Design and Development of a Blockchain-based Consent Management System

by

Prasanth Varma Kakarlapudi

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Faculty of Engineering and Applied Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

August, 2021

THESIS EXAMINATION INFORMATION

Submitted by: **Prasanth Varma Kakarlapudi**

Degree Name in **Program Name**

Thesis title: Design and Development of a Blockchain based Consent Management System

An oral defense of this thesis took place on August 9, 2021, in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Khalid Elgazzar
Research Supervisor	Dr. Qusay H. Mahmoud
Examining Committee Member	Dr. Akramul Azim
Thesis Examiner	Dr. Jing Ren

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

Development of a Blockchain based Consent Management System for Private Data

Data management is defined as obtaining, processing, safeguarding, and storing information about an organization to aid in making better business decisions for the firm. With the proliferation of smart devices, the amount of data available to enterprises has expanded tremendously. They often share the information gathered across organizations without the consent of the individuals who provided the information. As a result, we must protect the information from unauthorized access or exploitation. As a result, companies must ensure that their systems are transparent to build user confidence. To accomplish this, we are confident that Blockchain properties will accommodate, as transactions recorded on the network cannot be modified and are accessible to everyone on the network. This thesis introduces the architectural design of a blockchain system for controlled private data management, discusses the prototype implementation using Hyperledger Fabric, and presents evaluation results of the proposed system using Hyperledger Caliper.

Keywords: Blockchain; consent management; data privacy; Hyperledger; Ethereum.

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Prasanth Varma Kakarlapudi

STATEMENT OF CONTRIBUTIONS

I hereby certify that I am the sole author of this thesis, and I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

A survey of Blockchain for consent management has been disseminated in the following publication:

- Kakarlapudi, P.V.; Mahmoud, Q.H. A Systematic Review of Blockchain for Consent Management. In *Healthcare* 2021, *9*, 137. https://doi.org/10.3390/healthcare9020137.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my research supervisor Dr. Qusay H. Mahmoud, for his continuous support, motivation, and patience. This thesis would not have been possible without his advice.

Finally, I would like to express my sincere appreciation to my family and friends for their fantastic support.

Table of Contents

LIST OF TABLES

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

LIST OF FIGURES

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

LIST OF ABBREVIATIONS

ACL	Access-control list
API	Application Programming Interface
AWS	Amazon Web Services
BTC	Bitcoin
CCPA	California Consumer Privacy Act
CMS	Content Management System
CPU	Central Processing Unit
DAPP	Decentralized application
EC-2	Amazon Elastic Compute Cloud
ETH	Ethereum
EVM	Ethereum Virtual Machine
GDPR	General Data Protection Regulation
GoLang	Go Programming Language
IoT	Internet of Things
IPFS	InterPlanetary File System
PII	Personally Identifiable Information
PoS	Proof of Stake
PoW	Proof of Work
SDK	Software Development Kit
SFTP	SSH File Transfer Protocol
SUT	System Under Test
TPS	Transactions per second
VM	Virtual Machine

Chapter 1.

Introduction

There were 4.66 billion active internet users globally in January 2021, accounting for 59.5 percent of the global population [1]. With the world rapidly changing in terms of technology, every digital activity is recorded and tracked, potentially revealing sensitive information. In 2016, Cambridge Analytica had access to 87 million Facebook users [2], which were acquired via users who were using a third-party application known as “This Is Your Digital Life,” where they unknowingly gave access to the app, which collected their information and their friend’s network information as well [3]. Utilizing extensive information, the company attempted to manipulate the voters (US Presidential Election 2016). Hence, the growing amount of data being recorded, particularly personally identifiable information (PII), poses significant security, privacy, and data ownership concerns.

The current EU General Data Protection Regulation (GDPR), for example, acknowledges these difficulties by requiring companies that consume personal information to get authorization from the individual whose information is being collected. Individuals should also be able to audit who has accessed their information. A consent management system (CMS) is usually used as a platform between the users and organizations to manage user's consent for using their data. It helps the users to either accept or revoke the requests from the organizations. But when a leading company such as Google, Facebook controls the system, users are obligated and forced to trust their system without any other choice.

Additionally, individuals are not aware of the data transfers or sharing between the organizations for multiple reasons, such as improving the quality of a service provided.

Blockchain is a superior technology that can be used to develop a consent management system for using an individual's private data as the data written on Blockchain cannot be changed, which also implies that consent information, once recorded, cannot be deleted. Consent information can be considered as a blockchain transaction, and it can be stored on the network. Traceability is one of the main features of blockchain, which will be beneficial to the users. For instance, when an asset is sold on the blockchain system, it will also contain the previous owner details recorded in a separate block making the asset details traceable. Thus, the blockchain's essential features, such as its immutability and traceability, would enable trust between the users and companies (data consumers), making it the best choice for developing a CMS.

1.1 Motivation

In a traditional data management system, data is stored in a database where data consumers/organizations can control access. In contrast, blockchain maintains a distributed ledger where data on the network is available to everyone. The availability of the ledger to everyone poses a significant data privacy issue as personal data stored on the network can be used by others on the network. Considering that all the members on the network are authorized to use the data stored on the network, it still violates one of the significant GDPR policies. According to Art. 17 GDPR (right to be forgotten), the organizations should delete the data from all their databases or any other source when a user requests to erase his/her data without any delay [4]. This creates significant issues as the data noted on the network cannot be erased or deleted. If the data is deleted from the network, it will still contain a

record of what it used to be due to its traceability nature. Also, storing personal data on the network would increase the block size. The block size would eventually increase the network latency [5]. In conclusion, personal information cannot be stored on the Blockchain.

The data can be stored in a separate location, and its hash reference can be stored on the network. The disadvantage with this approach is that the hash reference of personal data might also be considered sensitive information in the near future [6]. Overall, the challenges with adopting Blockchain for private data management are given below.

- Unwanted Access: Personal data saved on the network is available to all the members.
- Privacy Violation: Data cannot be deleted from the network as data once recorded cannot be deleted.
- Network Performance: Network is decreased when personal data is stored.

1.2 Research Questions

The objective of this thesis is to explore the latest blockchain technology for enhancing data privacy and consent management systems for individuals, private and public organizations. The primary goal of this thesis is to design and develop a blockchain-based consent management system that allows data to be simply and securely exchanged between organizations, with users acting as data owners and controlling the flow of their personal information. This thesis is aimed to answer the following research questions:

RQ1: What is the major issue with the current Consent Management System?

The question will identify the latest issues with the consent management system. By identifying the concerns, a reliable prototype can be designed and developed to address the concerns.

RQ2: How do the features of Blockchain contribute and benefit consent management?

This is the most critical aspect of this thesis since it aims to understand how blockchain technology could help with a consent management system and how its features can be leveraged to create a trustworthy system.

RQ3: What are the limitations in the current blockchain-based consent management solutions, and how are they addressed?

With this question, the limitations of current solutions are determined. By recognizing the limitations, a solid foundation for evaluating the built prototype is established.

1.3 Contributions

The main contribution of this thesis is the implementation and performance analysis of the Blockchain-based consent management for private data. The main goal of the thesis is that the proposed design, implemented, and evaluated will serve as a prototype for solving the challenges discussed in the earlier section.

- Comparative analysis of available blockchain type and platform for designing the framework.
- Design and development of a blockchain-based consent management framework for private data.
- Implementation details of the proposed model on AWS cloud with a case study.

- Performance evaluation of the developed prototype using Hyperledger Caliper, which is a benchmark tool for measuring the performance of Blockchain implementations.

1.4 Thesis Outline

The rest of this thesis is structured out as follows.

Chapter 2 presents the background of data privacy and consent management, including discussing a use case-healthcare. Furthermore, this chapter discusses Blockchain, its architecture, features, and evolution. Finally, it discusses the different relevant works that have been conducted in various domains such as health, the Internet of Things (IoT), and others, as well as the gaps in existing solutions.

Chapter 3 discusses the proposed consent management system concept and its architecture in detail. Following that, the characteristics of the proposed prototype are discussed. Finally, a few use cases of the prototype implementation are provided.

Chapter 4 discusses the Hyperledger and Ethereum platforms in detail and defines crucial terms. The subsequent parts will cover the implementation on a local virtual machine and in the cloud. Following the implementation details, the application's functionality is demonstrated using a healthcare use case.

Chapter 5 describes the Cloud implementation's evaluation results using Hyperledger Caliper. J-Meter is used to evaluate the local machine implementation. The measured response time and throughput are used to evaluate the system on local setup, whereas the Caliper metrics are used to evaluate the setup on the Cloud. Also, early results of the Ethereum based system are presented.

Finally, **Chapter 6** includes a conclusion as well as information on the intended future work.

Chapter 2.

Background and Related Work

Data management is an administrative process that includes acquiring, validating, storing, protecting, and processing required data to ensure the accessibility, reliability, and timeliness of the data for its users [7]. Big data is more used than ever by organizations and companies to inform business decisions and gain deep insights into customer behavioral patterns, trends, and possibilities to create a unique customer experience [7].

While there are many steps involved in data management, such as data quality, data analytics, data governance, data architecture, and master data management, this thesis is mainly concentrated on the data privacy aspect of data management.

2.1 Data Privacy

Data privacy concerns how data should be acquired, kept, maintained, and shared with third parties, as well as ensuring compliance with existing privacy regulations (such as California Consumer Privacy Act- CCPA or GDPR). In contrast, Data Governance is about the policies involved in building the content [8]. In other words, data privacy refers to the process, policies, and technology to protect sensitive information from unauthorized access and use, internally and externally [51]. Sensitive information is data to be secured from unauthorized users to safeguard an individual's privacy or security. A few examples of sensitive data include Date of Birth (DOB), Social Insurance Number (SIN), credit card information, health data information, etc.

The data privacy process consists of significant activities and is mentioned briefly in Fig. 2.1.

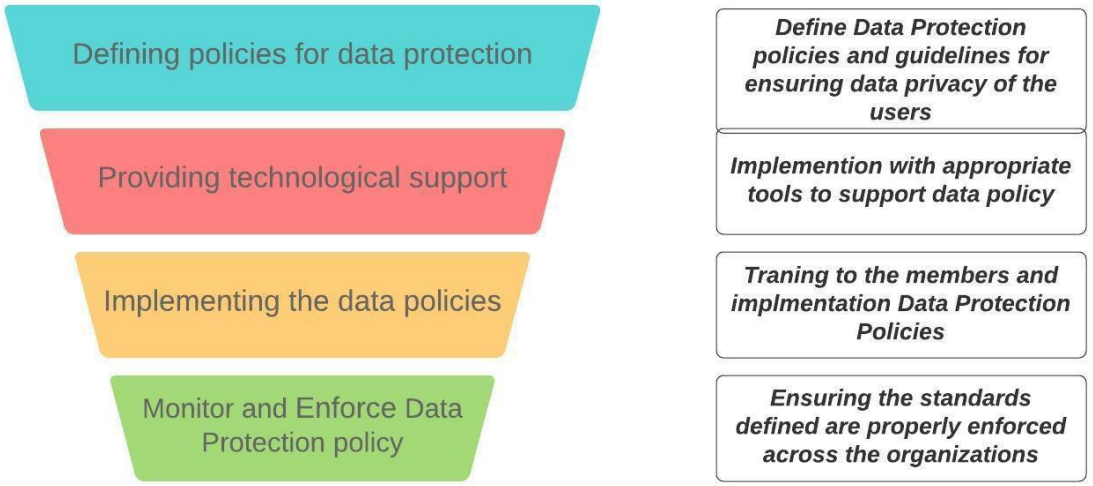


Figure 2.1 Data Privacy Process

The initial step involves defining the policies required to protect sensitive data from authorized access. Once the policies are defined, the second step includes implementing appropriate technologies, software, and tools that support the defined data protection policies. The next step is to enforce the policies and train the organization's members to follow the policies. The final step is to monitor and ensure the policies are practiced according to the standards across the organizations.

Usually, an individual is responsible for developing and implementing the data privacy policies using the latest technology (such as cloud storage systems to maintain availability), provide guidance on the processing of personal information, and deliver training to the organization's staff.

2.2 Consent Management

Consent management is a technique, strategy, or combination of policies that enables users to specify which information they are willing to share with various providers [31]. The following are the major phases in consent management: gathering consent, storing consent, and using the data collected.

- **Consent Collection:** A variety of sources can be used to obtain consent, including websites, mobile applications, etc. Consent management systems should collect all types of consent details, such as partial consent, etc. For example, a patient may be willing to share his data with a physician but may not share his data with a medical researcher, depending on the circumstances. In this instance, the user granted only partial agreement to the sharing of his personal information. As a result, the consent management system should be structured to allow for the acquisition of partial consent from the user.
- **Data Storage:** The data collected after obtaining the consent should be stored in a secure location as it might contain sensitive or non-sensitive information. The other main challenge is collecting only the user's required information instead of collecting complete data. Another critical factor is the duration of data storage because data cannot be stored for extended periods; instead, it can only be stored for the least amount of time possible. The organizations generally determine the duration of data storage, and they must also verify that the information is correct and up to date within that period.
- **Data Usage and transparency:** The collected information should be used for the reasons specified at the time of collecting consent. The system should enable the

organizations to request additional consent from the user when required. The system should be transparent and traceable. These capabilities will aid the user in auditing inter-organizational data-sharing exchanges. Finally, after the agreed period, data should be deleted from all the storage locations. Furthermore, data should also be erased upon user request.

Overall, an ideal consent management application should be capable of communicating the reason for data collection, storing it securely, and obtaining additional consent as needed.

2.2.1 Consent Management for Healthcare Research

In 2020, the volume of new healthcare data was identified to be approximately 2,314 exabytes [26]. It is one of the primary fields where a consent management system is needed as the patient's data is usually exchanged between multiple organizations for various reasons. The primary concern in healthcare and healthcare research is protecting the patient's/volunteer's data, which is extremely sensitive and considered confidential. The vital principles that should be followed during the healthcare research to protect the data are Collection Limitation, Data Quality, Purpose Specification, Use Limitation, Security Safeguards, Openness, Individual Participation, and Accountability [25]. The biggest problem in this approach is that the patient or volunteer is unaware of how their data will be used. Considering the following scenario; If a patient's data is shared between different organizations, the patient may not be aware of the extent to which the data has been shared. The patient is unaware of the information shared between doctors or hospitals. This is due to the CMS's lack of transparency. The patient should be aware of all the organizations that

have access to their data. Finally, collected data is used for unmentioned purposes during consent collection.

The key reason for selecting Blockchain technology is that it is developed for keeping transactions secure, which indicates that data cannot be erased from the network. Anyone can access the information available on the network. The following section goes into detail about blockchain technology.

2.3 Blockchain

A blockchain is essentially an immutable digital transaction log that is replicated and shared throughout the entire network on the blockchain. The methodologies used to carry out the confirmation and timestamping processes are implemented in software and mathematically assure that once approved, the details of the transaction described by the ledger cannot be changed by anyone, anywhere, without the application of more computing resources than the world currently has. Prior to blockchain technology, transactions (financial, etc.) were recorded using a centralized server and client-side model. In 1990, the idea of a secured chain of timestamps emerged from Haber, S and Stornetta, W. S. [9]. They sought to create a system that would be unable to alter the timestamps of a document. In 1998 Nick Szabo, a computer scientist, worked on the digital currency, 'bit gold,' [11]. Though it was not implemented, it was called the direct precursor to the Bitcoin architecture [12]. Stefan Konst published his theory of secure cryptographic chains, including the ideas for implementation in the year 2000.

However, Blockchain's concept benefited greatly in 2008 when it was used in Bitcoin as a distributed ledger technology (Bitcoin white paper published in October 2008) [10]. The mysterious Satoshi Nakamoto introduced Bitcoin to trade electronic coins

without a centralized party (banks, for example). Third parties are not always reliable, as they may become compromised, and there are transfer limits; additional amounts are also charged if a third party is associated with an ordinary transaction. Bitcoin overcome these limitations after its introduction. Bitcoin is a peer-to-peer cryptocurrency exchange that occurs without the involvement of a third party. It is resistant to fraud and is protected by sophisticated algorithms.

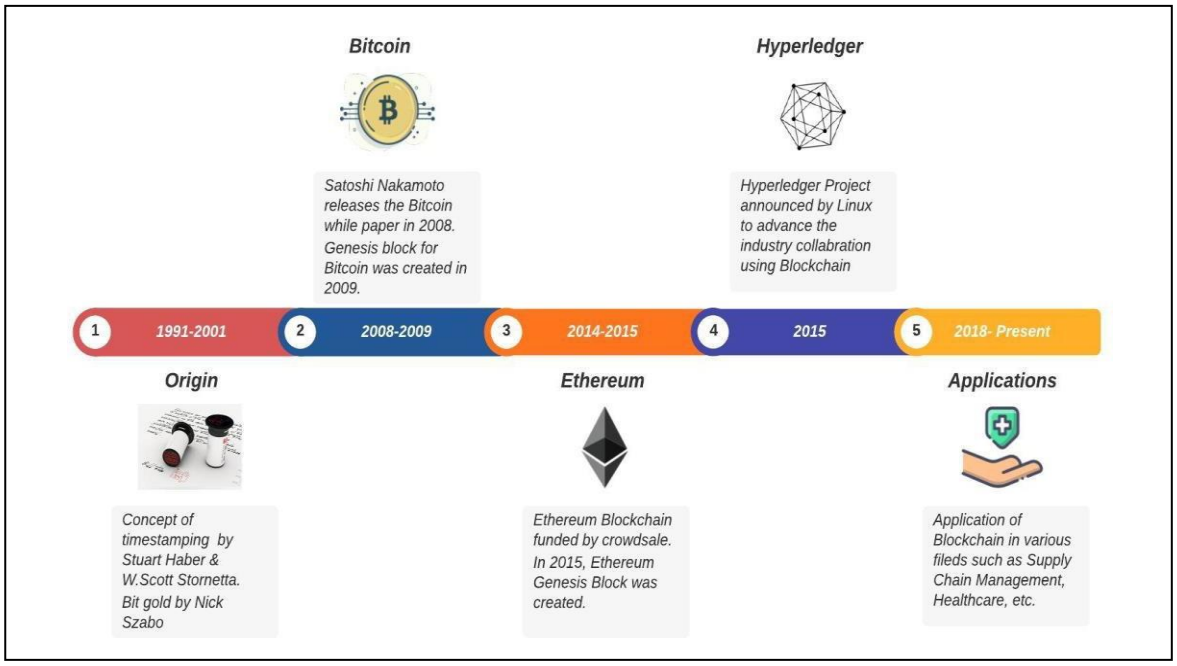


Figure 2.2 Evolution of Blockchain technology

From 2014, as shown in Fig. 2.2, blockchain technology is shifted away from cryptocurrencies, and its potential for other financial and inter-organizational transactions is studied. Blockchain 2.0, which refers to applications other than currency, is conceived. Ethereum, introduced by Vitalik Buterin's blockchain system, enters smart contracts (a computer program) into blocks that represent financial instruments such as bonds. A smart contract is a program that can act as a protocol or an agreement, which cannot be tampered

with. This concept was introduced by Nick Szabo in 1994 [13]. Building applications on Bitcoin protocol was challenging, and it was one of the main motivations for creating Ethereum. The Linux Foundation launched Hyperledger, an umbrella project of open-source blockchains and related tools, in December 2015[16], with contributions from IBM, Intel, and SAP Ariba to support the collaborative development of blockchain-based distributed ledgers [52]. The project's mission is to strengthen cross-industry collaboration by developing blockchains and distributed ledgers, with a specific focus on increasing the quality and reliability of these systems [52].

2.3.1 Blockchain Architecture

As the name indicates, a blockchain is made up of a chain of blocks that are linked together cryptographically. While the Bitcoin blocks contain the details of financial transactions, the content of a block could be anything that can be represented digitally. The sequence of the blocks is vital. As the term "chain" infers, all the blocks are connected to each other in a fixed, unalterable order determined by the time the block is formed. A block contains the following significant information.

- Data (Financial Transactions in Bitcoin).
- The hash value of the block. The block hash functions similarly to a "fingerprint" in that it serves as an identity for your input data and is unique to each block.
- Previous block’s hash value.

As previously established, the type of Blockchain determines the data recorded in a block. The hash value of the block is generated when the block is created. The first block is called the genesis block and does not contain the previous block's hash value; instead, it

begins as zero. When a new transaction takes place, a block with the corresponding data is added. As a result, a new block is appended, forming a chain in the process, thereby the name Blockchain as shown in Fig. 2.3. In an instance where data is changed in a block, as the information is changed, its hash value also changes [3]. Therefore, this block's hash value and the previous hash value of the subsequent blocks will not be the same, resulting in breaking the chain and making it invalid [3]. Still, there is a chance of recalculating the remaining block's hash values to make it valid again with the help of advanced supercomputers [3]. But recalculating the hash values would require enormous resources making it almost impossible.

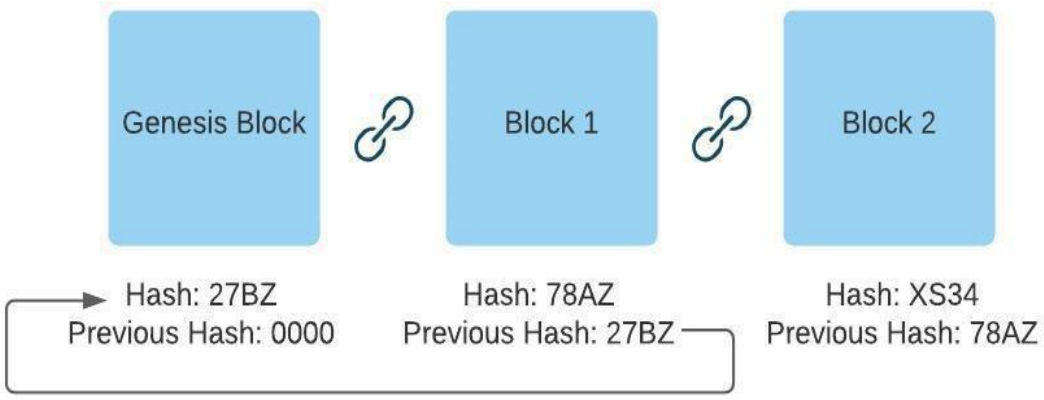


Figure 2.3 Working of a Blockchain

Blocks are not directly added to the network when a transaction is made. In Bitcoin, Proof of Work (PoW) is used to add blocks after verification and validation. It also makes sure that tampering no user may spend any of their holdings more than once. The PoW is a process where a miner creates a transaction with one or several unconfirmed transactions. Every peer in the network has the capability of becoming a miner. Miners gather all

pending transactions from the decentralized network before guessing a random number (nonce) to solve cryptographic puzzles [14]. Usually, miners compete to solve the puzzle, and whoever finds a solution can broadcast it to the network so that the other can validate it. After the validation, the block is added to the ledger. The ledger is public or "distributed" to prevent tampering; other users would quickly reject an altered version. Once the verification of a block is complete with PoW, another block will soon follow the chain. Every continuation of the ledger is notified to every member on the network.

Proof of stake (PoS) was introduced to solve the high energy consumption of Bitcoin mining. Similar to PoW, it is also used to mine and validate the block transactions. The main difference is that in PoS, the person with more coins will have more mining power. PoS miners are only allowed to mine a percentage of transactions corresponding to their ownership stake [15]. For example, a miner who holds three percent of the available coins can theoretically mine only three percent of the blocks.

2.3.2 Features of Blockchain

The core features of Blockchain are.

- **Immutability**: The inability to change or modify anything is referred to as immutability. This is one of the most essential blockchain features for ensuring that the technology remains as it is – a permanent, unchangeable network. Nobody can add transaction blocks to the ledger without the consent of most nodes, and it cannot be edited, deleted, or updated by any user within the network.
- **Decentralized**: The network is decentralized, and it refers that it does not have any government or a single person who is responsible for the framework. Instead, the

network remains decentralized through a group of nodes. Anything from cryptocurrencies, essential documents, contracts, and other valuable digital assets can be stored. You can control them directly via your private key with the help of blockchain.

- **Securit**y: All blockchain information is cryptographically hashed. In basic terms, the network information conceals the underlying nature of the data. Any input information is provided with a mathematical algorithm, which generates a different type of value, but the length is permanently fixed. Every block in the blockchain has its own hash and contains the previous block's hash. All hash IDs will be altered if any change or attempt to tamper with the data. It is also impossible to reverse the hash and very hard to bypass.
- **Consensus**: Consensus is a method for active nodes in a network to make decisions. The nodes can gain consensus instantly when very few nodes are on the network. A consensus is required for a system to work smoothly when millions of nodes are validating a transaction. The consensus is to be responsible for the network's lack of trust. The algorithms at the center of the system can be trusted, even if nodes don't always trust one other. As a result, every network choice for the blockchain is a win-win situation.
- **Distributed Ledgers:** A public ledger will typically give all relevant information about a transaction and its participants. The justification for private or federated blockchain, on the other hand, is a little different. In some cases, though, a large number of participants can see what is happening in the ledger. It is because all other users on the system maintain the network's ledger.

- **Faster Settlement:** Traditional financial systems are slow and inefficient. Processing a transaction once all agreements have been fulfilled can take several days. It's also simple to tamper with. Blockchain enables a faster settlement than traditional financial systems. This enables a user to send money more rapidly, saving time in the long run.

2.3.3 Private Blockchain

A participant's identity is anonymous in a public blockchain. It implies that anyone can hold a crypto address anonymously without revealing their identity. But participants can always desire greater transparency and accountability for the blockchain's administration, which is not feasible with a public blockchain. As a result, businesses want to establish a private blockchain. Private blockchains are often known as permissioned blockchains. Private blockchains are usually managed and administered by an entity (a single trusted individual or a joint venture entity). Unlike the public blockchain, permission is required to join the network. In addition, the trusted individual will set up the network and create an interface for the participants to use, allowing them to record transactions on the network. Private blockchains can be used in the business sector when information needs to be shared solely among a few nodes. A group of banks, for example, may create a private blockchain where transaction details are only communicated with the relevant parties.

Private and public blockchain systems are simply a distributed ledger that keeps track of all transactions between the users. But there are a few differences between them that are considered while choosing a platform for the CMS. The main differences between them are given in below Table 2.1.

Table 2.1 Comparison between public and private blockchain

Feature	Public Blockchain	Private Blockchain
Approach	Anyone can access the network without any permission.	Permission is required to join the network and to make a transaction.
Speed	Public blockchains are slow because the entire network must reach an agreement.	The number of participants is less and hence quick network speed.
Cost	Transactions costs are high.	Transaction costs are low.
Efficiency	A public blockchain platform will experience network congestion, resulting in slower speeds.	Private blockchains often have a small number of nodes. As a result, they are always effective.
Immutability	Completely immutable.	Partially immutable. In certain conditions, authorities have the authority to remove a block if they believe it is no longer appropriate.
Decentralization	A public blockchain is completely decentralized	A private blockchain is more centralized as it involves a central

		authority for the network administration.
Examples	Ethereum, Bitcoin	Corda, Hyperledger Fabric

In the further sections, the evolution of Blockchain technology from cryptocurrency to integration into various fields such as healthcare and education is discussed.

2.4 Related Work

When it comes to research, blockchain and its different forms and implementations make quite an important topic. The sections that follow discuss the use of Blockchain for consent management in diverse domains such as healthcare, the Internet of Things (IoT), identity management, and data storage. There are a few generic consent management systems included as well.

2.4.1 Blockchain for Consent Management in Healthcare

According to researchers, healthcare is one of the critical areas that could profit from Blockchain technology. Because it is based on the distributed ledger concept, medical records may be easily exchanged across hospitals/doctors/researchers for a variety of reasons, including maintaining a patient's data. MedRec is an implementation based on Ethereum that serves to keep and maintain auditable history and record of the medical transaction for providers, regulators, and patients [17]. Because it is based on Ethereum, different incentives are provided to miners who tend to authenticate the transaction. They also consider a second incentive technique that involves medical experts in the process of mining. Now, the process of mining makes it a little tricky because it includes gas prices for running the function of smart contracts, and it also raises security risks. Data sharing

implementation is discussed by Liang, X, and other authors through mobile applications using Blockchain [18]. However, data sharing is not discussed thoroughly by this implementation, like how information sharing occurs among firms. The utilization of health data is precluded by this design for research purposes. In addition, Medichain is considered a system that combines off-chain storage and Hyperledger Blockchain to store information related to healthcare [19]. Additionally, the proposed framework focuses on offering privacy and secrecy to users. However, it considers Hyperledger Composer and does not consider the implementation outcomes. With the use of Blockchain, Swetha, M. S., and the team discuss the system and framework for securing and protecting healthcare systems [20]. A permission-based blockchain, in [21], is presented with authority proof for healthcare data sharing. An emergency access control management system (EACMS) is introduced in [22] with the assistance of a Hyperledger composer. Tith, D. and his team presented a framework based on Hyperledger that is installed on a local network of 4 Linux-based computers and serves as a user interface for patients and clinicians [44]. An E-Health consent management framework using the Hyperledger Fabric on the IBM Blockchain platform was presented in [45]. The study included the deployment details of three providers (One patient and two providers). CrowdMed addressed the limitation of information sharing motivation by rewarding patients to provide more data for research reasons via reward tokens and a creative cost structure [46]. The evaluation results for the proposed framework have not been discussed.

2.4.2 Blockchain for Consent Management in IoT

This is another area where the application of Blockchain technology has the potential to alter the world. The Internet of Things refers to apps that are connected to the Internet

and capable of communicating with one another. While there are numerous benefits to IoT devices, the primary problem is that they gather sensitive information such as our location and can be used to create or understand a user's behavioral patterns. As a result, the integration of Blockchain technology with IoT devices will make privacy concerns nullified. The studies [32,33] examined the use of Blockchain technology in the IoT domain and the accompanying limitations. Cha, S.C., and others proposed the design of a blockchain-connected gateway that adaptively and securely respects user privacy settings for IoT devices on the blockchain network [33]. To ensure the security of IoT data, Sabrina F. developed a solution that addresses privacy data concerns. For entitlement management and control, a solution is a service-oriented model that utilizes a combination of public Blockchain (with smart contracts) and off-chain data [34]. The architecture is built on Ethereum and aids in the security of confidential data. In [42], the team introduced an Ethereum-based system for managing data collected from IoT devices. The prototype also complies with the GDPR. It enables users to manage their consent and, as a result, creating their data access policy [42].

2.4.3 Blockchain for Consent Management in Identity Management

Traditionally, personal identity is established by the use of documents such as a social security number, driver's license, or passport. However, there is no equivalent approach for guarding online identities that is nearly as effective [35]. A digital identity/ID can be produced and used in place of real identities for online transactions using blockchain technology. As it is immutable, there are very few chances of online fraud. Alan Colman and his team provide a novel method for archiving critical educational documents in [36], which they implement using Ethereum. The authors presented a system for storing data and

authenticating education-related documents, with the University or College doing the authentication and storing the documents on the Blockchain. We can always request verification because the information in Blockchain cannot be altered.

2.4.4 Blockchain for Consent Management in Data Storage

While direct sensitive information cannot be stored in a Blockchain network, encrypted data can be. One of the primary applications of Blockchain is the capacity to store data in conjunction with third parties such as the Cloud. They presented a novel technique termed interest groups in [37], in which each group adheres to a set of field data. Now, groups can sell, borrow, or rent the data they own. They also discussed possible incentives for a group that provides the most relevant information. Alessi, M. and his team developed a prototype [38] using Ethereum and IPFS (InterPlanetary File System). The prototype can store personal data and also provides requested data services.

There were also other areas, such as agriculture, that make use of Blockchain. In [39], the authors discuss the critical nature of farmer consent when utilizing Blockchain. Few prototypes are also proposed that are not domain specific. In the study presented by Agarwal, R. R. and the team, a generic consent management system- Consentio is designed and deployed on Hyperledger Fabric [43]. It mainly focuses on ensuring higher throughput and low latency for the transactions. Another generic CMS is presented in [47], where the framework is also based on Hyperledger Fabric. Users will be able to view a list of available companies in the presented system. They could either grant permission or change an existing one based on the list.

2.5 Gaps in Existing Solutions

It is, however, important to note that in the studies that have been mentioned, complete details about the implementation of the prototype have been discussed by a few. Not all systems that have been implemented have had discussions about how they are evaluated. Additionally, there exist privacy concerns and risks in prototypes. Ether is required by Ethereum implementation for invoking a function or operation, or for the mining process which is not suitable for the management of private data. Some implementations also considered the storage of personal data's hash references in Blockchain. It might lead to theft of data if the data is not specifically secured correctly in different off-chain locations and places. Constantly, blockchain is evolving and proposed systems must be advanced and updated according to it. For example, when Hyperledger Composer is involved, implementation is no longer valuable and valid because it is depreciated. Thus, a private system for data management should be designed and implemented that aims to address and manage all the challenges like storage of personal information in blockchain with the latest version of Blockchain Technology.

2.6 Summary

Blockchain is a broad issue with a variety of resources and applications. A brief overview of the many aspects of Blockchain technology that are relevant to our thesis has been provided. This encompasses the evolution of Blockchain technology, the working of Blockchain technology, particularly the features of Blockchain that are highly essential. The fundamental concept of data privacy has also been covered. Related work and the gaps in earlier proposed frameworks/implementations are also discussed in detail. The proposed framework will be discussed in greater detail in the following chapter.

Chapter 3.

Blockchain-based Consent Management System

In this thesis, we have concentrated on the development of a blockchain-based consent management system for personal data. To accomplish this, the proposed model was developed using a private blockchain. A private blockchain that does not charge a fee for writing to or reading data from the network, unlike other public blockchains such as Ethereum. Currently, a private blockchain is a widely considered blockchain system for developing enterprise solutions. This chapter discusses the proposed model in detail. It is designed so that; personal data can be stored off-chain in a cloud database and write only consent information on the Blockchain. A thorough discussion of the proposed system's architecture is provided. Additionally, it discusses a few applications for the proposed model.

3.1 Architecture

Considering the fact that personal data cannot be saved on the Blockchain due to some privacy guidelines, it was not considered. In addition, storage of hash references was avoided as personal data hashes might be referred to as personal data, according to researchers in the not-too-distant future [6]. The architecture of the system is shown in Fig. 3.1.

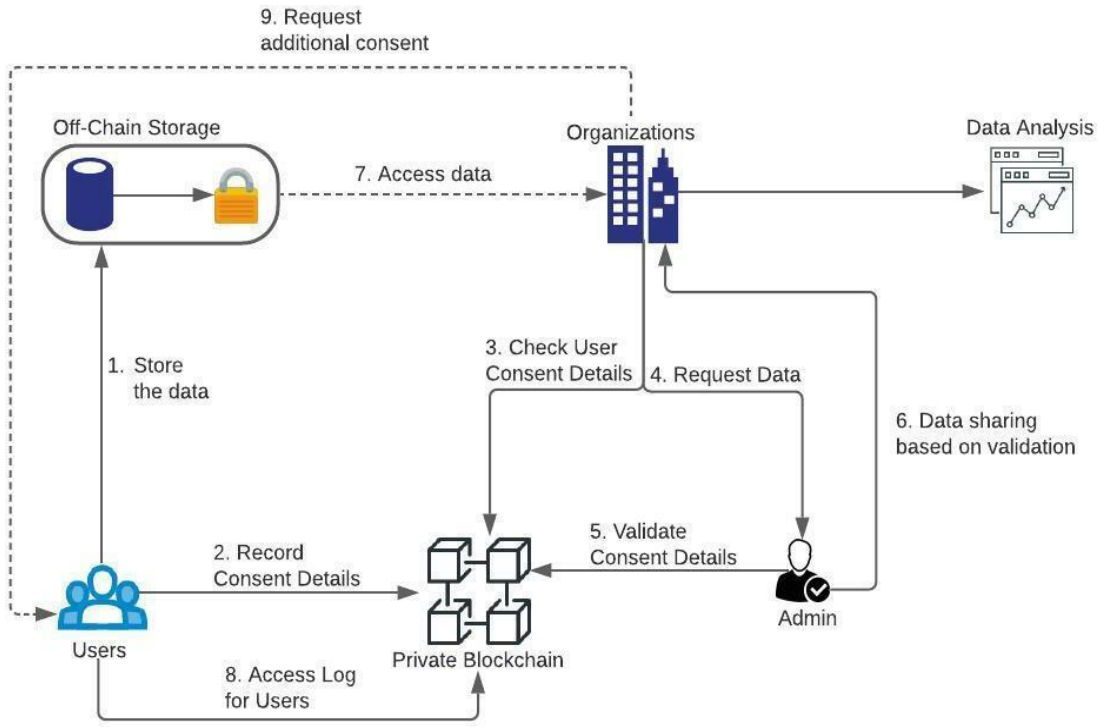


Figure 3.1 Proposed Solution's System Architecture

A blockchain is an immutable and tamper-proof ledger maintained by the nodes/users on the network. It does not require a third party to maintain the transactions. Instead, the ledger is maintained by the nodes on the network by using a consensus process to update the ledger's state. In a permissionless/public blockchain system, anyone can join the network with an anonymous identity. Costly techniques such as Proof of Work are used to determine the next block of transactions.

In contrast, nodes are not anonymous in permissioned blockchain systems. Approvals are required for a node to join the network. Private blockchains and traditional database systems are both centralized systems with a centralized authority. However, the

unique features of blockchain such as immutability and digital signatures make private blockchains more preferable. In the following upcoming sections, the working of the prototype is discussed in detail.

3.1.1 Role of Admin and Integrity Relationship Assumptions

Sharing of data to organizations, in our design, is controlled and managed by an admin. The admin will share data with the requested organizations when there are enough consent details provided by the user on the network. Additionally, an admin is required to ensure the maintenance of the database and that all data is deleted from the database and the organization's database upon the user's revoke request. The admin will also perform audits to make sure there is no unlawful storage of the data. The admin will be a trusted individual. For example, the admin can be from the government when the health information is involved. Therefore, the following integrity relationships are assumed:

- Users trust the admin for sharing their data and information with authorized organizations.
- Users are enrolled and registered successfully by the admin so that organizations can use it for invoking the functions of chaincode.
- All privacy laws would be followed by the research organizations, such as deleting the data when consent is revoked and evading unlawful data storage.

When more organizations are added to the network, the number of transactions will be increased. Therefore, the admin will have to handle more requests. In this case, the system can have multiple admins to make sure the requests are handled immediately.

3.1.2 Off-chain Storage

Motivation and information regarding data collection for the research (data collection) will be described to the user so that users are informed appropriately. Users, upon understanding the reasons for collecting data, can sign up for the service (through the user front end), as shown in Fig. 3.2. After signing up, they can store their data on a cloud database that is both secure and private. Furthermore, security features like using a private link for accessing stored data and blockage of public access are utilized. Important periodic actions and precautions would be taken by the admin for ensuring that the database is secure and healthy.

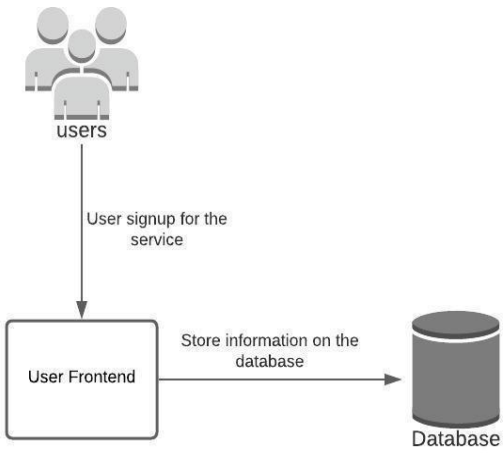


Figure 3.2 User signup and data storage

The following section details the Blockchain activities that take place on the network.

3.1.3 Blockchain Network

After signing up for the service and storing the uploading the relevant information on the database, consent is written or recorded on the network of Blockchain based on the Hyperledger Fabric (private blockchain) using the user front end. A private blockchain is

used for the proposed system for the following reasons: speed, efficiency, and no transaction cost. Details of consent include id (it is generated at the time of signing up for the service), name, email, consent details (partial/full), and organization details. For a healthcare instance, there exist a few types of research such as Prevention, physiological, and observational research [23]. It can be mentioned by a user if he is willing to offer partial (only to a specific type of research) or complete access. With the recording of the consent details by a user on Blockchain, they can be verified by the healthcare admin and access can be provided to research organizations as shown in Fig. 3.3.

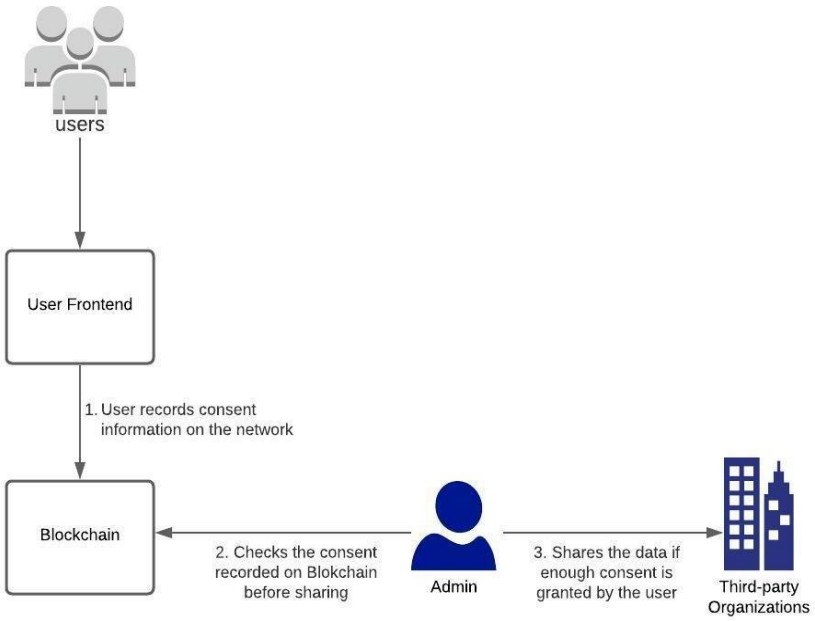


Figure 3.3 Data sharing after enough consent is provided by the user

With the use of the organization frontend, consent details of the user can be seen by the organization on the network, and access can be requested from the healthcare admin if the user has offered complete access. For example, if a user has given ‘Complete consent’

value in the consent field and ‘Any organization’ in the organization fields, then the data can be shared with all organizations that request the data. In this case, the admin will share the data immediately.

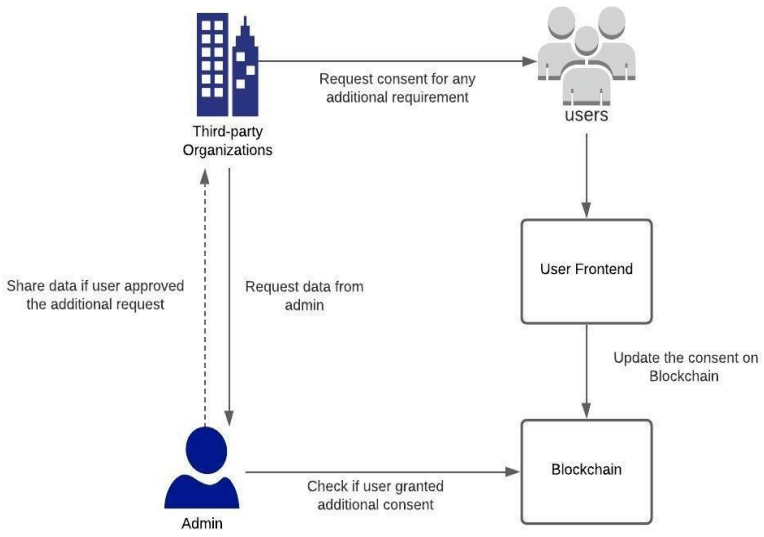


Figure 3.4 The process when additional consent is required

Additionally, if extended access is required, it can be explicitly sought through the organization's front-end interface (or website). The user can rescind or accept the request, and the information will be stored in the network as a transaction. After the request has been approved, the admin can share data with the organization on an as-needed basis. This process is depicted in Fig. 3.4.

When a user wants data deletion, the details can be recorded on the network. It will be updated as a form of transaction. The user must also notify the admin to erase his/her data. The admin checks the details from the user and deletes the data from the database immediately. Later, the information regarding the revoke request will be updated to the

respective organizations, informing them to delete data from their database or any other source. The process of deleting data from all sources is summarised in Fig. 3.5.

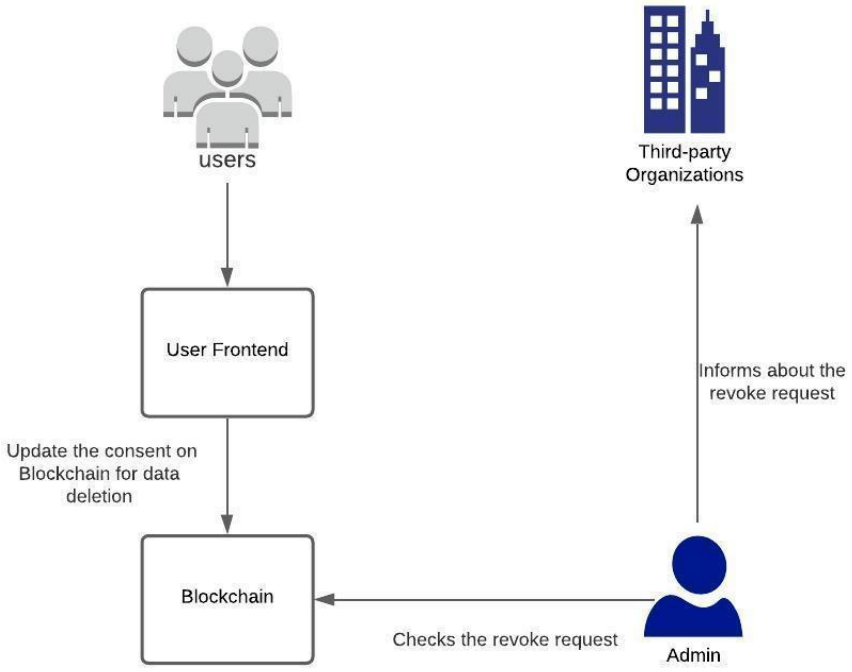


Figure 3.5 The process when the user requests data deletion

The next section discusses the chaincode used for the system.

3.1.4 Chaincode

The smart contract (chaincode) is installed on the network with a few key functionalities. Chaincode is a piece of code that is written in one of the supported languages, such as Go or Java [40]. It is installed on the peers, allowing communication with the network's shared ledger. The main functions of the chaincode are to record consent information from the network, query the user consent details, and provide the history information. The ledger's history information functions similarly to a log for users, allowing them to view the list of organizations with which they have shared data.

The organizations should join the network and install the chaincode on peers to use the chaincode functionalities. The pseudocode of a few functions from the chaincode is given in the below Fig. 3.6.

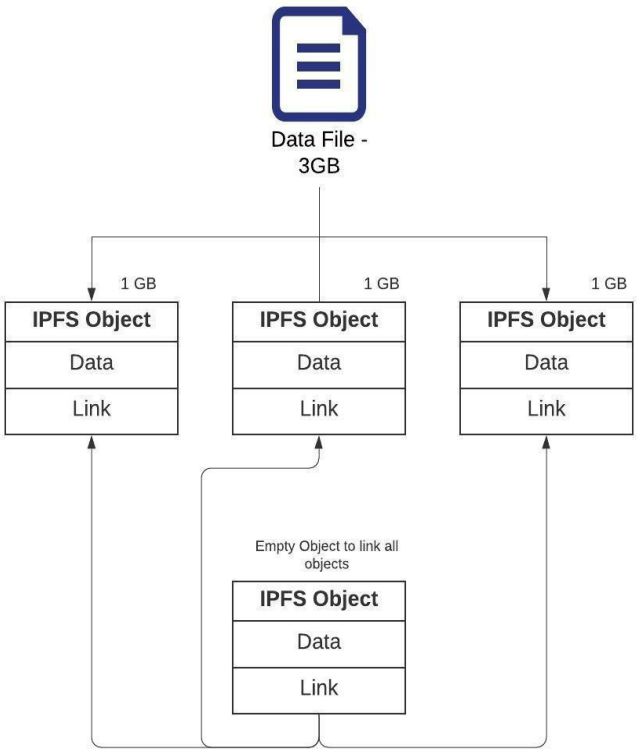
```
Input: User Consent /* It contains ID, Name, EMAIL, Consent, Organization */
Output: Message /* Success or Failure of adding new consent*/
Function CreateConsent:
    if User Consent details format is correct then
        Details are written on the Blockchain
        Return Success Message

    else if User Consent details (ID) already exists then
        Details are not written on the Blockchain
        Return Failure Message

Input: User ID
Output: Display details /* Details exist or does not exist*/
Function ReadConsent:
    if User ID details exist on Blockchain then
        Query the Blockchain for user details based on ID
        Return Display Details

    else if User ID details does not exist on Blockchain then
        Return Failure Message
```

Figure 3.6 Pseudo-code of the chaincode

3.1.5 Data Sharing

For data sharing with different research firms, two different strategies have been considered, including the use of AWS and IPFS. Ultimately, the strategy tends to rely on the location of the research firm and the data size. Over AWS, smaller files are shared as it would serve to evade personal data replication or duplication. In addition, stringent policies such as Access Control List (ACL) and guidelines will be considered by the admin while considering the sharing of data. Organizations, after the time period, will not have any access to the provided data or files. Through IPFS, large files can be shared as shown in Fig 3.7. There are many advantages of sharing the file through IPFS. When we use IPFS to host our static websites, we can avoid the risks associated with single points of failure

and reap the benefits of a distributed infrastructure. Once the period is over, files will be deleted to make the data safe. The system can be managed by making the admin restrict and manage access to information once the agreed period is completed or over.



-/

Figure 3.7 Data sharing with IPFS

It is important to understand that the admin must belong to a trusted and reliable government organization. Therefore, the trust between the admin and users will be quickly established. Using AWS or the cloud, information can be adequately secured. Security features of such an approach have been mentioned. Thus, our system can contribute to controlled and secure management of data that users can use and trust.

3.2 Features of the Proposed System

Table 3.1 presents the information regarding the features of the proposed system based on Hyperledger.

Table 3.1 Features of proposed blockchain solution

Feature	Issues	Solutions with our system
Blockchain Storage	The main issue with the Blockchain is that the sensitive data cannot be stored on the network, as it cannot be deleted from the network if the user requests it.	To avoid this, we have opted to store the personal data in a separate storage location. We have also avoided storing hash references of sensitive data on the network as the hash reference of the sensitive data might also be considered as personal information soon. We have used cloud storage instead, and the other advantage of not storing data on the Blockchain network is the network speed. The consent data can be fetched very fast.
Access log	The main issue with the current Consent Management systems is that the users are not aware	Users will be in control and can either accept or revoke the requests from the organization. Also, the chaincode installed on the network will allow the users to fetch the history information of

	of the organizations that are accessing their data.	their consent. This certainly brings out the traceability and transparency in our proposed system.
Security	The data stored on the cloud could be leaked if the database is not regularly maintained according to the latest standards.	Having a trusted individual to oversee the maintenance of the database will help make the system secure. Also, user revoke requests can be looked into quickly, and make sure that the data is deleted from all sources in the database. Data sharing through AWS will be influential and simple in removing the access to the organizations once a revoke request is placed by the user. Additional audits with the organizations can also be performed by the trusted entity to make sure the data is deleted completely from the organization's system.
Privacy	Unauthorized Users or malicious nodes.	Using a permissioned HF will make sure that there are no unauthorized organizations in the network. Also, additional attribute-based control could

		be set up to provide more granular access to the users with the help of chaincode. Malicious nodes can be detected easily by controlling throughput and potential users per second.
Scalability	Improving the system performance.	The system's throughput can be increased by increasing the amount of storage and the type of instances placed in the cloud. It is possible to do so by utilizing high configured EC2 instances such as t2 large, etc. Additional members can be added to the network by adding another docker swarm instance to it. As a blockchain solution, the system is theoretically indefinitely scalable
Hyperledger Fabric	The private data management systems should be adaptive to the fast-growing Blockchain technology.	We have used the latest fabric version which has the newer chaincode lifecycle. Additionally, we have used a React front-end to interact with our

		network instead of Compose which is now depreciated.

3.3 Use Cases

The proposed model is designed for multiple industries or areas with a private Blockchain platform. A few use cases for the proposed solution are provided below.

- **Healthcare:** The users could be patients or volunteers that share their data with the hospitals or research organizations. The admin would be someone from the government that will share the data. Hospitals/Research organizations could use the data from the volunteers to perform any medical analysis. So, to perform the analysis, consent is required from the users. The system could help them obtain consent and data from the users quickly. In general, it benefits both patients and organizations. The patients will have a list of organizations that have access to their data, and the organizations can utilize the system to achieve permission to access sensitive information.

- **Internet of Things:** Governments are establishing smart infrastructure in urban areas as a result of the development of IoT technologies. Citizens who use public infrastructure should be aware of who has access to their data and, if possible, regulate access to the data. They can choose whether or not to share data obtained via the latest infrastructure (electricity meters) with any other entities. People could be users, and the admin can be a trusted individual from the government.

- **Education:** The users can be the students in this use case. They can store their documents, such as transcripts, degrees, etc., on the database. The admin can be a

person working in the educational institute. The organizations could be firms that wish to hire students and require documentation for verification, etc. In this scenario, blockchain can also be used as an identity management application.

3.4 Summary

This chapter discusses the proposed model in detail, including a breakdown of the architecture. Additionally, we have discussed the proposed framework's solutions to cover the gaps in the earlier systems. The comparison between private and public blockchain is also discussed in detail. Additionally, a few use cases have been provided to demonstrate the proposed model. The following chapter will cover the proposed model's implementation technique.

Chapter 4.

Prototype Implementation

The implementation methodology for the proposed system is described in detail in this chapter. The chapter's first section discusses the platforms (Hyperledger and Ethereum) considered for implementing the system. The early system implementation details using Ethereum have been discussed. This chapter describes the system implementation details employing Hyperledger on a local virtual machine and on the cloud. The advantages of implementing the proposed framework on a multi-host cloud over a local virtual machine are also discussed.

4.1 Platforms Considered for Prototype

Both private and public blockchain systems share specific characteristics, such as immutability and resistance to tampering. Tamper resistance is provided in both systems by full replication, and hash reference of the previous block is included in the next block. However, a few differences make the private blockchain system more advantageous than the public blockchain system for developing a CMS.

Ethereum (a public blockchain) and Hyperledger Fabric (private blockchain) are considered in the initial stages of designing the prototype. The Ethereum introductory paper was released in 2013 by Vitalik Buterin, the project's founder, prior to the project's launch in 2015 [48]. It is primarily used to develop decentralized applications (dApp). Hyperledger was introduced developed to build applications for use across various industries [49]. Table 4.1 presents the differences between Hyperledger Fabric and Ethereum.

Table 4.1 Comparison between Hyperledger and Ethereum

Characteristic	Ethereum	Hyperledger Fabric
Governance	Ethereum Developers	Linux Foundation
Operation	Permissionless, public	Permissioned, private
Smart Contract Language	Solidity	Go, Java, Javascript
Currency	Ether	None
Consensus	Proof of Work (PoW)	Pluggable Mechanism

If a CMS is developed based on Ethereum, then Ether is involved in every step, such as deploying a smart contract and invoking a function. Additionally, it requires a lot of resources in order to achieve consensus. In Hyperledger, the currency is not involved. It is a permissioned blockchain system where nodes require permission to join the network. It has a higher throughput than Ethereum [50]. The following sections discuss Ethereum and Hyperledger Fabric in detail.

4.1.1 Ethereum

Ethereum is used to build Decentralized Applications (DApp). The participants in the network are known as nodes. Agreements between two peers or nodes can be stored on the networks known as smart contracts. Smart contracts are written in Solidity, which is a high-level language whose syntax is similar to that of JavaScript. The platform's fundamental cryptocurrency is Ether. So, nodes use Ether as a currency to verify the transactions which are made by the other nodes on the network. Once the transactions are verified by the nodes, it is added to the Blockchain.

EVM is an Ethereum Virtual Machine that should be used by the nodes to deploy a smart contract onto the network. It is also responsible for calculating the complexity of the transaction and verifying the transactions. A smart contract complies with an Ethereum Bytecode and then the bytecode is run on the EVM. There are two types of Ethereum accounts- external and contract accounts. An external account will have an Ether balance which will have a private and a public key that can be used to make a transaction. Contract accounts also have an Ether balance and can be used to make a transaction. The main difference between them is that the contract account is associated with a smart contract instead of a human being. The transactions made by the contract account are due to smart contracts.

4.1.2 Hyperledger Fabric

It is an open hub enabling enterprise-grade blockchain initiatives to incubate and mature through all stages of development and commercialization [27]. Under Hyperledger, there are many different frameworks such as Fabric, Besu, Iroha, Sawtooth, and Burrow. Hyperledger Fabric is one of the frameworks that is used for developing enterprise applications with a modular architecture [27]. It's designed to be a component-based system with plug-and-play features, including pluggable consensus and distinct membership services for different user roles [27]. Fabric blockchain runs smart contracts in the form of programs called chaincode, and transactions are the only way to interact with a chaincode [27]. Only endorsed transactions may be committed to the blockchain and update the global state [27]. Hence all transactions on the network must be endorsed [27]. There are two sorts of transactions that can use Chaincode.

- **Deploy transactions:** Allow for the creation of a new chaincode, with the code as a parameter. The chaincode is deemed placed on the blockchain once the transaction has been validated and executed successfully.
- **Invoke Transaction:** Allow for the execution of a chaincode program on the blockchain. An invoke transaction instructs the client to run a specific function from a chaincode. An invoke transaction results in successful chaincode execution and, as a result, modification of the local/global state with a returned output.

In Hyperledger, there are some essential terms, and they are thoroughly explained.

- **Peer**: In general, peers are quite similar to participants or nodes present on the network and tend to share and use the ledger privately. For instance, for Blockchain and Ethereum, all the nodes are the same and equal. In Hyperledger, however, there exist several peer types, including endorsing peer, committing peer, and anchor peer. Outside the network, anchor peers are determined and identified. Additionally, in the absence of an anchor peer, the connection between two networks becomes impossible. Committing peers are accountable for the maintenance of the network ledger. Meanwhile, for validation purposes, endorsing peers are used.
- **Consensus**: It is generally a mechanism that is utilized for validating a block before its addition to the chain. In Fabric, there exist two different kinds of consensus mechanisms: Voting and Lottery. In Fabric, there are three phases, including Validation, Ordering, and Endorsement.

- **Chaincode**: Actually, it is a computer program or a smart contract that can be developed in many languages, including GO and JavaScript, and it tends to run on peers. It is considered a self-executable application in which different guidelines and terms of an agreement between a seller and buyer are written into code lines. Therefore, a chain code is utilized to implement business logic in Hyperledger that manages communication between the ledger and the applications.
- **MSP**: It is also recognized as Membership Service Providers, and clients need to possess authorized credentials for joining a network. Clients need MSPs for accessing credentials as they are a semi-abstract element or component.

For addressing the below challenges and issues, our system has been implemented with a cloud database and Hyperledger Fabric that runs on four instances of cloud VM:

- Ensuring transparency and traceability in the system of consent management that raise the level of user trust.
- On Blockchain, not storing and including personal information, which does not enable data erasure. Thus, off-chain storage has been considered on Cloud rather than personal information storage on Blockchain. Additionally, hash references are not stored.
- For the protection of data of patients and respecting the privacy of the user, we have focused on the system's privacy aspects by eliminating the data completely from the database of the cloud upon the revoke request.
- The utilization of advanced Hyperledger Fabric versions and react frontend for communicating with the network rather than the depreciated Composer.

The early implementation of the system utilizing Ethereum and Hyperledger (on a virtual machine) is discussed in the following sections before moving on to the final cloud implementation.

4.2 Early Ethereum-based Implementation

AWS Blockchain templates are used to create and deploy the Ethereum blockchain network. These templates use a cloud formation stack on AWS to create a Blockchain network. The templates on AWS are like Infrastructure as Code (IaC). Using the template, we can either connect to the main Ethereum network, which is public, or to a private Ethereum network. After the network setup, Metamask is used for connecting to the Ethereum based networks and it allows running the DApps from the browser. A private network was created and used for the system.

4.3 Early Hyperledger-based Implementation

The prototype was implemented locally on a virtual machine initially. The following requisites are installed on the system; Curl, NodeJs, Git, Python, Go Language, Docker CE, Docker Compose, and library tools. Once the prerequisites were installed, fabric samples are downloaded using curl. The environment variables are updated to ensure the working of GoLang.

After installing the prerequisites and downloading the fabric, the test network ((2 peers, orderer, 3 CA, 2 CouchDB) is initialized. Channel is created after starting the test network. The chaincode for writing data onto the network is deployed on the channel. Fig 4.1 shows the steps involved in deploying a chaincode to the network.

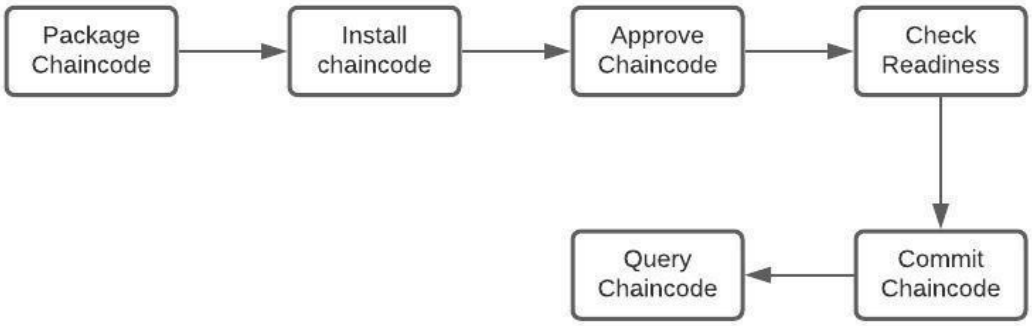


Figure 4.1 Chaincode Lifecycle

To interact with the chaincode through the front-end application, few steps should be completed before such as enrolling the admin and registering the application user [28]. These interactions are between the CA and the application. Once the admin user and application user are enrolled, the credentials are stored in a wallet. Suppose the credentials are present and have the appropriate authorization attributes associated with them. In that case, the user of the application will be able to access chaincode functions after obtaining references to the channel name and contract name from the sample application [28]. This is the backend node application that is used by a backend server to interact with the network.

With the backend running, react applications for organizations and users were built to interact with the network. The users can write their consent details on the network using the front-end. The organizations can check the user consent details from the network and request the data from the healthcare admin accordingly.

The major limitation of implementing the system locally is that the throughput of the system is very low. If there is an issue with the virtual machine or the laptop, the

application will be affected. To increase the throughput of the system and to avoid the single point of failure, the application is implemented on the Cloud. Also, the organizations are hosted on multiple EC2 instances instead of using a single instance. This will increase the TPS and will be decentralized, with each organization having its own virtual machine.

4.4 Cloud-based Implementation

Four virtual machines have been created for the implementation of the prototype. The instances are of Ubuntu 18.04 with the following specifications, including 50GB storage, 2 CPUs, and 4GB of Memory. Similar to local implementation, all the prerequisites were installed. During the installation of prerequisites, few issues were experienced because some versions of the software were not compatible with each other.

Accordingly, few environmental variables have been added and updated to accommodate GoLang's smooth working. As mentioned earlier, the key reason for implementing Hyperledger Fabric on several VM or virtual machines is to achieve better system performance in terms of transaction throughput and response time because organizations must check the user details for requesting information from the admin. Therefore, better results were achieved through the implementation of HF on several VMs. Also, having multiple organizations located on different EC-2 instances makes it more distributed.

Crypto materials have been prepared initially for three organizations and one orderer organization. A Central Authority (CA), two ledgers, and two peers are included in each organization. In combination, there are three CAs, six ledgers, and six peers. For the orderer organization, there is a CA, and there are three orderers. With the generation of certificates

for all participants, the MSP of the organization is created. The organization's MSP is vital in the development of genesis block. It is the first block that does not really include any form of transaction data in it. However, it involves the MSP ids of the three specific organizations and their certificates. Channel consortium and name are included in the channel configuration transaction that will be utilized in the channel. The development of channel tx and genesis block is depicted in Fig 4.2.

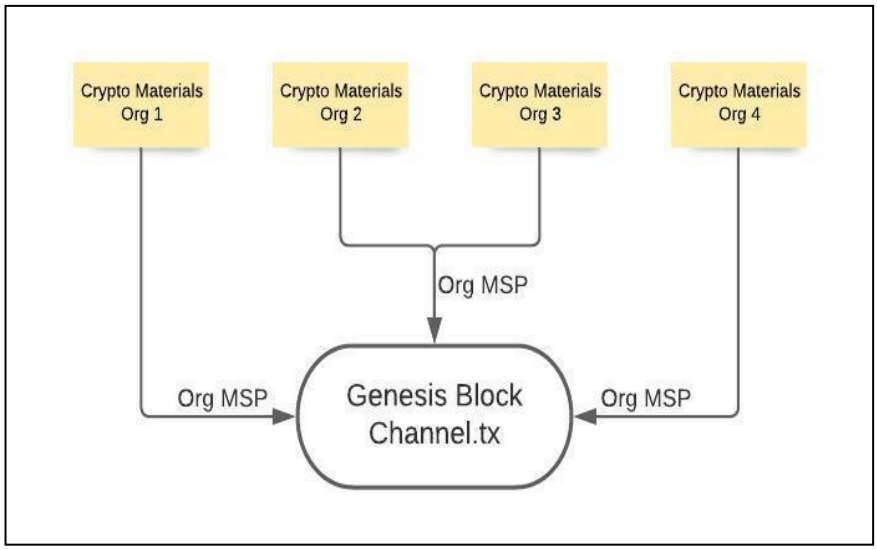


Figure 4.2 Genesis Block Development

All the certificates are generated in a virtual machine, and they are later moved to their respective machines with the use of SFTP. A docker swarm network was also created to ensure communication among them. In Figure 4.3, the addition of organizations to the channel is illustrated.

After the installation of Fabric on all the machines, the focus was on the development of chaincode to be installed and applied on peers. All the latest versions of Fabric have an

advanced approach to the deployment and development of chaincode. The chaincode was packaged, installed, and committed by the peers as per the latest chaincode lifecycle. It should be noted that Chaincode lifecycle refers to the whole process, which is introduced explicitly from the version 2.0 Fabric. A chaincode has been developed that can record the given information; name, email, consent (if it is partial or complete), and organization (organization name to which the user gave consent). Two peers are included in an organization, and one is an endorsing peer. On the endorsing peer, the chaincode is implemented. The chaincode is installed successfully on all three organizations. On the Blockchain, data was recorded with the use of Hyperledger Fabric Node SDK.

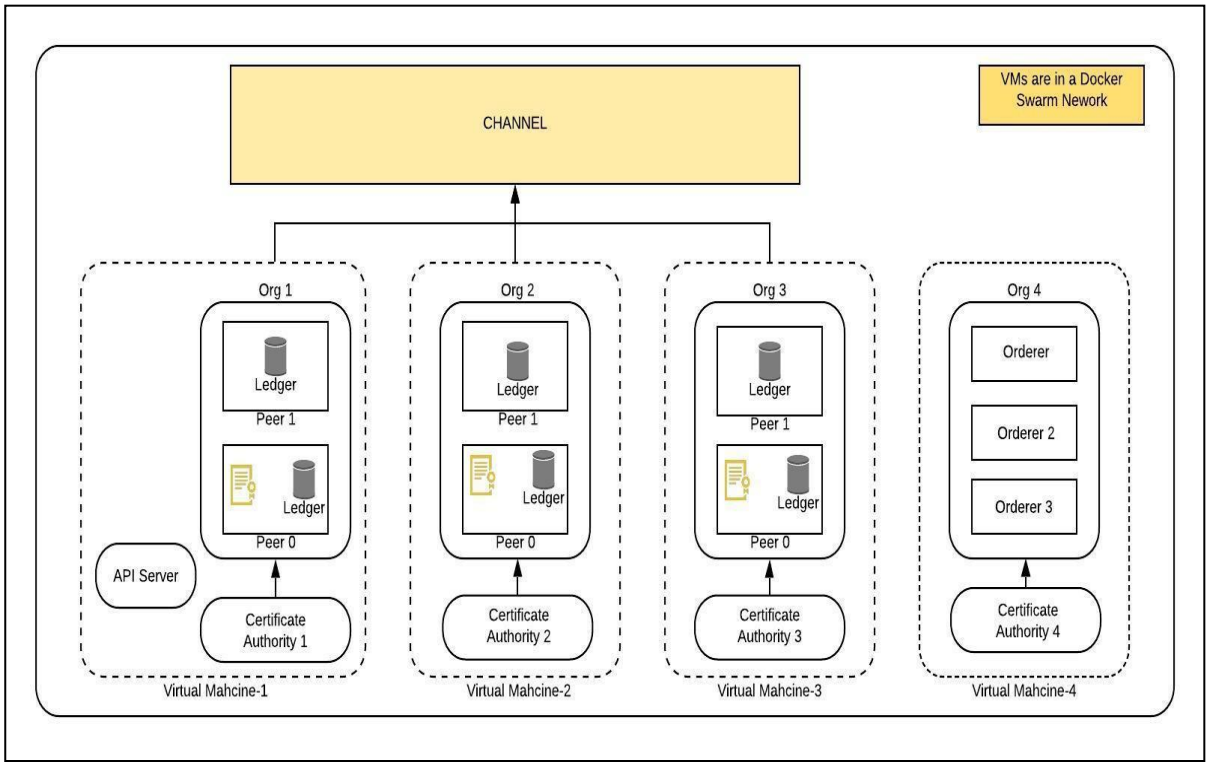


Figure 4.3 Setup of Fabric on Virtual Machines

React has been used for developing the frontend that serves to invoke the functions of chaincode with the use of API endpoints, as illustrated in Fig. 4.4. The users, once logged

in, can store their data or information on the database of the cloud from the react applications directly and write details about consent on the blockchain. In addition to it, they can see if there are any messages or notifications from the organizations that request any type of additional information. Organizations joining the network can access functionalities of chaincode to see the details of the user's consent from the network while requesting full access from the admin if the users have provided enough approvals. It should be noted that the cloud database is an AWS S3 storage bucket that can store the files uploaded by the users. Since it is a private bucket, it blocks public access. Additionally, necessary steps have been taken to keep it safe and secure. In the process of implementation, the main challenges are:

- Understanding blockchain concepts for designing a proper framework for private data management.
- Implementation of different software that will assist in Fabric installation.
- Insufficient and complicated information is present about Fabric SDK usage.
- Managing the development of our system’s front end, such as CORS or Cross-Origin Resource Sharing.

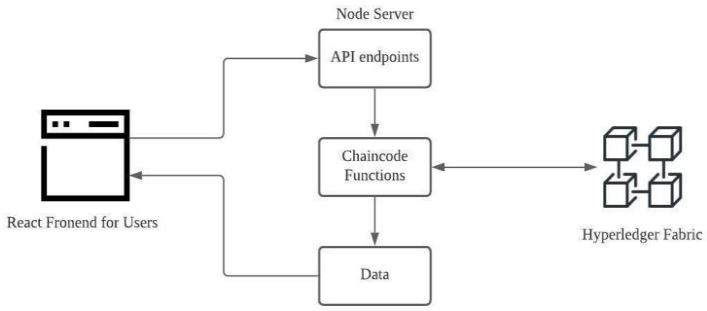


Figure 4.4 Interaction between the frontend and Blockchain network

The following section details the system's implementation using the use case of healthcare.

4.5 Use case- Healthcare Implementation

The term "health research," sometimes also called "medical research" or "clinical research," refers to research that is performed to learn more about human health [23]. Health research is extremely important since it tries to enhance disease prevention and treatment through scientific discovery. Medical judgments were made largely on clinicians' best estimations and expertise in the absence of health research, which resulted in many instances being inaccurate [23]. The guesswork may be reduced with the introduction of health research because drugs are now thoroughly evaluated and confirmed to be effective before being prescribed. In addition, data from 9000 breast cancer patients was gathered, and the information gathered finally led to the invention of Herceptin, which is now available (used for treating breast and stomach cancer) [24]. As a result, without collecting and analyzing medical data from volunteers or patients, health research would be impossible to conduct.

The following sections will demonstrate the working of the application. We have shown both the user’s and organization’s level front end that will interact with the Blockchain network.

4.5.1 Adding Consent to the Network

Consent can be added to the Blockchain using the frontend as mentioned before. The details such as the ID, Name, Email, Consent details, and Organization can be collected and posted on the network as shown in Fig. 4.5.

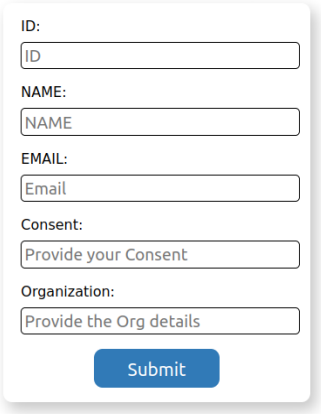


Figure 4.5 Consent Details are added to the network.

4.5.2 Reading Consent from the Network

Once the consent details are written, organizations can now check the network for consent information of the users, as shown in Fig. 4.6. If the user has given enough approvals, they can request data from the healthcare admin. If they need additional consent, they can request the user from the front end.

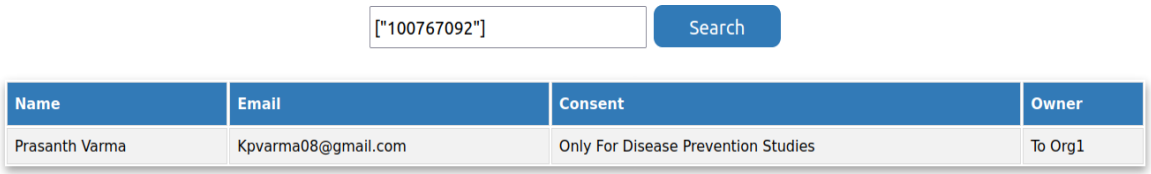


Figure 4.6 Reading consent details from the network

If they have required consent, they can request the healthcare admin to share the files. The admin can validate and share the files accordingly.

4.5.3 Modifying Consent on the Network

Users can always modify the consent Details on the network. They can do it by making another transaction to update the consent information on the network. In Fig. 4.7, if a user updated his/her consent to ‘None’ in the organization field, it implies a revoke

request. A user can provide partial consent initially and change it to complete consent later and vice versa.

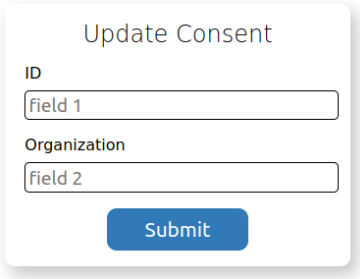


Figure 4.7 Modifying consent on the network.

Once the details are updated, the latest consent information can be found from the network.

4.5.4 Access Log for Users

Users can check the blockchain network to see the data sharing logs between the organizations. This will help the user to increase the trust level in the system. They are also displaying the history information as shown in Fig. 4.8 will bring the required traceability and transparency to the system.

Figure 4.8 History log of a user's consent information

The user initially provided access to Org1 and mentioned that it should be used for Disease Prevention Studies alone. Later, the consent details are changed to 'None' (indicating a revoke request).

4.6 Summary

The development of a prototype based on the proposed design from the previous chapter was discussed in detail in this chapter. The working of relevant Hyperledger Fabric's components has been discussed. In addition, the chapter examined the advantages of implementing the prototype on the cloud when compared to implementing it on a local virtual machine. The evaluation results of the prototype implementation will be discussed in more detail in the following chapter.

Chapter 5.

Evaluation Results

This chapter assesses the extent to which this thesis accomplishes the objective mentioned at the outset of this thesis. The conclusion section is divided into two key sections. The first section contains early results of system implementation utilizing Ethereum and Hyperledger. The next section includes the system implementation on the cloud using Hyperledger. Local implementation using private blockchain (Hyperledger) is carried out in a single virtual machine, whilst cloud implementation is carried out on four EC-2 instances.

5.1 Early Experimental Implementation Results

Initially, the system was designed and developed with a public blockchain (Ethereum). Following that, it was tested on a single virtual machine utilizing a private blockchain (Hyperledger). The results of Ethereum's implementation results are presented in the next section.

5.1.1 Ethereum-based Implementation

A smart contract was developed for the public blockchain (Ethereum) system to allow data transfer between the admin and a third party. The algorithm of the smart contract is shown in Figure 5.1. The key reason for incorporating a price into the smart contract is that the Ether required for the smart contract to succeed can be used as an incentive for customers, benefiting both the organization and the customers.

Algorithm: Transfer of Data

1. **Input**: address thirdparty,
   address admin,
   Amount of Ether
2. **State:** 0. AWAITING_PAYMENT,
   1. AWAITING_TRANSFER,
   2. TRANSFER_COMPLETE
3. **State** → CurrState
4. **Modifier** → onlythirdparty
5. **Constructor:** _admin, _thirdparty
6. **Function 1** : (Payable)
   Require → AWAITING_PAYMENT
   Function is Payable,
   Then currState→ AWAITING_TRANSFER
7. **Function 2:**
   Modifier → onlythirdparty
   Require → AWAITING_TRANSFER
   admin→send(this.balance)
   Change of ownership
   currState → TRANSFER_COMPLETE
   Event: Trigger an event to inform other nodes
8. **END**

Figure 5.1 Smart Contract Algorithm

The smart contract was deployed in a test network. It is triggered when a third-party requests data. The average transaction cost for deploying this contract is 0.000312216 Ether. The average transaction fee for invoking the first function is 0.000028952 Ether. The amount of Ether used up for calling the second function, which is invoked after the successful transfer of data is 0.000037976 Ether.

5.1.2 Hyperledger-based Implementation

The configuration of the local virtual machine is given in Table 5.1. Peers for each organization are installed on separate ports within the same virtual machine in this configuration.

Table 5.1 System Configuration of local virtual machine setup

Configuration	Value
Instance Type	Ubuntu 20.04
No. of processors	4
Memory	6.1 GB
Storage	50 GB

J-meter was utilized to analyse the network's performance. We have taken 100 threads for the evaluation with a ramp-up speed of one second. Ramp-up speed is the rate at which new concurrent users attempt to access the system during a load test [41]. The experiment is designed to determine the system's throughput using a J-meter load test. The read throughput (the time required to retrieve data from the network) is evaluated. As mentioned previously, this experiment uses 100 threads with a one-second ramp-up period. We repeated this test four times to ensure that there are no significant differences in the final output. Fig. 5.1 depicts the combined results of all four experiments. The results are distinguished from one another using a different colour.

The y axis represents the number of transactions per second, while the x-axis represents the number of active threads. As illustrated in Fig. 5.2, there was always an average of 20 unsuccessful transactions per 100 users. Thus, the success rate of local implementation is approximately 80%. In none of the experiments conducted on a local machine, we obtained complete successful transactions.

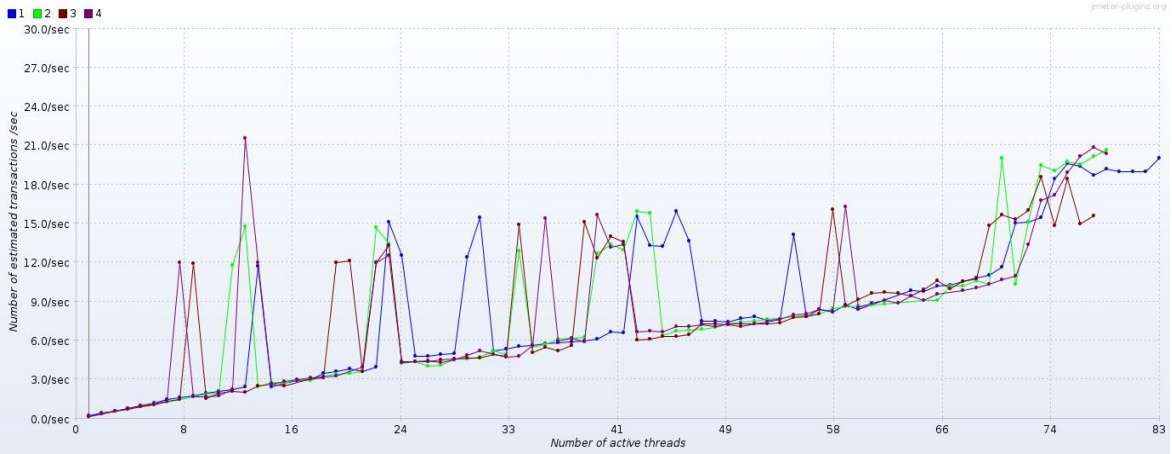


Figure 5.2 Combined results of the throughput in local machine

The average throughput (TPS) is 9.5, with a success rate of roughly 80%. This is primarily because the system was implemented on a single machine. In general, the success rate is meager when compared to cloud implementation. Overall, the read throughput is significantly less.

Another critical measure is the response time of the system. It is also evaluated using a J-meter load test with 100 users. The combined response time results of all experiments are shown in Fig. 5.3.

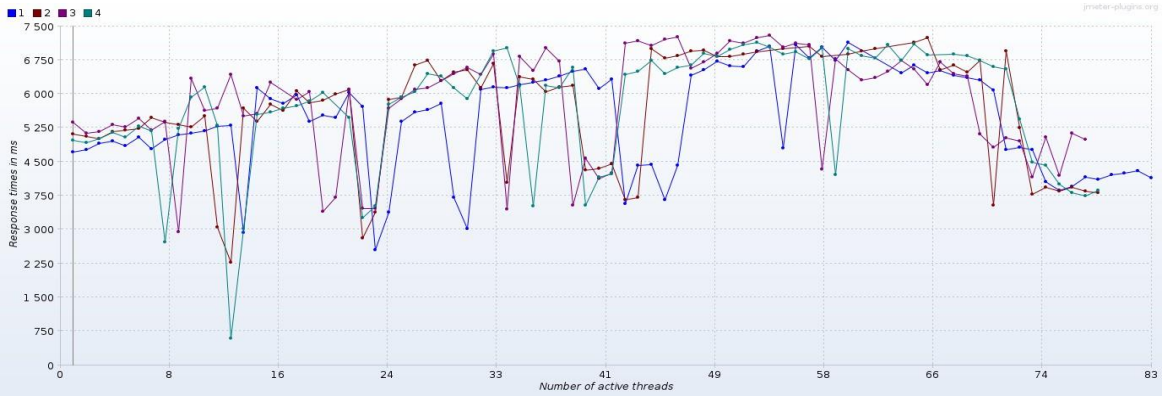


Figure 5.3 Combined results of the response time in local machine

The average response time for all results is approximately 5302.4 milliseconds (almost 5.3 seconds) for approximately 80 completed transactions. As a result of the unsuccessful transactions, it is concluded that the network was not stable on the local machine.

5.2 Cloud-based Implementation

We created a permissioned network to which only specified organizations can be added. All virtual machines were configured identically and are in a private Virtual Private Cloud (VPC) on AWS (Amazon Web Services). To test our system's performance, we ran experiments on it using Hyperledger Caliper [29]. The details of our system configuration are included below in Table 5.2.

Table 5.2 System Configuration of Cloud setup

Configuration	Value
Instance Type	t2. medium
Amazon Machine Image (AMI)	Ubuntu 18.04
No: of CPUS	2
Memory	4 GB
Storage	50 GB

Caliper is used to evaluate the system's performance, and its details are discussed in the following sections.

5.2.1 Hyperledger Caliper

Hyperledger Caliper is a blockchain performance evaluation tool that enables users to compare the performance of a blockchain implementation to a collection of predefined

use cases [29]. Hyperledger Caliper generates reports that include a variety of performance indicators for use with the Hyperledger Besu, Hyperledger Burrow, Ethereum, Hyperledger Fabric, FISCO BCOS, Hyperledger Iroha, and Hyperledger Sawtooth blockchain systems [29]. Caliper now supports the following fabric SDK versions: 1.1.0 (1.1), 1.4.11 (1.4, latest), and 2.1.0. (2.1, latest-v2). There are four critical measures for evaluating a system's performance, and they are as follows:

- **Read Latency:** Read latency is the time interval between submitting a read request and receiving a response [30].
- **Transaction Throughput:** The rate at which valid transactions are committed by the blockchain SUT (System Under Test) in each period is known as transaction throughput [30]. This is not the rate at a single node but rather the rate over the entire SUT, i.e., committed at all network nodes. At a network size of a certain magnitude, this rate is given in transactions per second (TPS).
- **Transaction Latency:** Transaction Latency is a network-wide view of the amount of time taken for a transaction's effect to be usable across the network [30]. The time covered by the measurement is from the point at which the request is submitted to the point at which the result is generally available on the network.

To establish Caliper on our system, we gathered all the necessary crypto materials in the first virtual machine (vm1). To ensure the correct operation of the Caliper, node and npm were updated to their latest versions. Caliper was used in conjunction with Docker, and the following steps were taken to launch the container:

- Decided on an image version. Version 0.4.1 of the Caliper image.

- Mount a container directory to your local working directory.
- Set the binding and run parameters that are required as shown in the below Fig. 5.4.

The Fabric version that is used in our implementation is 2.1.0.

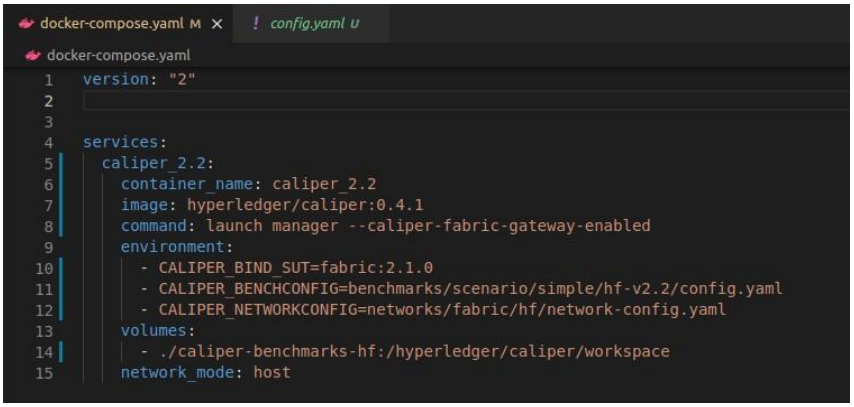


Figure 5.4 Setup for Hyperledger Caliper

The network-config file is a YAML file that is used to create the configuration file. The network-config file has been composed to meet our configuration. The network configuration shown in Fig. 5.5 is a snippet of the network configuration used to connect to the Caliper.

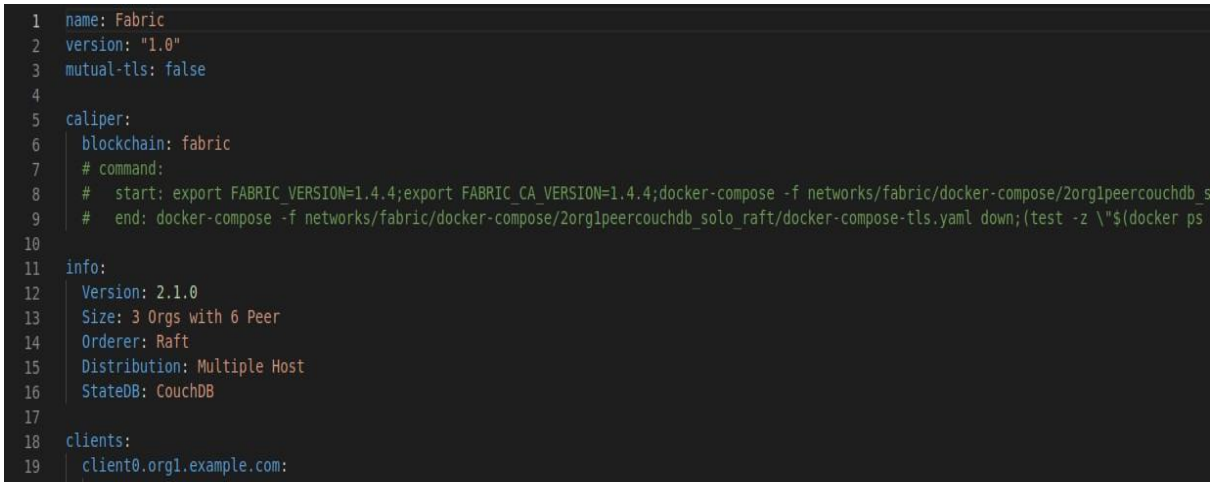
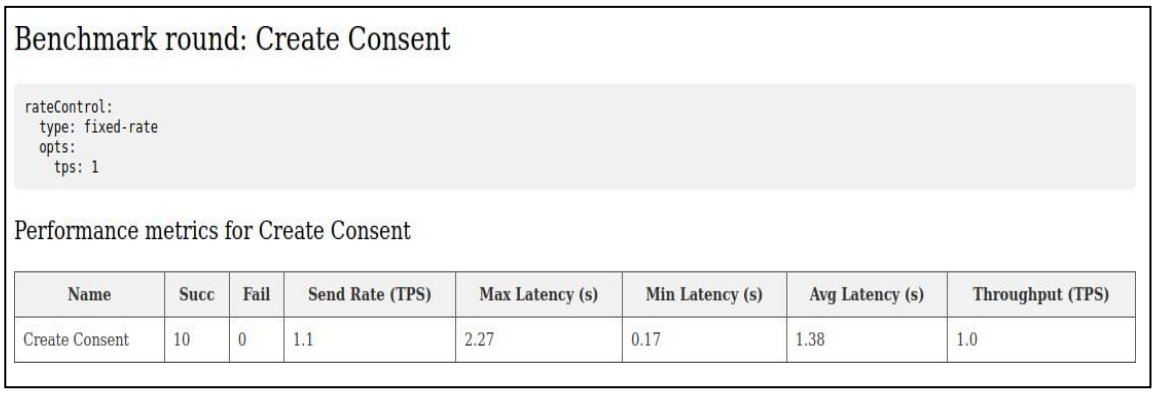


Figure 5.5 Network-config file snippet

Once the configuration is completed, the docker container is started. To begin, two test cases were created: one for reading data and another for reading/writing data to the network, both of which were fixed rates. The following section contains Caliper's results.

5.2.2 Experiment-1

In this experiment, the throughput and latency of the system are evaluated. We began by experimenting with a minimal number of transactions with a send rate of 1 TPS. We measured the throughput and latency of the system by executing ten transactions. We achieved a throughput of 1 transaction per second with ten transactions (TPS). The transaction processing speed (TPS) was 1.1 transactions per second. The most considerable latency was 2.27 seconds, and the minimum was 0.17 seconds. The average latency was approximately 1.38 seconds. The experiment's outcome is depicted in Fig. 5.6.



Benchmark round: Create Consent

```
rateControl:
  type: fixed-rate
  opts:
    tps: 1
```

Performance metrics for Create Consent

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Create Consent	10	0	1.1	2.27	0.17	1.38	1.0

Figure 5.6 Caliper results (Create Consent) of experiment 1

Fig. 5.7 illustrates the findings from the evaluation of Read consent measures. The average latency was 0.02s, while the maximum and minimum values were respectively 0.02 and 0.01s. The overall throughput of the Read consent experiment is 1.1TPS.

Consequently, we began increasing the transaction volume for analysis purposes.

Benchmark round: Read Consent

Test description for the query performance of the deployed contract.

```
rateControl:
  type: fixed-rate
  opts:
    tps: 1
```

Performance metrics for Read Consent

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Read Consent	10	0	1.1	0.02	0.01	0.01	1.1

Figure 5.7 Caliper results (Read Consent) of experiment 1

5.2.3 Experiment-2

We increased the number of transactions for the subsequent experiment to 1000 and 2000 for the Create and Read consent experiments, respectively, with a send rate of 40 and 220 TPS. This increased the throughput of the system. The results for this experiment are shown in Fig. 5.8.

Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
999	1	40.0	13.87	0.49	9.89	29.5
2000	0	213.1	12.54	0.57	7.69	133.2

Figure 5.8 Caliper results for experiment 2

The highest latency in terms of results is 13.87 seconds for the Create consent experiment and 12.54 seconds for the Read consent experiment. The most negligible latency is 0.49s, and the maximum delay is 0.57s, respectively. The throughput for Create consent is 29.5 transactions per second, whereas the throughput for Read consent is 133.2 transactions per second.

The average latency was 9.89 seconds for Create consent and 7.69 seconds for Read consent, as shown in Fig. 5.4. With 2000 transactions, throughput was 133.2 TPS with an average latency of 7.69s. The maximum latency was 12.54s, whereas the minimum latency was 0.57s.

5.2.4 Experiment-3

To determine the difference in throughput and latency, the send rate was increased to 100 and 350 TPS. The number of transactions seeking Read consent has been increased from 2000 to 2500, while the number of transactions requesting Create consent has remained constant. Fig. 5.9 illustrates the outcomes of Create consent, whereas Fig. 5.10 illustrates the results of Read consent.

Benchmark round: Create Consent

```
rateControl:
  type: fixed-rate
  opts:
    tps: 100
```

Performance metrics for Create Consent

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Create Consent	1000	0	98.2	34.35	1.97	18.94	27.1

Figure 5.9 Caliper results (Create Consent) of experiment 3

Benchmark round: Read Consent

Test description for the query performance of the deployed contract.

```
rateControl:
  type: fixed-rate
  opts:
    tps: 500
```

Performance metrics for Read Consent

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Read Consent	2500	0	329.6	16.11	7.32	11.41	151.2

Figure 5.10 Caliper results (Read Consent) of experiment 3

The increase in transmit rate resulted in an increase in Read consent transaction throughput from 133.2 TPS to 151.2 TPS. The maximum Read consent latency is 16.11 seconds, and the minimum generated consent delay is 7.32 seconds. On average, the delay is 7.32 seconds. The average read consent latency rose by 3.72 seconds.

In general, the network was able to handle a higher volume of requests without experiencing any performance concerns. We noticed only one failed transaction during the evaluations, indicating that the workload is being divided among the organizations to maintain the network's stability.

When the system was deployed on a local machine, it had a low throughput and a success rate of only 80 percent. However, when implemented on the cloud, the system achieved a higher transaction success rate. Additionally, it produced a higher throughput for a more significant number of transactions than a locally implemented solution. This is due to the deployment of multiple hosts, which resulted in increased network stability compared to the system implementation on the single virtual machine. Even if the system is configured locally using multiple virtual machines, issues may still arise. For example, problems with hardware or software will serve as a single point of failure. In conclusion, a cloud-based system will be far more stable than a local virtual machine-based one.

5.3 Comparison with Related Work

The comparison of the presented system in this thesis with other systems is shown in Table 5.3.

Table 5.3 Comparison between systems

Papers	Implementation / Performance Evaluation	Comments
[17]	Prototype implementation details are given.	Does not provide implementation details.
[18]	Implementation with few performance analyses.	Does not cover the details of data sharing.
[19]	Implemented with Hyperledger Composer.	Does not have performance analysis. The composer is now deprecated.
[20]	The prototype is mentioned in this paper.	Does not provide implementation details
[21]	Uses multichain to implement private data management.	Performance evaluation of the system is not reported.
[22]	Implementation with the performance analysis is covered in this paper.	The composer is used in the system, which is now deprecated. Discusses only a few metrics of Fabric.
Proposed System	We have included the implementation details and performance analysis.	Used the latest version of Fabric. Response time and transaction throughput for fetching the details from the network have been calculated and reported.

As shown in Table 5.3, the research papers [17] and [20] address only prototypes in the healthcare domain. The paper [18] does not examine the specifics of data exchange between users and other departments of the healthcare department. Hyperledger Fabric is used to implement the prototype. Additional evaluation details, such as latency and throughput, which are also Fabric metrics, should have been included [30]. The prototype discussed in [19] involves the use of Hyperledger Composer which is deprecated. It also does not involve any evaluation of the implemented prototype. Deprecated Hyperledger Composer is also used in [22]. In [22], the response time for retrieving patient data is 5683ms, and no other metrics are discussed. In our prototype, we have used Caliper to

measure the system. Additionally, we demonstrated metrics with varying quantities of data records.

5.4 Summary

Early implementation details with public and private blockchain are presented in this chapter. The average gas costs associated with invoking smart contract functions are presented. Initially, the system using Hyperledger was configured on a single virtual machine. When examining the system's performance, it had an average success rate of 80% for successful transactions. So, the system was implemented on the cloud. In order to improve transaction success rates, multiple instances were used for implementation rather than a single virtual machine. Because the cloud-based approach used a higher number of instances, the results were improved. The result was a transaction success rate of 99 percent.

Chapter 6.

Conclusion and Future Work

We examined the design and implementation of a consent management system for private data in this thesis. We discussed the implementation details of our proposed system from the perspective of a healthcare case study. Our proposed system is intended for usage by individuals and organizations. In our use case, patients can offer consent details and share their medical files via the Blockchain network, while organizations can request data from users for medical data research. We created this system with the security of sensitive data in mind. Our technology leverages Blockchain's key advantages, such as immutability, to provide users with traceability and transparency. Additionally, the technology improves the present consent collection process with Blockchain by informing the user of the purpose for data collection. Additionally, we covered the ways for sharing sensitive data that are included in the process of sharing via IPFS and Amazon S3. We reviewed the AWS access policy that will be used to ensure that the data is not accessible after the agreed-upon period.

6.1 Conclusion

The main objective of this thesis is to explore blockchain technology and experiment with it for enhanced data security and personal information management for individuals, corporations, government entities, and public institutions. To achieve this, we have addressed the following research questions critical in designing and developing a transparent, traceable, and immutable CMS.

RQ1: What is the major issue with the current Consent Management System?

In the second section, 2.2.1, the issues of the present CMS are detailed using a use case (healthcare). The primary concerns are around the CMS's lack of transparency and traceability. Additionally, consent is frequently obtained in mass rather than for specific reasons as required. Consent must be expressed explicitly. Additionally, it should be easy to obtain a summary of consent history and revoke consent as quickly as possible to grant consent.

RQ2: How do the features of Blockchain contribute and benefit consent management?

Blockchain features such as immutability are mentioned in the second chapter (section 2.3.2). The Blockchain is a distributed ledger that stores transactions in an append-only fashion. All data is shared in its entirety among a large group of nodes. The Blockchain data structure combines data into immutable blocks that are deterministically verifiable. The Blockchain provides an excellent framework for implementing a consent management system. The reasons being: Individuals have been denied transparency on the consent process. To be effective, a consent platform must earn the user's trust. Transparency, trustworthiness, and security are all provided by the Blockchain.

RQ3: What are the limitations in the current blockchain-based consent management solutions, and how are they addressed?

The second chapter (section 2.5) discusses the shortcomings of the current system. The existing system's primary drawbacks are using cryptocurrencies such as Ether to conduct transactions in CMS, privacy concerns, and the use of deprecated Hyperledger versions. Also, many prototypes discuss storing the hash references of PII on Blockchain. This should be avoided, as hash references may be regarded as personal information as

well. The HF was installed on four virtual machines and connected via a docker swarm network. We feel that our technology would benefit users by allowing them to audit their data. Our system will provide users a greater degree of control over their data. Additionally, it assures that data is used legitimately and sparingly by research institutions, e.g., patient data is deleted after specified periods. We have addressed all the issues in the current systems and designed a prototype for CMS. We have also given the evaluation details of the system implemented on the local machine and the cloud.

6.2 Future Work

The implementation has shown that Blockchain can be used for developing a CMS. The features of Blockchain provided users with a new level of trust. However, there are few limitations of the proposed prototype and are as follows.

- The system has no precautions in place to ensure the integrity of the data collected from users. It enables users to upload data to the database without validating its accuracy.
- Using an ID, consent can be updated on the network. ID is generated at the time of service registration and functions similarly to a private key. Users will have difficulties updating their consent in the event of ID loss.
- The cost of the production-ready application will require a certain amount compared to traditional systems as it highly depends on the resources allocated, such as processing units, memory, storage, etc. This will also affect the network speed of Blockchain in reading the data from or writing the data to.

Blockchain can also be used for data validation. We intend to incorporate this feature into our system by making a few minor design changes. We intend to address the second challenge by utilizing the Attribute-based access control (ABAC) technique to implement the smart contract. This will also increase the users' trust. Finally, we intend to deploy the application in a practical situation that benefits both users and organizations.

Bibliography

1. “Global digital population as of January 2021”. Available online: https://www.statista.com/statistics/617136/digital-population-worldwide/. [Accessed: 04-June-2021].
2. Isaak, J.; Hanna, M.J. “User data privacy: Facebook, Cambridge Analytica, and privacy protection”. Computer 2018, 51, 56–59.
3. Kakarlapudi, P. V., & Mahmoud, Q. H. (2021, February). A Systematic Review of Blockchain for Consent Management. *In Healthcare* (Vol. 9, No. 2, p. 137). Multidisciplinary Digital Publishing Institute.
4. “Art. 17 GDPR Right to erasure (‘right to be forgotten’)”. Available online: https://gdpr-info.eu/art-17-gdpr/. [Accessed: 04-June-2021].
5. Thakkar, P., Nathan, S., & Viswanathan, B. (2018, September). Performance benchmarking and optimizing hyperledger fabric blockchain platform. *In 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)* (pp. 264-276). IEEE.
6. Esposito, C., De Santis, A., Tortora, G., Chang, H., & Choo, K. K. R. (2018). Blockchain: A panacea for healthcare cloud-based data security and privacy? *IEEE Cloud Computing*, 5(1), 31-37.
7. “What is Data Management?”. Available online: https://www.ngdata.com/what-is-data-management/. [Accessed: 06-June-2021].
8. “5 things you need to know about Data Privacy.” Available online: https://dataprivacymanager.net/5-things-you-need-to-know-about-data-privacy/. [Accessed: 06-June-2021].
9. Haber, S., & Stornetta, W. S. (1990, August). How to time-stamp a digital document. *In Conference on the Theory and Application of Cryptography* (pp. 437-455). Springer, Berlin, Heidelberg.
10. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 21260.

11. “What is blockchain history.” Available online: https://www.icaew.com/technical/technology/blockchain/blockchain-articles/what-is-blockchain/history. [Accessed: 06- June-2021].
12. “The Mysterious Disappearance of Satoshi Nakamoto, Founder & Creator of Bitcoin”. Available online: https://www.huffpost.com/entry/the-mysterious-disappeara_2_b_7217206. [Accessed: 06- June-2021].
13. Szabo, N. Formalizing and securing relationships on public networks. First Monday 1997, 2, 9.
14. “Blockchain Consensus Algorithm: Proof of Work (POW)”. Available online: https://www.cisin.com/coffee-break/technology/blockchain-consensus-algorithm-proof-of-work-pow.html. [Accessed: 06-Jan-2021].
15. “Proof of stake”. Available online: https://www.investopedia.com/terms/p/proof-stake-pos.asp. [Accessed: 06-June-2021].
16. “Linux Foundation Unites Industry Leaders to Advance Blockchain Technology”. Available online: https://web.archive.org/web/20170717193806/https://www.linuxfoundation.org/news-media/announcements/2015/12/linux-foundation-unites-industry-leaders-advance-blockchain. [Accessed: 06-June-2021].
17. Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016, August). Medrec: Using blockchain for medical data access and permission management. *In 2016 2nd international conference on open and big data (OBD)* (pp. 25-30). IEEE.
18. Liang, X., Zhao, J., Shetty, S., Liu, J., & Li, D. (2017, October). Integrating blockchain for data sharing and collaboration in mobile healthcare applications. *In 2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)* (pp. 1-5). IEEE.
19. Rouhani, S., Butterworth, L., Simmons, A. D., Humphery, D. G., & Deters, R. (2018, July). MediChain TM: a secure decentralized medical data asset management system. *In 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and*

Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 1533-1538). IEEE.

20. Swetha, M. S., Pushpa, S. K., Muneshwara, M. S., & Manjunath, T. N. (2020, December). Blockchain enabled secure healthcare Systems. *In 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)* (pp. 1-6). IEEE.
21. Al Asad, N., Elahi, M. T., Al Hasan, A., & Yousuf, M. A. (2020, November). Permission-Based Blockchain with Proof of Authority for Secured Healthcare Data Sharing. *In 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)* (pp. 35-40). IEEE.
22. Rajput, A. R., Li, Q., Ahvanooey, M. T., & Masood, I. (2019). EACMS: Emergency access control management system for personal health record based on blockchain. *IEEE Access*, 7, 84304-84317.
23. "Particpating inHealth Research Studeis". [Online] Available: https://guides.library.harvard.edu/c.php?g=389023&p=2639499 [Accessed: 08-June-2021].
24. Gostin, L. O., Levit, L. A., & Nass, S. J. (Eds.). (2009). Beyond the HIPAA privacy rule: enhancing privacy, improving health through research.
25. "Privacy". Available online: https://marcomm.mccarthy.ca/pubs/share2.html. [Accessed: 08-June-2021].
26. "Total amount of global healthcare data generated in 2013 and a projection for 2020". [Online] Available: https://www.statista.com/statistics/1037970/global-healthcare-data-volume/. [Accessed: 10-June-2021].
27. Dhillon, V., Metcalf, D., & Hooper, M. (2017). The hyperledger project. *In Blockchain enabled applications* (pp. 139-149). Apress, Berkeley, CA.
28. "Hyperledger – Write First App." Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/write_first_app.html. [Accessed: 10-June-2021].
29. "Hyperledger Caliper." Available online: https://hyperledger.github.io/caliper/v0.3.2/fabric-config/. [Accessed: 12-June-2021].

30. “Hyperledger Blockchain Performance Metrics.” Available online: https://www.hyperledger.org/wp-content/uploads/2018/10/HL_Whitepaper_Metrics_PDFVersion.pdf. [Accessed: 12-June-2021].
31. “Consent Management.” Available online: https://www.gartner.com/en/information-technology/glossary/consent-management (accessed on 12 June 2020). [Accessed: 12-June-2021].
32. Conoscenti, M.; Vetro, A.; De Martin, J.C. Blockchain for the Internet of Things: A systematic literature review. *In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, Agadir, Morocco, 29 November–2 December 2016.
33. Cha, S.C.; Chen, J.F.; Su, C.; Yeh, K.H. A blockchain-connected gateway for BLE-based devices in the Internet of Things. *IEEE Access* 2018, 6, 24639–24649.
34. Sabrina, F. A Novel Entitlement-based Blockchain-enabled Security Architecture for IoT. *In Proceedings of the 2019 29th International Telecommunication Networks and Applications Conference (ITNAC),* Auckland, New Zealand, 27–29 November 2019.
35. Monrat, A.A.; Schelén, O.; Andersson, K. A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access* 2019, 7, 117134–117151.
36. Chowdhury, M.J.M.; Colman, A.; Kabir, M.A.; Han, J.; Sarda, P. Blockchain as a notarization service for data sharing with personal data store. *In Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, New York, NY, USA, 1–3 August 2018.
37. Doku, R.; Rawat, D. Pledge: A private ledger based decentralized data sharing framework. *In Proceedings of the 2019 Spring Simulation Conference (SpringSim)*, Tucson, AZ, USA, 29 April–2 May 2019.
38. Alessi, M.; Camillo, A.; Giangreco, E.; Matera, M.; Pino, S.; Storelli, D. Make users own their data: A decentralized personal data store prototype based on ethereum and

ipfs. *In Proceedings of the 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech)*, Split, Croatia, 26–29 June 2018.

39. Topart, L.; Genestier, P.; Picaud, Y. Blockchain brings confidence to facilitate the flow of data in the agricultural field. *In Proceedings of the 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, Paris, France, 28–30 September 2020.
40. "What is chaincode?". Available online: https://fabrictestdocs.readthedocs.io/en/latest/chaincode.html. [Accessed: 15-June-2021].
41. "The Importance of Ramp Up and Ramp Down User Load". Available online: https://www.loadview-testing.com/blog/the-importance-of-ramp-up-and-ramp-down-user-load/#:~:text=Ramp%20up%20speed%20during%20load%20test%20is%20speed,increase%20slowly%20before%20the%20start%20of%20peak%20time. [Accessed: 15-June-2021].
42. Rantos, K., Drosatos, G., Kritsas, A., Ilioudis, C., Papanikolaou, A., & Filippidis, A. P. (2019). A blockchain-based platform for consent management of personal data processing in the IoT ecosystem. *Security and Communication Networks*, 2019.
43. Agarwal, R. R., Kumar, D., Golab, L., & Keshav, S. (2020, May). Consentio: Managing consent to data access using permissioned blockchains. *In 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (pp. 1-9). IEEE.
44. Tith, D., Lee, J. S., Suzuki, H., Wijesundara, W. M. A. B., Taira, N., Obi, T., & Ohyama, N. (2020). Patient consent management by a purpose-based consent model for electronic health record based on blockchain technology. *Healthcare Informatics Research*, 26(4), 265-273.
45. Agbo, C. C., & Mahmoud, Q. H. (2020, October). Design and Implementation of a Blockchain-Based E-Health Consent Management Framework. *In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 812-817). IEEE.

46. Shah, M., Li, C., Sheng, M., Zhang, Y., & Xing, C. (2019, July). CrowdMed: A blockchain-based approach to consent management for health data sharing. *In International Conference on Smart Health* (pp. 345-356). Springer, Cham.
47. Aldred, N., Baal, L., Broda, G., Trumble, S., & Mahmoud, Q. H. (2019). Design and Implementation of a Blockchain-based Consent Management System. arXiv preprint arXiv:1912.09882.
48. "Ethereum." Available online: https://ethereum.org/en/whitepaper/. [Accessed: 22-June-2021].
49. "Hyperledger Fabric." Available online: https://www.hyperledger.org/wp-content/uploads/2018/08/HL_Whitepaper_IntroductiontoHyperledger.pdf. [Accessed: 22-June -2021].
50. Dabbagh, M., Kakavand, M., Tahir, M., & Amphawan, A. (2020, September). Performance Analysis of Blockchain Platforms: Empirical Evaluation of Hyperledger Fabric and Ethereum. *In 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)* (pp. 1-6). IEEE.
51. "Enterprise Data Management." Available online: https://www.slideshare.net/JBHSYED/enterprise-data-management-110708313. [Accessed: 22-June -2021].
52. "Blockchain applications in the United Nations system: towards a state of readiness."52. Available online: https://www.unjiu.org/sites/www.unjiu.org/files/jiu_rep_2020_7_english.pdf. [Accessed: 22-June -2021].

APPENDICES

Appendix A: Selected Smart Contract Code

The source code is the chaincode that is deployed on the network to collect data from users. The chaincode contains functionalities that users and organizations can utilize to communicate with the Blockchain network. The below code provides us the details of one main functionality of the chaincode.

```
func (t *SimpleChaincode) GetConsentHistory(ctx contractapi.TransactionContextInterface, consentID string)
    ([]HistoryQueryResult, error)
    {
    log.Printf("GetConsentHistory: ID %v", consentID)
    resultsIterator, err := ctx.GetStub().GetHistoryForKey(consentID)
    if err != nil {
        return nil, err
    }
    defer resultsIterator.Close()
    var records []HistoryQueryResult
    for resultsIterator.HasNext() {
        response, err := resultsIterator.Next()
        if err != nil {
            return nil, err
        }
        var consent Consent
        if len(response.Value) > 0 {
            err = json.Unmarshal(response.Value, &consent)
            if err != nil {
                return nil, err
            }
        } else {
            consent = Consent{
                ID: consentID,
            }
        }
        timestamp, err := ptypes.Timestamp(response.Timestamp)
        if err != nil {
            return nil, err
        }
        record := HistoryQueryResult{
            TxId:      response.TxId,
            Timestamp: timestamp,
            Record:    &consent,
            IsDelete:  response.IsDelete,
        }
        records = append(records, record)
```

Figure A-1: Code snippet from the chaincode

Appendix B: Network-config File for Setting up Caliper (with crypto materials)

The Figure shows a portion of the network-config file used to configure the caliper. IP addresses for the appropriate EC-2 instances should be provided. As seen in the Figure, Certificate Authority 1 is assigned to the IP address of instance 1. Additionally, while testing the network, crypto-materials should be carefully copied.

```yaml
certificateAuthorities:
  ca.org1.example.com:
    url: https://54.221.184.151:7054
    httpOptions:
      verify: false
    tlsCACerts:
      path: networks/fabric/kp/crypto-config/peerOrganizations/org1.example.com/tlsca/tlsca.org1.example.com-cert.pem
    registrar:
      - enrollId: admin
        enrollSecret: adminpw

  ca.org2.example.com:
    url: https://3.80.29.181:8054
    httpOptions:
      verify: false
    tlsCACerts:
      path: networks/fabric/kp/crypto-config2/peerOrganizations/org2.example.com/tlsca/tlsca.org2.example.com-cert.pem
    registrar:
      - enrollId: admin
        enrollSecret: adminpw

  ca.org3.example.com:
    url: https://3.89.228.238:10054
    httpOptions:
      verify: false
    tlsCACerts:
      path: networks/fabric/kp/crypto-config3/peerOrganizations/org3.example.com/tlsca/tlsca.org3.example.com-cert.pem
    registrar:
      - enrollId: admin
        enrollSecret: adminpw
```

Figure B-1: Network-config file

Appendix C: Config File for Benchmarking

The sample config file that is needed for benchmarking the caliper tests is given below.

```yaml
test:
  workers:
    type: local
    number: 1
  rounds:

    - label: Create Consent
      txNumber: 500
      rateControl:
        type: fixed-rate
        opts:
          tps: 200
      workload:
        module: benchmarks/scenario/simple/hf-v2.2/create.js

    - label: Read Consent
      description: Test description for the query performance of the deployed contract.
      txNumber: 1000
      rateControl:
        type: fixed-rate
        opts:
          tps: 600
      workload:
        module: benchmarks/scenario/simple/hf-v2.2/query.js
```

Figure C-1: Benchmarks used for the caliper testing