and are open to Sybil types of attacks in which a malicious node impersonates several different nodes, called Sybil nodes, simultaneously in an attempt to disrupt the proper functioning of the network [26].

Due to the authoritative nature of Fabric, fine-grained access control to records is provided, improving privacy. Moreover, better performance is achieved since only the Peers that are part of a transaction must achieve consensus [20].

Another essential point to consider when selecting a Blockchain Architecture is its consensus protocols, even more so in systems that require a high degree of security. According to [11], Hyperledger Fabric has multiple consensus protocols, unlike Ethereum, which is based on PoW as a consensus, and on an implementation of a Hard Fork performed to its main branch in which PoS is wanted to be implemented as a new protocol.

2.8. Our Proposal's Features

Our system is based on a rigorous architectural design that solves the problems identified and has prevented creating an EHR in Colombia. Additionally, we made each decision within each design iteration based on the state of the art that supported it. Through the POC carried out, we demonstrate how sensitive information such as clinical data can be safely exchanged using blockchain characteristics, more specifically the permissioned blockchain network, to adapt to the context of the Colombian health system. We did not find any previous papers that implemented a design solution using version 2.2.0 of Hyperledger Fabric, which delivers essential private data enhancements such as sharing and verifying private data, Collection-level endorsement policies, and delivers enhanced governance around smart contracts [27].

Additionally, we used a benchmarking tool oriented to Blockchain systems known as Hyperledger Caliper v0.4.1, which allowed us to tune the network created in initial iterations and optimize it to satisfy the requests found in the Colombian health system. In the articles reviewed, authors used load testing tools to analyze network performance, but which do not have configurations aimed at detecting errors in the blockchain network and

improving them; furthermore, some of the papers consulted did not even mention [28] the tool used for the performance evaluation of the system.

A comparison of some key characteristics in the handling of sensitive data with some proposals related to our system is shown in Table 6.

Table 6. Comparison with related works.

Paper	Network	Technology	Privacy	Integrity	Access Control	Latency/Throughput Testing
ChainFS [29]	Private	Ethereum	✓	X	X	✓ / X
Block DS [30]	Private	Agnostic	✓	X	✓	X / X
MedRec [31]	Public/Private	Ethereum	✓	✓	X	X / X
Tamper-resistant mHealth [32]	Private	Fabric 0.5	X	✓	X	✓ / X
PREHEALTH [22]	Private	Fabric 1.4	✓	✓	X	✓ / X
Our work	Private	Fabric 2.2.0	✓	✓	✓	✓ / ✓

3. System Architectural Design

We presented a diversity of current social problems related to health in the Colombian system and available technological tools that can be used and structured to carry out the solution described in this section.

3.1. Architectural Drivers

3.1.1. Quality Attributes

According to the literature review, we can conclude that the QAs have an enormous effect in establishing how a system will be designed. Therefore, they will be an essential input in the design of the architecture to be proposed.

3.1.2. Utility Tree

In Figure 3, the utility tree designed from the elicited drivers can be observed, a scale of L = Low, M = Medium, and H = High is used to classify each of the two dimensions, and besides, as we will see below the scenarios described in the next section are prioritized in order first to choose those with the following combination (H,H), (H,M) or (M,H) [8].

Once the QAs' prioritization process has been completed, the QAs that must be taken into account for the design and approach of the architecture are Security, data confidentiality, and interoperability.

3.1.3. Quality Attribute Scenarios

The QA scenarios were obtained by analyzing the requirements and cases obtained from the Colombian EHR law and experts related to the health field. A methodical selection process was performed with these architectural drivers, which describes the Utility Tree of Figure 3. Therefore we propose to choose the attributes described in Table 7 as leading modelers and governors of the architecture.

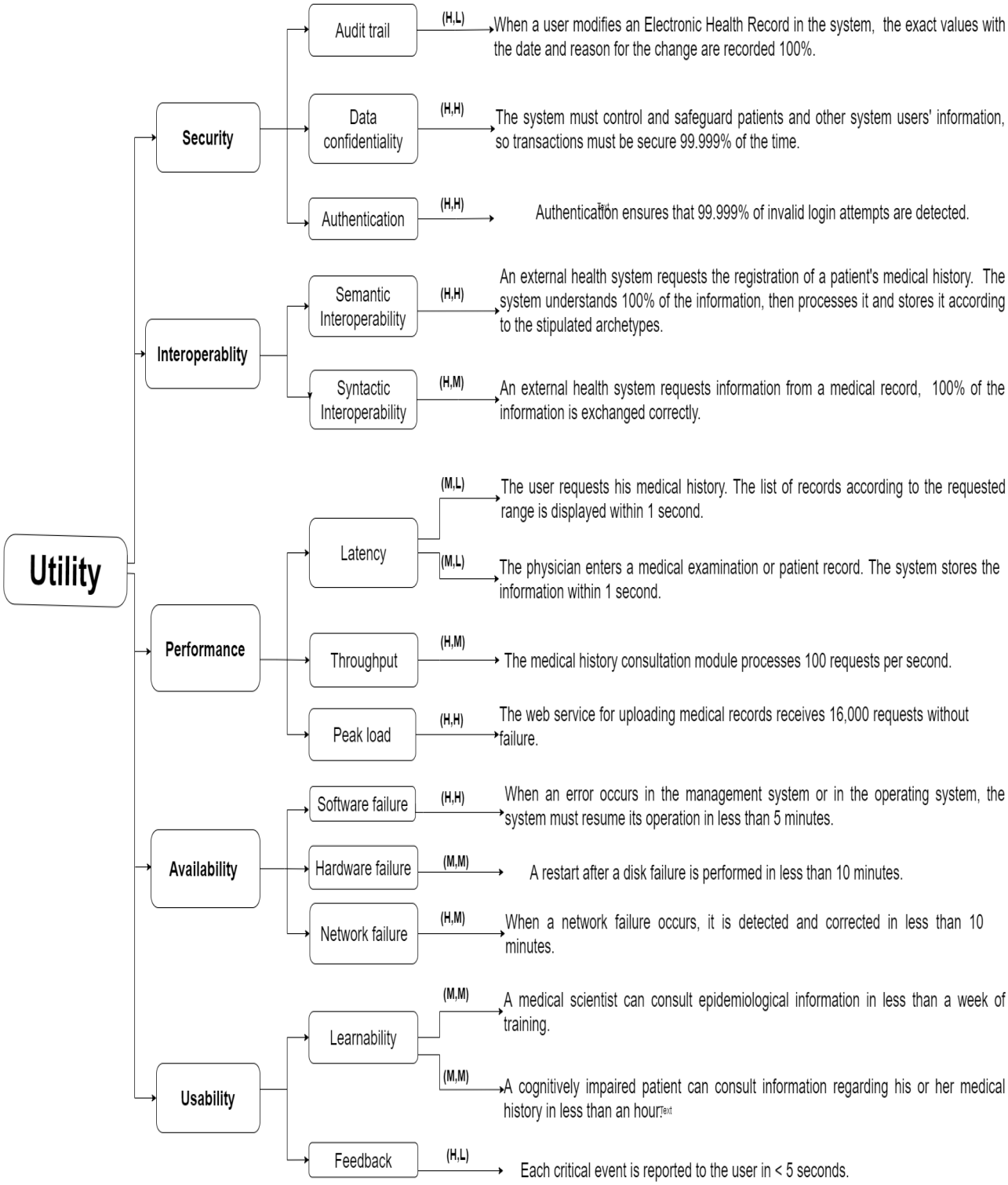

Figure 3. System Utility Tree.

Table 7. EHR Quality Attribute Scenarios.

Id	Quality Attribute	Scenario
QA-1	Security	When an electronic medical record is altered in the system, the precise values of the change are recorded 100% of the time.
QA-2	Security	When an intruder alters an electronic medical record in the system, the change's precise values are recorded 100% of the time.
QA-3	Security	When a user makes a change in the system, it is possible to know who performed a transaction in the system 100% of the time.
QA-4	Interoperability	A hospital with an external health system requests a Patient's Medical History registration in another system. 100% of the information is understood by the system, processed, and stored according to the archetypes stipulated.
QA-5	Availability	When an error occurs in the system, it must resume its operation in less than 1 min.
QA-6	Availability	When a network failure occurs in one of the EPS or health network entities, the system availability is affected less than 5%.

3.2. System Architecture Design Process

3.2.1. Step 1—Review of Inputs to the Design Process

In this step, we review the architectural drivers necessary for the design process, including the primary functional requirements and QA scenarios. Table 8 shows the Design purpose and primary functional requirements.

Table 8. Architectural design inputs.

Category	Description
Design purpose	We will design a greenfield system (a new system) that is part of a mature domain. The primary purpose is to build a detailed design in order to be able to build a system that contributes to the protection of the data associated with the EHR.
Primary Functional Requirements	Based on the context of the EHR and the primary functional requirements obtained from it, we determined that the main Use Cases are: • CU-7 Request Medical History • CUS-13 Information Exchange between health entities • CU-14 Download Patient's Medical Record • CU-8 Authorize Third Parties • CUS-14 Store accesses to the system

Table 9 illustrates the prioritization of QAs with the support of the utility tree depicted in Figure 3.

Table 9. QA Scenarios according to the utility tree.

Scenario ID	Importance for the Health Network	Difficulty of Implementation
QA-1	High	Low
QA-2	High	High
QA-3	High	High
QA-5	High	High
QA-6	High	Medium

3.2.2. Iteration 1: Establish an Overall System Structure

Once we described the inputs to our architectural process, we use the steps described by ADD in this first iteration of the design process to choose the design concepts that satisfy the selected motivators. In this first iteration, we select the appropriate design concepts in order to structure the complete system.

According to the literature review and the analysis of the previous sections' motivators, the first architectural decision made is to choose the Blockchain Architecture as the reference architecture to satisfy the identified critical security attributes. This style will be the one taken into account for the design and construction of the prototype solution. We select the Technology Stack, which is fundamental in a software architecture design process, to choose the most appropriate framework. We rely on the extensive systematic literature review made in Section 2.7.2 where we conclude that Hyperledger is the most appropriate Architecture for the proposed system.

3.2.3. Construction

A sequence diagram explaining the process of the proposed architecture is sketched in Figure 4.

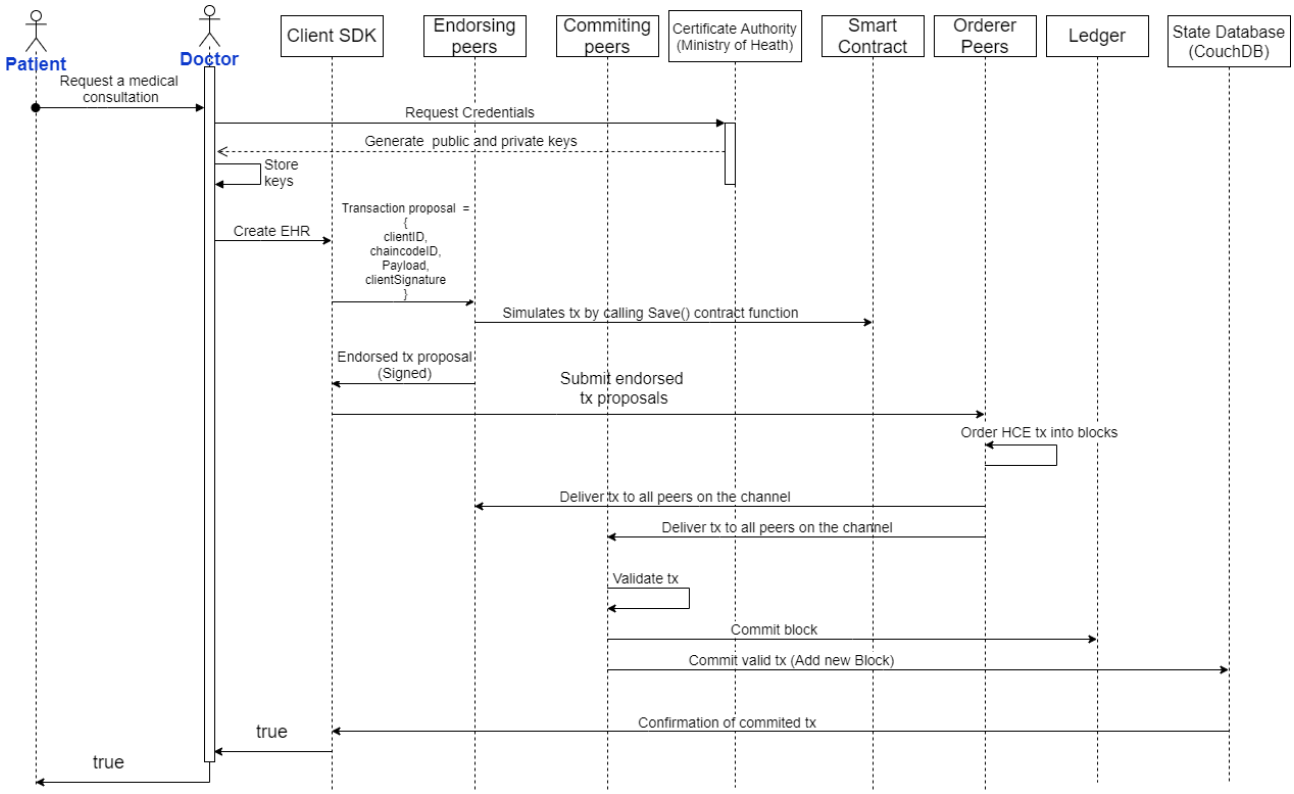

Figure 4. Sequence Diagram of the proposed system.

4. The Proposed System—Iteration 2

System Architecture

We then select Hyperledger Fabric v2.x as the reference architecture to solve the QAs related to security, scalability, and traceability of clinical data, Figure 5 represents a general description of the system according to the objective of our first iteration of structuring the complete system.

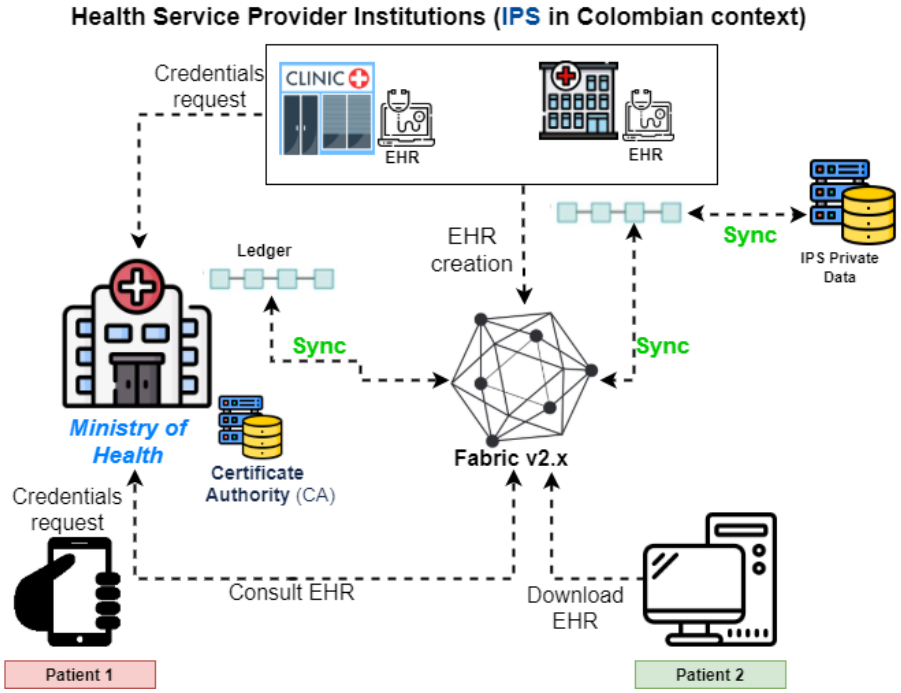

Figure 5. General system architecture.

Figure 5 also depicts the responsibility of each institution within the network, homologating their respective role within the Hyperledger Fabric network. There will be a central entity or organization in charge of managing the network's certification authority within the network. Therefore, it will be responsible for allowing access to the respective Health Care Provider Entities (EPS) and their respective nodes. In our design, this organization will be the Ministry of Health and Social Protection since it is the governmental entity that regulates the provision of health services in Colombia. Additionally, it will provide the ordering service within our blockchain network to maintain the order and integrity of the transactions within our Blockchain and ensure the consensus protocol's execution.

Within our architecture and according to the described QAs, we will proceed to build a first hyperledger channel over our blockchain network, which will be in charge of handling the flow process related to the EHR creation, modification by the healthcare actors that have the required permissions, and the patient's request for their EHR.

We set up a scenario in which there is a network supervising entity. This homologation within the Colombian system would be in charge of the Ministry of Health, and we also involved two organizations or hospitals within the Colombian health system, which would simulate the exchange of patients' medical records.

The second architectural decision to satisfy the highest priority QAs for the EHR, is to select Docker so that each practitioner in the network is inside a container. This container consists of a set of the image layer. Moreover, it is isolated from the host, which provides an extra layer of security since it is isolated to the host's errors or attacks. In this way, Smart Contracts will also be executed in an isolated and secure way, thus reinforcing the architecture's security capabilities. Finally, Docker allows creating the containers associated with the network participants in a simple way once the Docker Compose YAML is configured. Each hospital and health entity has a peer. In these network peers, the distributed ledgers were stored using CouchDB and a Chaincode in charge of simulating an EHR's storage and content.

The architecture comprises nodes that belong to the ordering service over the network that helps to order the requests or transactions sent by the different health entities. Each health entity is composed of a single node to demonstrate the core functionality of the architecture.

The Certificate Authority is an indispensable component within the proposed architecture, as it controls the enrollment of the health entities within the network and the different actors in charge of sending the blockchain asset (Medical History). In this architectural design. As mentioned in the Ministry of Health is the organization that owns this primary component, as we see in Figure 6 and will be in charge of granting access to the network.

Inside each participant, we observe that a Chaincode written in Golang is hosted, which will contain the following functions:

- Store medical history information in the ledger.
- Query the medical history information.

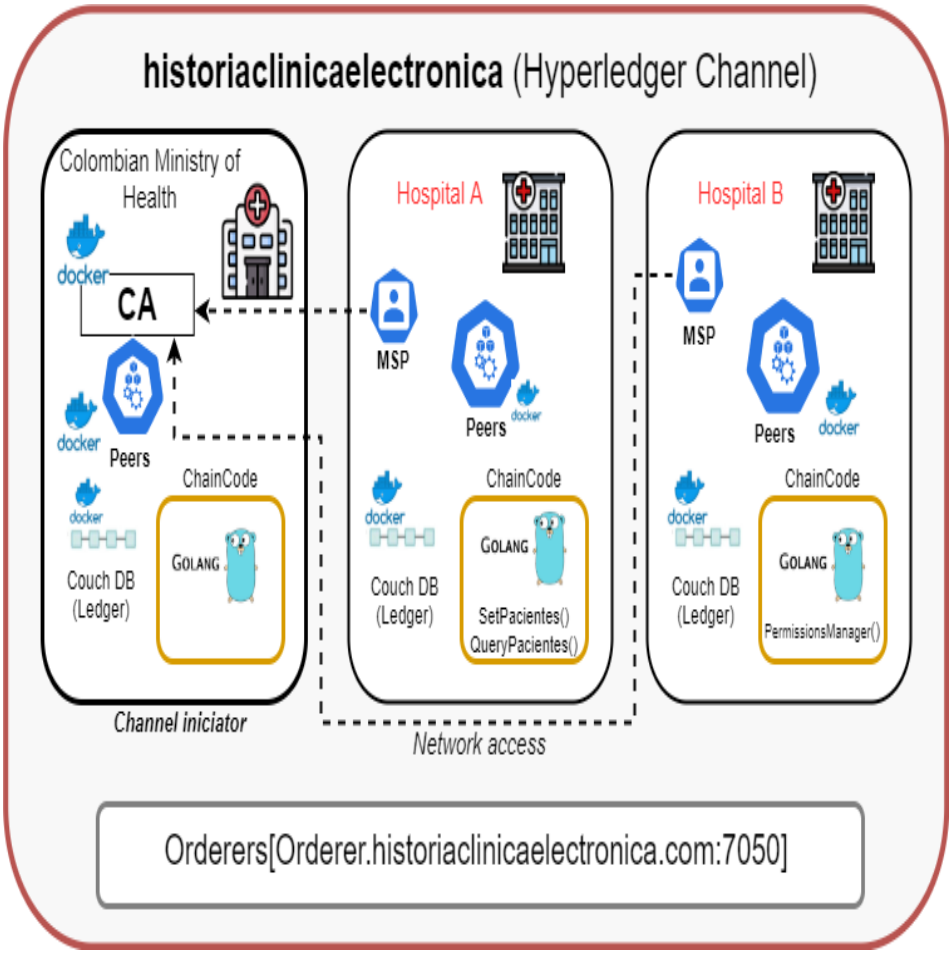

Figure 6. POC Implementation Overview.

5. Results

To analyze the proposed architecture, we proceed to perform a POC associated with the Blockchain network described in Figure 6. The implementation consists of a virtual machine with Ubuntu 20.0.4 LTS, CPU Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz 1.99GHz, and 8 GB of RAM virtualized using VMware WorkStation 15.5.

5.1. Hyperledger Installation

We used the scripts located in the official Hyperledger Fabric Latin America GitHub to install the Hyperledger Fabric requirements, which can be found at the following URL: https://raw.githubusercontent.com/blockchainempresarial/curso-hyperledger-fabric/master/scripts/prerreq.sh (accessed on 22 August 2020).

5.1.1. Creation of Cryptographic Material

The first step to build a Hyperledger Fabric network is to create the cryptographic material; here, we create the certificates and private keys for each of the components and the organizations' default users within the network. To perform the security tests to the architecture, we use the Cryptogen tool, a tool provided by Hyperledger that has the functionality to provide the configuration of the networks in Hyperledger to perform proofs of concept [33]. Its use is to create the certifying authority hosted within the Health Ministry node according to the architecture built, and that will be in charge of distributing digital certificates and granting access to the network.

We created a file called crypto-config.yml whose purpose will be to serve as a template or guide for Cryptogen for the creation of cryptographic material, which is a directory of encryption configuration that contains several certificates and keys of the peers and orderers.

There are two main sections within the crypto-config YAML file: "OrdererOrgs", which is defined as the ordering service, and the "PeerOrgs", where the standard organizations of the health network are registered. The EnableNodeOUs attribute allows the MSP to classify the participants' identities into Clients and Peers.

There are two sections in the "crypto-config" file:

- "OrdererOrgs", which is the section where we defined the ordering service.
- "PeerOrgs", which is the section where the health network's standard organizations are registered.

The "EnableNodeOUs" attribute allows the MSP to classify the participants' identities into Clients and Peers.

Within the PeerOrgs section, the number of nodes per organization to be created (Template->Count) and also the default number of transactional users for which the cryptographic material will be created (Users-Count) are set. The structure of the cryptogen file for the cryptographic material can be seen in Scheme 1, which includes the architectural design sketched in Figure 6.

5.1.2. General Configuration of the Genesis Block and the Channel

To create the initial configuration of the network, including the genesis block or initial block of the Blockchain and the channel configuration transaction, we proceed to create the file configtx.yaml (Configuration Transaction) and to use the program **configtxgen** which allows creating artifacts related to the channel configuration based on this YAML file. We define the orderer type organizations and the standard organizations within the network with their respective peers and initial configurations for this configuration. We also choose the ordering service that we are going to use, among which are: Solo, Kafka, and RAFT; besides, the link to the directory that has the crypto materials such as the policies (MSP) Membership Service Providers of each organization which manages the cryptographic operations, such as signing, verifying, issuing, and chaining.

Once we made the organization configurations, we proceed to set up the genesis block's general configuration and the channel, using the Profiles section. The Profiles section has attributes such as "Consortiums" where it is specified to which organizations the ordering service or Orderer will provide the service and "HistoriaClinicaChannel" which is the channel to which the configured organizations will join. As indicated in [12], each organization within the network can join more than one channel to segregate the permissions and improve the privacy of the data that will circulate through the health network.

```
OrdererOrgs:
    - Name: OrdererMinisterioSalud
      Domain: historiaclinicaelectronica.com
      EnableNodeOUs: true
      Specs:
        - Hostname: orderer
          SANS:
            - localhost
PeerOrgs:
    - Name: Eps1
      Domain: eps1.historiaclinicaelectronica.com
      EnableNodeOUs: true
      Template:
        Count: 1
        SANS:
          - localhost
      Users:
        Count: 1
    - Name: Eps2
      Domain: eps2.historiaclinicaelectronica.com
      EnableNodeOUs: true
      Template:
        Count: 1
        SANS:
          - localhost
      Users:
        Count: 1
    - Name: Eps3
      Domain: eps3.historiaclinicaelectronica.com
      EnableNodeOUs: true
      Template:
        Count: 1
        SANS:
          - localhost
      Users:
        Count: 1
```

Scheme 1. Structure of the cryptogen file of the POC.

5.1.3. Creation of Anchor Transaction Files

We create the anchor peers of our HCE channel for each health care provider. Members on a channel have one or multiple anchor peers, and all other peers can discover and communicate with them. This configuration avoids having a single point of failure.

5.1.4. Docker Services Definition

In this phase, we configure the file in order to have the peers use CouchDB. We made this design decision to increase performance, one of the QAs that the system has. Moreover, to obtain redundancy on the CA server and avoid bottlenecks and damage to the network, we also selected RAFT as a consensus method to have a fault tolerance attribute for architecture. We should note that in the POC, we used SOLO for security analysis purposes as the first architectural aim.

5.1.5. Channel Setup

We used the "Channel.tx" transaction created through the Configtxgen command to create the channel since it contains the information on how we want to produce it. Following the creation of the channel, we joined the different organizations to transact

within the network. It is necessary to use the identities and security certificates created using the cryptographic material to carry out this process.

5.2. Smart Contract Deployment

Peers invoke the chaincode or Smart Contracts when a doctor or healthcare provider within the network wishes to transfer or change an asset or EHR in the general ledger. We created a smart contract using the Go programming language to create and fetch an asset. We selected Go language because it is the Hyperledger Fabric core, so using another language such as Java will be less efficient because Fabric will convert that wrapper written in Java to Go. This architectural decision satisfies the latency and throughput required within the described architectural drivers. We used chaincode v2 for constructing the network, which uses a four-step deployment model.

- Packaging the ChainCode
- Installation on each peer (Packaged chaincode)
- Endorsement Policies (Approve a smartcontract definition for your organization)
- Chaincode Commit in the network

5.3. Execution and Testing of the Smart Contract

We invoke the Store method inside the Chaincode to test the creation of a medical record; in Figure 7, we can identify the document created using the Smart Contract in our distributed ledger.

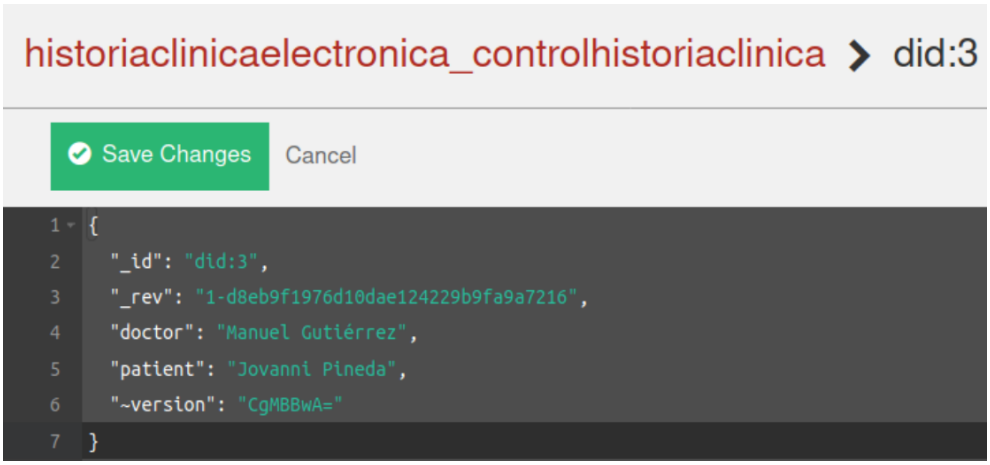

Figure 7. Creation of a EHR in the Ledger.

6. Security and Privacy Evaluation

To evaluate the proposed architecture through the POC performed, we rely on [7] which indicates that the approach to characterize security has three characteristics:

- Confidentiality
- Integrity
- Availability

Based on these three characteristics, we will analyze the security performance of this proposal in a descriptive analysis.

6.1. Confidentiality Assessment

The proposed architecture achieves patients' confidentiality and anonymity through different architectural decisions made in constructing the system:

1. The patient's identifier during the entire process of exchanging information with the system is the CA's public key so that the record of their actions does not contain personal information like their identity document.

2. The Hyperledger Fabric architecture channels allow for segregation of permissions to participants within the network, adding another layer of security over the patient's EHR.

According to the literature review made in our descriptive analysis, traditional architectures for the Health Sector do not use these characteristics, which on the other hand, are Blockchain's pillars.

6.2. Integrity Assessment

Within the proposed architecture, the endorsement policies provide another layer of security. No matter whether Hyperledger Fabric authenticates a participant or node within the channel, if that organization does not gain the proper permissions to endorse a transaction, and if they try executing the chaincode from a Health Care Entity that does not have the right to endorse the transaction in our architecture, the system will reject the operation.

We implemented Secure Hashing for the unique identification of participants in the decentralized network since the proposed architecture uses ECDSA (Elliptic Curve Digital Signature Algorithm) as the encryption algorithm to manage EHR data integrity in our Blockchain architecture.

We used the transport layer security (TLS) cryptographic protocol and hashing of the data to achieve integrity within our architecture, known as Digital Signature, where the receiver obtains the original data and the secure digitally signed hash. The receiver or patient node can recalculate the hash of the original data received and compare it with the received hash to verify the document's integrity.

Finally, there is no control of the integrity of both the transaction and the content of the patient information in traditional architectures. At the same time, Blockchain is based on a computational model to ensure its transactional integrity.

6.3. Performance Metrics

To test the performance of our architecture, we use Hyperledger Caliper, a benchmarking tool for measuring blockchain performance; it has multiple performance indices for measuring transaction throughput, e.g., number of transactions per second (TPS) or transaction latency [34].

We tested the patient's Store HCE chaincode method. For this, we used HL7 FHIR release four patient's as an example to simulate a patient storage transaction in the blockchain network. We took the information samples regarding the latency in milliseconds from related proposed systems [22,31,35] then we established the mean (average) latency of our architecture.

The results showed that our architecture has the third-best performance where the best result is the architecture proposed in [22] and the lower performance system is the one proposed in [22].The mean and other metrics results are shown in Table 10. However, and as is shown in Figures 8 and 9 our solution's average latency decreases as the number of transaction increases, so, our system demonstrates efficient latency and throughput metrics, for example, when working with 100.000 transactions.

Table 10. Latency and Throughput Statistical information in milliseconds.

Stats	Our Proposed Solution	PREHEALTH	MedRec—Azaria et al.	Blockstack—Ali et al.
Mean	150.67	58.00	77.07	530.00
Std	75.22	0.00	8.38	0.00
Min	100.00	58.00	63.20	530.00
25%	111.00	58.00	73.60	530.00
50%	122.00	58.00	79.60	530.00
75%	145.00	58.00	81.02	530.00
Max	300.00	58.00	86.90	530.00

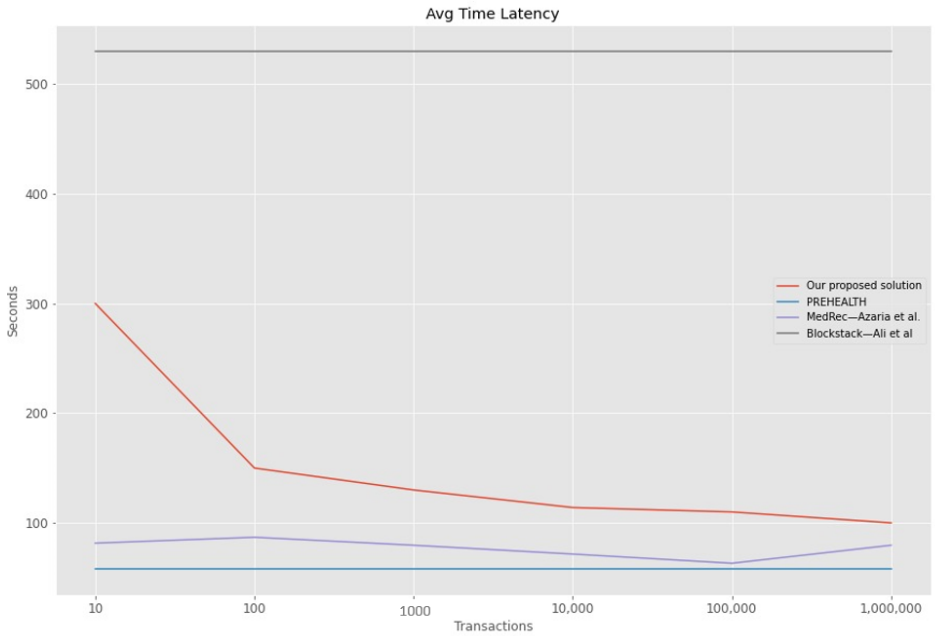

Figure 8. Average Latency in seconds—Line chart.

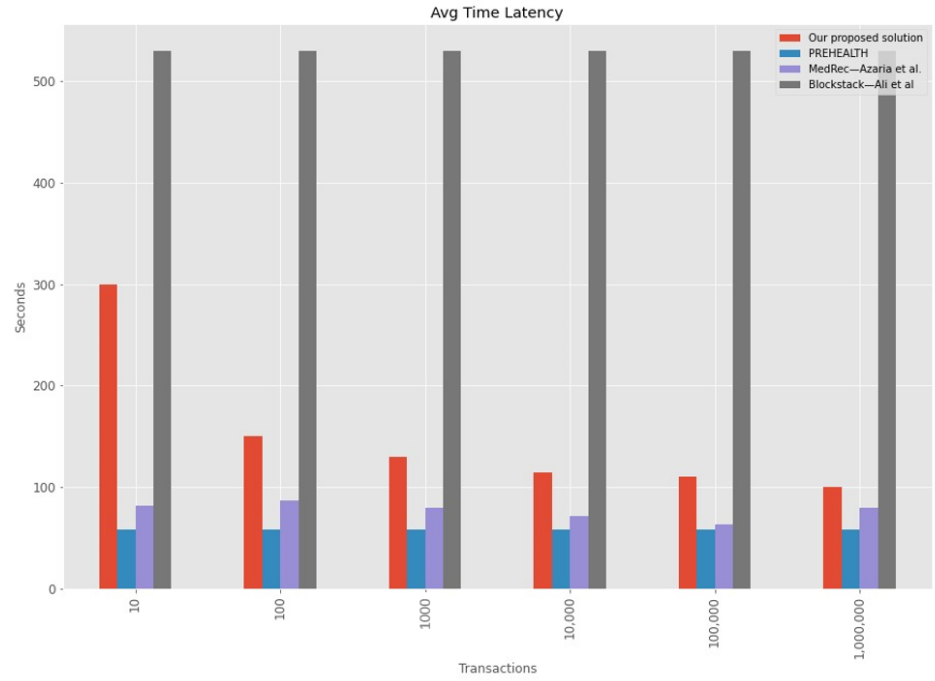

Figure 9. Average Latency in seconds—Barchart.

6.4. Availability Assessment

As a Peer to Peer architecture, Blockchain ensures that the information will be available on each network's participating nodes. In multiple architectures reviewed, we found a single point of failure that drastically affects information availability. On the contrary, Blockchain, and more specifically Hyperledger Fabric, distribute the information of each of the transactions in ledgers which synchronize their information through the Anchor Peers and the use of the Gossip data dissemination protocol that allows the information to be transmitted to the desired nodes within the network as we observed in the POC carried out.

6.5. Another Security Characteristics

The authors of [36] mentioned that there are four layers of enterprise security that we must consider when building a Blockchain-enabled technology solution, among which are the physical IT infrastructure layer and the middleware layer that includes requirements for encryption levels, encryption in data storage, transfer and data at rest, and visibility of data between network participants. Therefore, following these guidelines, we describe below how the proposed architecture satisfies these security layers with the proposed architecture.

According to our proposed architecture, communications within the health network use the cryptographic material created in Section 5.1.1 composed of the CA's digital certificates and private keys. Our architecture proposes that the Colombian Ministry of Health provide another additional security ring to host the CA, mainly because the Colombian government infrastructure will guard the CA's access and provide additional security to the Hyperledger network architecture to satisfy the isolation levels proposed.

Finally, and with Hyperledger Fabric's help, our architecture ensures that each copy of the ledger has the same data as we observed in the POC; hence, the system reaches the non-repudiation because of the append-only databases that avoid modifications in the sequence of log records. Thanks to each participant's unique identity within the network, Blockchain signs each action executed with its public and private encryption keys, which guarantees that a sender cannot deny a message. In the specific case of the EHR, the system will record any modification done by a healthcare provider to a patient clinical data permanently.

We observed that traditional architectures are based on the protection of resources, while our Blockchain-based architecture has an architecture oriented towards protecting information itself. What provides exciting features to guarantee the security of sensitive information. Additionally, we saw different approaches through which the proposed architecture provides security to the HCE system, which would be very difficult to implement through traditional architectures.

7. Discussion

Using a Software Architecture design process to satisfy the needs of a system is of considerable usefulness; we could evaluate how integrating architectural processes with the methodical analysis of literature support each of the architectural decisions and diminishes the possibility of electing an unsuccessful architecture. It is also important to mention that a badly constructed architecture would require consequent and costly adjustments to the original model or expose confidential information thanks to a misconstrued architecture.

We conclude that Hyperledger Fabric is a (configurable) pluggable blockchain architecture regarding the project's technology stack. It allows configuring and exchange the implementations on which software developers and architects want to work, such as the ordering service, which is crucial to operating its architecture. Moreover, as we can see in the results section, it will synchronize within the network any change to an asset in each ledger, which allows for achieving QAs sought in this paper.

Although blockchain architectures can be secure and efficient, there are still problems in the implementation and costs that these can bring, so some organizations still view this with suspicion, among which are the health organizations. This mistrust is because they are

novel technologies and provide little support and lack of integration with legacy systems, making it a problem in implementing existing solutions.

However, related to the technical issues associated with efficiency and security, we can demonstrate blockchain-type architectures' relevance to provide state-of-the-art technology to protect sensitive information.

8. Conclusions

In this paper, we proposed a software architecture design oriented to satisfy the architectural motivators of the EHR in the Colombian context, among others this includes the functional requirements and the QAs. A rigorous literature analysis empowers this architectural design process to make design decisions based on scientific works. We supplemented this analysis with a clear and detailed definition of the QAs through the Quality Scenarios, which allowed choosing some structures or models based on component-and-connector architectural styles. We evidenced this process through the choice of Blockchain as a reference architecture in the POC performed and in the security assessment made.

Applying tactics to refine the architecture design for security, such as the integrity verification of the assets (HCE) in the network through hash values, is one of the proposed architecture's pillars. Other pillars of this featured architecture are the identification, authentication, and authorization of the actors in the network, and access limits to the participants within the network through the segregation in Hyperledger Channels in order to detect, resist, and react to attacks, and can be a reference framework in contexts requiring secure storage and computation.

We can also conclude that in sectors that have legacy systems, the implementation of Blockchain is complex, as in the Colombian health sector, of which this study is part. The integration of these legacy systems based on traditional architectures poses a complicated technological implementation route due to knowledge factors and process stabilization, discouraging stakeholders from implementing these solutions.

Therefore and as future work, we are interested in building the complete Hyperledger Fabric system in a technological infrastructure according to Colombia's health system guidelines. We are planning to implement the proposed architectural design in one of Bogotá's hospital networks to perform the relevant anonymity and security tests on more realistic scenarios at the data and hardware configuration level.

We are also interested in performing an additional architectural design iteration to adopt the architectural drivers for architectural decisions that we did not consider in this paper. For example, the utility tree also describes the interoperability as a secondary QA, which it is interesting to integrate with HL7 FHIR. This interoperability will generate interest from multiple healthcare actors that currently use different health standards.

Author Contributions: Conceptualization and Investigation, E.A.P.R. and L.G.M.-S. Composed the manuscript and contributed to Scheme design and implementation, E.A.P.R. and L.G.M.-S. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank Politécnico Grancolombiano University and the Faculty of Engineering for providing me with all the resources and tools that were necessary to carry out the research process. Finally, I would like to thank LUMON SAS and the LUMON LV TECH research group for supporting the development of this article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EHR	Electronic Health Record
POC	Proof of Concept
EMR	Electronic Medical Records
ADD	Attribute Driven Design
ASR	Architecturally Significant Requirements
QA	Quality Attribute
DLT	Distributed Ledger Technology

References

1. Certsuperior. Robos de Datos en el Sector Médico. Available online: https://www.certsuperior.com/robos-de-datos-en-el-sector-medico/ (accessed on 10 September 2020).
2. Coventry, L.; Branley, D. Cybersecurity in healthcare: A narrative review of trends, threats and ways forward. *Maturitas* **2018**, *113*, 48–52. [CrossRef] [PubMed]
3. Lee, H.A.; Kung, H.H.; Udayasankaran, J.; Kijsanayotin, B.; Marcelo, A.B.; Chao, L.; Hsu, C.Y. An Architecture and Management Platform for Blockchain-Based Personal Health Record Exchange: Development and Usability Study. *J. Med. Internet Res.* **2020**, *22*, e16748. [CrossRef] [PubMed]
4. Esmaeilzadeh, P.; Mirzaei, T. The potential of blockchain technology for health information exchange: Experimental study from patients' perspectives. *J. Med. Internet Res.* **2019**, *21*, e14184. [CrossRef] [PubMed]
5. Salud, M.D. Resolucion 1995 DE 1999. Available online: https://www.minsalud.gov.co/Normatividad_Nuevo/RESOLUCI%C3%93N%201995%20DE%201999.pdf (accessed on 13 March 2020).
6. Ministerio de Salud y Protección Social—National Government Enacted the Interoperability Law of Electronic Medical Record. Available online: https://www.minsalud.gov.co/English/Paginas/This-is-How-the-Electronic-Medical-Records-Will-Work-in-Colombia.aspx (accessed on 19 March 2021).
7. Bass, L.; Clements, P.; Kazman, R. *Software Architecture in Practice*, 3rd ed.; Prentice Hall: Hoboken, NJ, USA, 2012.
8. Cervantes, H.; Kazman, R. *Designing Software Architectures: A Practical Approach*; Addison-Wesley Professional: Boston, MA, USA, 2016; p. 289.
9. Sajana, P.; Sindhu, M.; Sethumadhavan, M. On Blockchain Applications: Hyperledger Fabric And Ethereum. *Int. J. Pure Appl. Math.* **2018**, *118*, 2965–2970.
10. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Manubot* **2019**. Available online: https://git.dhimmel.com/bitcoin-whitepaper/ (accessed on 15 June 2021).
11. Jabed Morshed Chowdhury, M.; Ferdous, S.; Biswas, K.; Chowdhury, N.; M Kayes, A.S.; Alazab, M.; Watters, P. A Comparative Analysis of Distributed Ledger Technology Platforms. *IEEE Access* **2019**, *7*, 167930–167943. [CrossRef]
12. Hyperledger Fabric Docs—Documentation Release Master Hyperledger. 2020. Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/ (accessed on 13 September 2020).
13. Kuo, T.T.; Zavaleta Rojas, H.; Ohno-Machado, L. Comparison of blockchain platforms: A systematic review and healthcare examples. *J. Am. Med. Inform. Assoc.* **2019**, *26*, 462–478. [CrossRef] [PubMed]
14. Xia, Q.; Sifah, E.B.; Asamoah, K.O.; Gao, J.; Du, X.; Guizani, M. MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain. *IEEE Access* **2017**, *5*, 14757–14767. [CrossRef]
15. Xia, Q.; Sifah, E.; Smahi, A.; Amofa, S.; Zhang, X. BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments. *Information* **2017**, *8*, 44. [CrossRef]
16. Dubovitskaya, A.; Xu, Z.; Ryu, S.; Schumacher, M.; Wang, F. Secure and Trustable Electronic Medical Records Sharing using Blockchain. In *AMIA Annual Symposium Proceedings*; American Medical Informatics Association: Bethesda, MD, USA, 2017.
17. Fernandes, A.; Rocha, V.; da Conceicao, A.F.; Horita, F. Scalable Architecture for sharing EHR using the Hyperledger Blockchain. In Proceedings of the 2020 IEEE International Conference on Software Architecture Companion (ICSA-C), Salvador, Brazil, 16–20 March 2020; pp. 130–138. [CrossRef]
18. Areej, S. Which Programming Language Is Best to Use for Fabric's Chaincode? 2020. Available online: https://www.researchgate.net/post/Which-programming-language-is-best-to-use-for-Fabrics-chaincode (accessed on 20 February 2021).
19. Chen, X.; Zhao, S.; Wang, C.; Song, H.; Jiang, J.; Qi, J.; Li, T.O.; Chan, T.H.H.; Wang, S.; Cui, H. Efficient, DoS-resistant Consensus for Permissioned Blockchains. *arXiv* **2018**, arXiv:1808.02252.
20. Valenta, M.; Sandner, P. Comparison of ethereum, hyperledger fabric and cord. *Frankfurt School Blockchain Center* **2017**, *8*, 1–8.
21. Cachin, C. Architecture of the hyperledger blockchain fabric. In Proceedings of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers, Chicago, IL, USA, 25–29 July 2016; Volume 310, p. 4.
22. Stamatellis, C.; Papadopoulos, P.; Pitropakis, N.; Katsikas, S.; Buchanan, W.J. A privacy-preserving healthcare framework using hyperledger fabric. *Sensors* **2020**, *20*, 6587. [CrossRef] [PubMed]
23. HYPERLEDGER. Hyperledger Whitepaper. Available online: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf (accessed on 4 February 2020).

24. McAfee. Informe Sobre Amenazas Contra Blockchain. Available online: https://www.mcafee.com/enterprise/es-es/assets/reports/rp-blockchain-security-risks.pdf (accessed on 24 December 2020).
25. Atzei, N.; Bartoletti, M.; Cimoli, T. *International Conference on Principles of Security and Trust*; Springer: New York, NY, USA, 2017; pp. 164–186.
26. Nitish, B.; Sugata, S. A Review of Techniques to Mitigate Sybil Attacks. *arXiv* **2012**, arXiv:1207.2617.
27. Hyperledger-fabric. What's New in Hyperledger Fabric v2.x —Hyperledger-Fabricdocs Master Documentation. Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatsnew.html (accessed on 30 May 2021).
28. Stellabelle. Explain Delegated Proof of Stake Like I'm 5 | by Stellabelle | HackerNoon.com | Medium. Available online: https://medium.com/hackernoon/explain-delegated-proof-of-stake-like-im-5-888b2a74897d (accessed on 22 August 2020).
29. Tang, Y.; Zou, Q.; Chen, J.; Li, K.; Kamhoua, C.A.; Kwiat, K.; Njilla, L. ChainFS: Blockchain-Secured Cloud Storage. In Proceedings of the IEEE International Conference on Cloud Computing, San Francisco, CA, USA, 2–7 July 2018; pp. 987–990. [CrossRef]
30. Do, H.G.; Ng, W.K. Blockchain-Based System for Secure Data Storage with Private Keyword Search. In Proceedings of the 2017 IEEE 13th World Congress on Services, Honolulu, HI, USA, 25–30 June 2017; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2017; pp. 90–93. [CrossRef]
31. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. MedRec: Using blockchain for medical data access and permission management. In Proceedings of the 2016 2nd International Conference on Open and Big Data, Vienna, Austria, 22–24 August 2016; pp. 25–30. [CrossRef]
32. Ichikawa, D.; Kashiyama, M.; Ueno, T. Tamper-resistant mobile health using blockchain technology. *JMIR mHealth uHealth* **2017**, *5*, e7938. [CrossRef] [PubMed]
33. Hyperledger. Hyperledger Fabric Docs—Cryptogen. Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/commands/cryptogen.html (accessed on 17 March 2021).
34. Matt, Z.; Brian, W.; Mark, M. *Hands-On Smart Contract Development with Hyperledger Fabric V2*; O'Reilly Media, Inc.: Newton, MA, USA, 2021.
35. Ali, M.; Nelson, J.; Shea, R.; Freedman, M.J. Bootstrapping trust in distributed systems with blockchains. *USENIX* **2016**, *41*, 52–58.
36. Nitin, G.; O'Dowd, A.; Novotny, P.; Desrosiers, L.; Venkatraman Ramakrishna, S.A.B. *Blockchain—An Enterprise and Industry Perspective— Blockchain with Hyperledger Fabric*, 2nd ed.; Packt Publishing Ltd.: San Francisco, CA, USA, 2020.