electronics

MDPI

Article

Design and Development of a Blockchain-Based System for Private Data Management

Prasanth Varma Kakarlapudi * and Qusay H. Mahmoud

Department of Electrical, Computer, and Software Engineering, Ontario Tech University, Oshawa, ON L1G 0C5, Canada; qusay.mahmoud@ontariotechu.ca
* Correspondence: prasanthvarma.kakarlapudi@ontariotechu.net

Abstract: The concept of blockchain was introduced as the Bitcoin cryptocurrency in a 2008 whitepaper by the mysterious Satoshi Nakamoto. Blockchain has applications in many domains, such as healthcare, the Internet of Things (IoT), and data management. Data management is defined as obtaining, processing, safeguarding, and storing information about an organization to aid with making better business decisions for the firm. The collected information is often shared across organizations without the consent of the individuals who provided the information. As a result, the information must be protected from unauthorized access or exploitation. Therefore, organizations must ensure that their systems are transparent to build user confidence. This paper introduces the architectural design and development of a blockchain-based system for private data management, discusses the proof-of-concept prototype using Hyperledger Fabric, and presents evaluation results of the proposed system using Hyperledger Caliper. The proposed solution can be used in any application domain where managing the privacy of user data is important, such as in health care systems.

Keywords: blockchain; data privacy; consent management

Citation: Kakarlapudi, P.V.; Mahmoud, Q.H. Design and Development of a Blockchain-Based System for Private Data Management. *Electronics* **2021**, *10*, 3131. https://doi.org/10.3390/electronics10243131

Academic Editors: Juan M. Corchado, Javid Taheri and Stefanos Kollias

Received: 4 November 2021
Accepted: 14 December 2021
Published: 16 December 2021

1. Introduction

Maintaining a user's sensitive information is one of the primary responsibilities of an organization, as data is one of an organization's most significant assets. With the swift rise of modern technology, businesses recognize the enormous value of utilizing and sharing data. This invokes the importance of data privacy. Data privacy governs how information is gathered, shared, and used. Practical data privacy concerns frequently revolve around (a) the extent to which data is shared with third parties, and (b) how data is legitimately gathered and stored. To increase user trust in data management, companies must demonstrate system transparency by providing the following information: (a) the objectives of data collection, (b) the data processors (third parties) involved, and (c) the extent of data being used. In this paper, a use case of healthcare research was used to present the working of a proposed private data management system.

The term "health research," sometimes also called "medical research" or "clinical research," refers to research that is performed to learn more about human health [1], which is essential to improve disease prevention and treatment. When research was not a part of healthcare, doctors would make the medical decisions based on their best estimates and experience, which were often incorrect [1]. The guesswork was eliminated with the introduction of health research, as the medicines are now entirely tested and proven successful before use. For example, data was collected from 9000 breast cancer patients, which led to the eventual development of Herceptin (used for treating breast and stomach cancer) [2]. Health research is certainly not possible without collecting and analyzing medical data from volunteers or patients.

The main priority in healthcare research is protecting the volunteer's data, which is highly sensitive and exposed. The essential principles that should be followed during the research to protect the data are collection limitation, data quality, purpose specification, use

limitation, security safeguards, openness, individual participation, and accountability [1]. The main concern in this process is that the patient/volunteer does not have complete knowledge regarding their data's usage. For example, if a patient's data is exchanged between various institutions, the patient may be unaware of the data sharing. Another essential concern is the storage location. Once the data is collected, it should always be stored in a secure location. Finally, it is crucial to collect consent periodically from the patient for the usage of their data. Having transparency, data security, and periodic consent collection in the system increases the degree of reliability of the system for the patients. This could eventually help the medical researchers, as more patients may be inclined to provide their data for the research. Blockchain can be highly crucial and influential for providing transparency in the system, as it is a decentralized ledger for recording transactions, which cannot be changed at a later date. Blockchain technology has the potential to standardize the management of trusted information, allowing medical researchers to access and use volunteer data while maintaining their information's protection.

Blockchain technology was introduced in 1991 by two researchers, Stuart Haber and W. Scott Stornetta, who wanted to design and implement a system that timestamps documents and that cannot be tampered with [3]. It gained much attention when used as an underlying technology in the cryptocurrency Bitcoin, introduced by Satoshi Nakamoto in 2008 [4]. Bitcoin is a peer-to-peer electronic cash system without a centralized third party, such as a bank. With Bitcoin, only the payment transactions are recorded transparently on a ledger, whereas a blockchain can record any form of transactions, such as votes, assets, and product inventories. A blockchain is an encoded digital ledger stored on multiple computers in a public or private network and comprises data records or "blocks" [5]. Currently, there are a vast number of blockchain-based applications that can make an immediate impact on society. Banks and insurance companies have shown the most interest in blockchain. Using this technology in a voting system in a democratic election is one example of a government using it.

As mentioned previously, blockchain is a ledger that records transactions or agreements made between nodes or network participants. Usually, in a blockchain, a block is formed when a transaction is submitted and verified by the other participants. Every block contains data, timestamps, the hash value of the block, and the previous block's hash value. The blocks are linked cryptographically, as every block stores the previous block's hash value, forming a chain. When a change is made to a transaction, the block's hash value will change, breaking the block's cryptographic link. Figure 1 manifests the structure of a blockchain. Finally, the technology brings enhanced security, greater transparency, and traceability to any system. There are, however, a few limitations to using blockchain for a consent management system (C.M.S.).

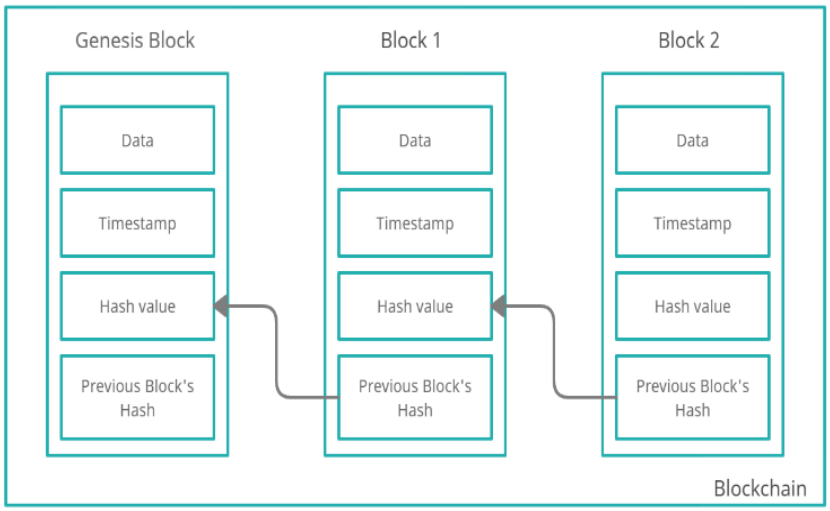

Figure 1. The structure of a Blockchain system.

According to the GDPR (General Data Protection Regulations), data should be removed or deleted after the agreed period, or whenever a user requests it. This privacy regulation is detrimental to blockchain technology, since it cannot be erased from the network once personal information is recorded. However, it can be avoided by storing personal and sensitive data on a different storage location, such as a cloud/interplanetary file system (IPFS) or locally and using the blockchain as an access log that contains the details of data transfer between the organizations. Overall, the advantages of this technology outweigh the privacy law setback.

In this paper, a controlled private data management system with blockchain and cloud is introduced. The proposed system collects consent from the users and stores it on the blockchain network as a form of transaction. The system allows users to store their data on a private cloud database from the front end. This system also presents a front end for third-party organizations that require a user's data. The user's consent details can be checked on the organization's front end, and data can be requested accordingly. The user can either approve or revoke the data request. Either transaction will be recorded on the network as a form of a new transaction. Later, users will have a log of organizations that access their data, making this proposed system completely transparent and traceable. The contributions of this paper are:

- The design and development of a blockchain-based consent management framework for private data;
- Implementation details of the proposed model on the Amazon Web Services (AWS) cloud with a case study;
- Performance evaluation of the developed prototype using Hyperledger Caliper—a benchmark tool that measures the performance of blockchain implementation.

The rest of the paper is organized as follows. In Section 2, we have discussed the background and related work. Section 3 provides the architecture of the system, and Section 4 discusses the implementation details with a use case. Performance results are shown in Section 5. Conclusions and future work are presented in Section 6.

2. Background and Related Work

The healthcare sector stores a massive amount of sensitive data, such as patient age, medical reports, etc. The amount of new data in 2020 is estimated to be approximately 2314 exabytes [6]. The leading data management issue in the healthcare industry is to ensure the confidentiality of the stored data. Second, consent should be obtained before performing any task, such as classification analysis. By disclosing user data transfers between companies or researchers, the user's trust is increased. The main reason for selecting blockchain technology for such consent operations is that it is built to keep the transactions unaltered. The only way to change the consent details is by adding another transaction to the network, stating that consent details are changed. Various blockchain platforms were considered to implement this system. Hyperledger was chosen, as it does not require a mining fee to add blocks to the network and because it is a permissioned blockchain.

Hyperledger is a modular blockchain framework, an open-source project from the Linux Foundation, designed by I.B.M. It was developed to meet enterprise-grade applications and industry-level solutions [7]. It is a private blockchain; thus, only a few people have access to see the transactions made. There are a few essential terms in Hyperledger, which are discussed in detail.

- Peer: Peers are similar to nodes or participants on the network but share a ledger privately among themselves. In the case of Ethereum and blockchain, all nodes are equal. However, in Hyperledger, there are a few different types of peers, such as anchor peers, committing peers, and endorsing peers. Anchor peers are identified outside of the network, and without an anchor peer, two networks cannot be connected. Committing peers are responsible for maintaining the ledger on the network. Finally, endorsing peers are helpful for validation purposes.

- Consensus: Consensus is a mechanism used to validate a block before adding it to the chain. Two types of consensus mechanisms are available in Fabric: lottery and voting. There are three phases of consensus in Fabric: endorsement, ordering, and validation.
- Chaincode: This is the smart contract (a computer program) that can be written in multiple languages, such as JavaScript, G.O., etc., and that runs on the peers on the network. Nick Szabo introduced a smart contract in 1998, and it is a self-executable program in which the terms of the buyer–seller agreement are written directly into lines of code [8]. In Hyperledger, a chain code is used to implement the business logic that governs how applications communicate with the ledger.
- M.S.P. (Membership Service Provider): Clients must have authenticated credentials to join a private network. M.S.P.s are a semi-abstract component that gives clients access to credentials.

In academic research, blockchain and its various forms and implementations are a widely discussed topic. The following sections go into how blockchain can be used for consent management in different fields, including healthcare, IoT, identity management, and data storage. A few general consent management systems are also mentioned.

2.1. Blockchain for Consent Management in Healthcare

Healthcare is one of the critical industries that could profit from blockchain technology, because since it is based on the distributed ledger concept, medical records may be easily exchanged across hospitals/doctors/researchers for various reasons, including maintaining a patient's data. MedRec is an implementation based on Ethereum that keeps and maintains an auditable history and records of the medical transaction for providers, regulators, and patients [9]. Because it is based on Ethereum, different incentives are provided to miners who authenticate the transaction. They also consider a second incentive technique that involves medical experts in the process of mining. Now, 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 [10]. However, this implementation does not discuss data sharing, such as 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 [11].

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 blockchain, Swetha, M.S. and the team discuss the system and framework for securing and protecting healthcare systems [12]. A permission-based blockchain, presented with authority proof for healthcare data sharing, is presented and discussed [13]. An emergency access control management system (EACMS) is introduced in [14], with the assistance of a Hyperledger composer. Tith, D. and his team presented a framework based on Hyperledger installed on a local network of four Linux-based computers and served as a user interface for patients and clinicians [15]. An E-Health consent management framework using the Hyperledger Fabric on the I.B.M. blockchain platform was presented in [16]. 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 who provided more data for research reasons via reward tokens and a creative cost structure [17]. The evaluation results for the proposed framework have not been discussed.

2.2. Blockchain for Consent Management in Identity Management

Traditionally, personal identity is established using documents such as a Social Security number, a driver's license, or a passport. However, there is no equivalent approach for guarding online identities nearly as effectively [18]. A digital identity can be produced and used as a real identity 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 [19], 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. Verification can always be requested because the information in a blockchain cannot be altered.

2.3. 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. A novel technique, termed interest groups, is presented in [20], in which each group adheres to a set of field data, allowing groups to 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 [21] using Ethereum and an IPFS (InterPlanetary File System). The prototype can store personal data and provides requested data services.

Furthermore, there are additional fields/sectors that make use of blockchain. 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 [22]. In [23], the team introduced an Ethereum-based system for managing data collected from IoT devices. The prototype also complies with the GDPR—General Data Protection Regulation. It enables users to manage their consent and, as a result, create their data access policy [23]. In [24], 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 [25]. It mainly focuses on ensuring higher throughput and low latency for the transactions. Another generic C.M.S. is presented in [26], 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.

2.4. Gaps in Existing Solutions

It is important to note that a few of the studies mentioned, in detail, how the prototypes were implemented. Not all systems that have been implemented have had discussions about how they are evaluated. Additionally, there exist privacy concerns and risks with prototypes. Ethereum implementations require Ether for invoking a function or operation or for the mining process, which is not suitable for managing private data. Additionally, some implementations explored storing personal data's hash references in blockchain, which is not recommended, as it might lead to data theft if the data is not secured correctly in different off-chain locations and places. Blockchain is constantly 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 to address and manage all the challenges, such as the storage of personal information in blockchain with the latest version of blockchain technology.

3. Proposed Solution

In this section, the design of the proposed solution is presented in detail. It is designed so that personal information can be stored off-chain in a cloud database and only consent information is written to the blockchain. The proposed system's architecture is thoroughly discussed.

3.1. Architecture

Due to several privacy guidelines, such as Article 17 of GDPR (the right to be forgotten), personal data cannot be saved on the blockchain, hence 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 [27]. The architecture of the system is shown in Figure 2. Users can utilize blockchain to record their consent details (through blockchain transactions) and use it as an access log, as demonstrated. Users can securely maintain their data in the cloud. Organizations can request the users for consent to use their data. When users grant appropriate consent to an organization, the admin can share the user's data. Finally, when using a blockchain transaction, users can revoke an organization's access at any time.

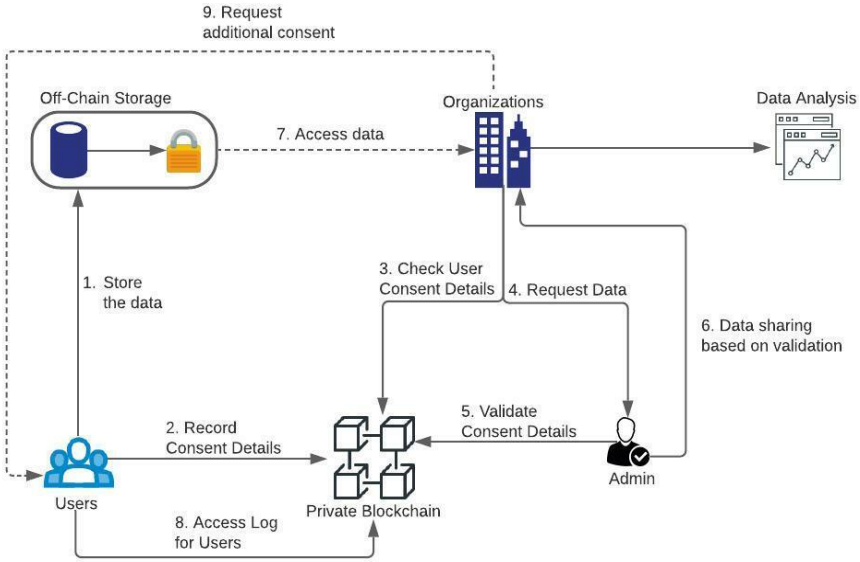

Figure 2. The proposed solution's system architecture.

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. Therefore, a permissioned blockchain was used to design and develop the system, instead of a public blockchain. The following sections will describe the prototype's functionality in detail.

3.1.1. Role of Admin and Integrity Relationship Assumptions

The sharing of data to organizations, in this design, is controlled and managed by an admin. The admin will share data with the requested organizations when the user on the network provides enough consent details. Additionally, an admin is required to ensure the maintenance of the database and that all data is deleted from the central database and any 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 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 rules would be rigorously adhered to by research organizations, including the deletion of data when consent is revoked, avoiding 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 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 to be informed appropriately. Upon understanding the reasons for collecting data, users can sign up for the service (through the user front end). After signing up, they can store their data on a cloud database that is both secure and private. Furthermore, security features, such as using a private link for accessing stored data and the blockage of public access, are utilized. The admin would take necessary periodic actions and precautions to ensure that the database is secure and healthy. To comply with privacy requirements, the servers utilized can be located within the country. Selecting the right availability zone accomplishes this. The other advantages of storing on the cloud is that servers are located in a warehouse where the access is restricted. Furthermore, the files stored on cloud servers are protected by encryption. This means they have been jumbled, making it far more difficult for hackers to access them. Data access can also be easily restricted to only authorized users. The following section details the blockchain activities that take place on the network.

3.1.3. Blockchain Network

A private blockchain is used for the design. After signing up for the service and storing the relevant information on the database, consent is written or recorded on the network of blockchains based on the Hyperledger Fabric using the user front end. Details of consent include I.D. (generated at the time of signing up for the service), name, email, consent details (partial/full), and organization details. For healthcare, a few types of research exist, such as prevention, physiological, and observational research. It can be mentioned by a user if they are 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.

With the use of the organization's front end, the 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 a "complete consent" value in the consent field and an "any organization" value 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.

Additionally, if additional access is required, it can be explicitly sought through the organization's front-end interface. A notification is sent to the user's account when a research organization requests additional consent. The user can accept or deny the request, and the information will be recorded in the network as a transaction. If the request has been approved, the admin can share data with the organization on an as-needed basis.

When a user wants their data deleted, the details can be recorded on the network. It will be updated as a form of transaction. 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 the data from their database or any other source. The following 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 [28]. It is installed on the peers, allowing for 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 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 shown below, in Figure 3.

```
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. Pseudocode of the chaincode.

3.1.5. Data Sharing

Data sharing among researchers has the potential to lead to substantial new discoveries in the field. In the case of healthcare, sharing data for research will boost confidence and trust in medical trial findings. When data is transferred between administrators and organizations, it must be protected and secure. For this reason, two distinct strategies for secure data sharing with other research organizations have been proposed, such as the use of AWS or an 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 this would serve to evade personal data replication or duplication. In addition, the admin will consider stringent policies such as an access control list (A.C.L.) and guidelines while considering the sharing of data. Organizations, after a set period, will not have any access to the provided data or files. Through an IPFS, large files can be shared, as shown in Figure 4. There are many advantages of sharing the file through an IPFS. When an IPFS is used to host static websites, the risks associated with single points of failure can be avoided and the benefits of a distributed infrastructure can be maximized. 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.

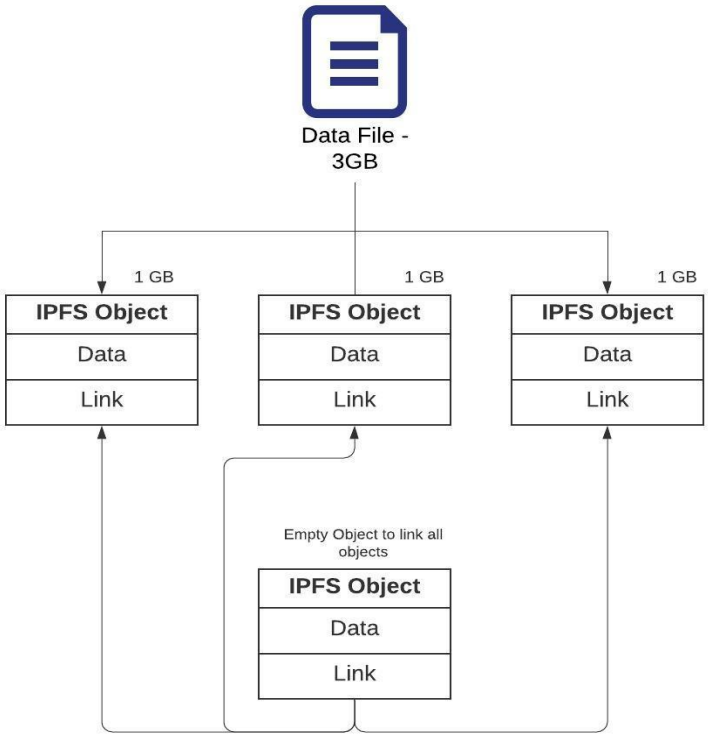

Figure 4. Data sharing with an IPFS (example).

It is essential 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, the system can contribute to controlled and secure data management that user can use and trust. Table 1 presents the features of the proposed system.

Table 1. Features of the proposed system.

Factors	Issues	Solutions with the Proposed 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 if the user requests it.	To avoid this, the personal data will be stored in a separate storage location. Storing hash references of sensitive data on the network will be avoided as the hash reference of the sensitive data might also be considered personal information soon. Cloud storage will be used instead, and another advantage of not storing data on the blockchain network is the network speed. The consent data can be fetched very quickly.
Access log	The main issue with the current consent management systems is that the users are not aware of the organizations accessing their data.	Users will be in control and can either accept or revoke the requests from the organization. The chaincode installed on the network will allow the users to fetch the history information of their consent details. This certainly brings out the traceability and transparency in the proposed system.

Table 1. *Cont.*

Factors	Issues	Solutions with the 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. User revoke requests can be investigated quickly to ensure 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 the user places a revoke request. The admin can also perform additional audits with the organizations to ensure the data is deleted entirely from the organization's system.
Privacy	Unauthorized users.	Using a permissioned H.F. will make sure that there are no unauthorized organizations in the network. Additional attribute-based controls could be set up to provide more granular access to the users with the help of chaincode.
Scalability	Improving the system performance.	The system's throughput can be increased by increasing storage and instances placed in the cloud. It is possible to do so by utilizing highly 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.	The latest fabric version, which has the newer chaincode lifecycle, was utilized with a React front-end to interact with the network instead of Compose, which is now depreciated.

3.2. 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 may be patients or volunteers that share their data with the hospitals or research organizations. The admin would be a government representative that will share the data. Hospitals/research organizations could use the data from the volunteers to perform medical analysis. 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 because of the development of IoT technologies. Citizens who use public infrastructure should know who has access to their data and, if possible, regulate access to the data. They can choose to share data obtained via the latest infrastructure (electricity meters) with any other entities. People could be users, and the admin could be a trusted government representative.

- Education: The users could be the students and could store their documents, such as transcripts, degrees, etc., on the database. The admin could 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.

The following section discusses the implementation details of the proposed system.

4. Prototype Implementation

Before implementing the system on the cloud, the prototype was initially implemented locally on a virtual machine. The following requisites are installed on the system; Curl, Nodejs, Git, Python, Go Language, Docker C.E., Docker Compose, and library tools. Once the prerequisites are installed, fabric samples are downloaded using curl. The environment variables are updated to ensure the working of the Golang.

After installing the prerequisites and downloading the Fabric, the test network (two peers, orderer, three CAs, two CouchDBs) is initialized. A channel is created after starting the test network. The chaincode for writing data onto the network is deployed on the channel. Figure 5 shows the steps involved in deploying a chaincode to the network.

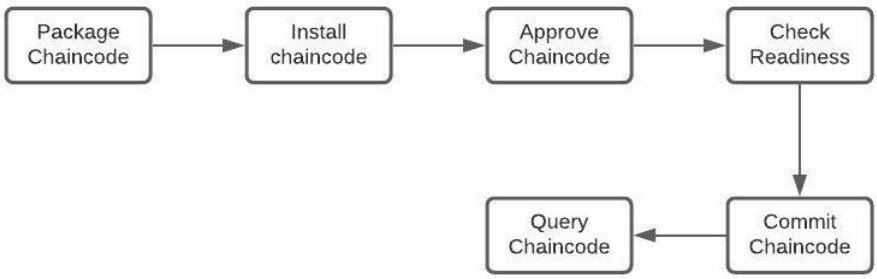

Figure 5. Chaincode lifecycle.

A few processes must be performed before interacting with the chaincode through the front-end application, such as enrolling the admin and registering the application user [29]. These interactions are between the C.A. and the application. Once the admin user and application user are enrolled, the credentials are stored in a wallet. When the credentials are present and have the appropriate authorization attributes associated with them, the application user will be able to access chaincode functions after obtaining references to the channel name and contract name from the sample application [29]. This is the back-end node application that is used by a back-end server to interact with the network.

React applications [30] for organizations and users were built to interact with the network with the back end running. 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. The system is implemented in the cloud to enhance throughput and avoid a single point of failure. The organizations are hosted on multiple EC2 instances instead of using a single instance, which will increase the TPS and will be decentralized, with each organization having its own virtual machine.

4.1. Current Implementation on the Cloud

Four virtual machines (EC-2 instances) on AWS have been created for the implementation of the prototype. They are all set up in a virtual private cloud (VPC) to ensure that they are secure and that only the administrator can modify them. The instances are of Ubuntu 18.04 with the following specifications, including 50 GB storage, 2 CPUs, and 4 GB of memory. Similar to local implementation, all the prerequisites were installed.

Accordingly, few environmental variables have been added and updated to accommodate GoLang's smooth workings. As mentioned earlier, the principal reason for implementing Hyperledger Fabric on several virtual machines (VM) 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 H.F. on several VMs.

Crypto materials were prepared for three organizations and one orderer organization. A central authority (C.A.), two ledgers, and two peers are included in each organization. In combination, there are three C.A.s, six ledgers, and six peers. For the orderer organization, there is a C.A., and there are three orderers. With the generation of certificates for all participants, the M.S.P. of each organization is created. The organization's M.S.P. is vital in the development of the genesis block. It is the first block that does not include any form of transaction data in it. However, it involves the M.S.P. IDs of the three specific organizations and their certificates. The channel consortium and name are included in the channel configuration transaction that will be utilized in the channel. The development of the channel tx and genesis block is depicted in Figure 6.

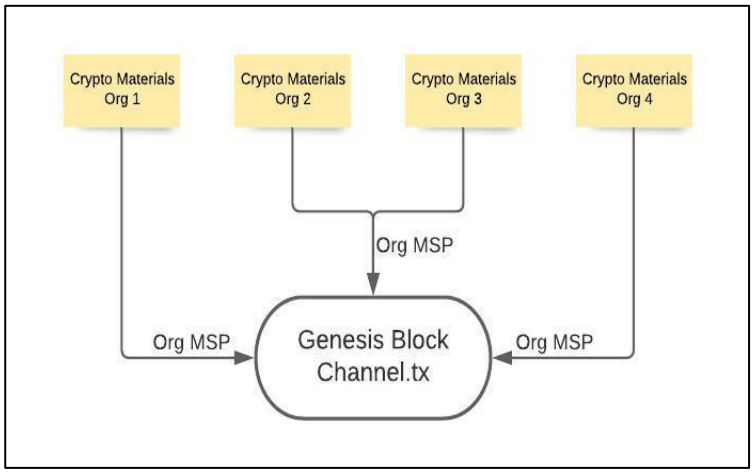

Figure 6. Genesis block development.

All the certificates are generated in a virtual machine and are later moved to their respective machines using the SFTP (S.S.H. file transfer protocol) tool FileZilla [31]. A Docker swarm network was created to ensure communication among them. In Figure 7, the addition of organizations to the channel is illustrated.

After installing Fabric on all the machines, the focus was on developing 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 Fabric, version 2.0. A chaincode has been developed to record the given information: ID, name, email, consent (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.

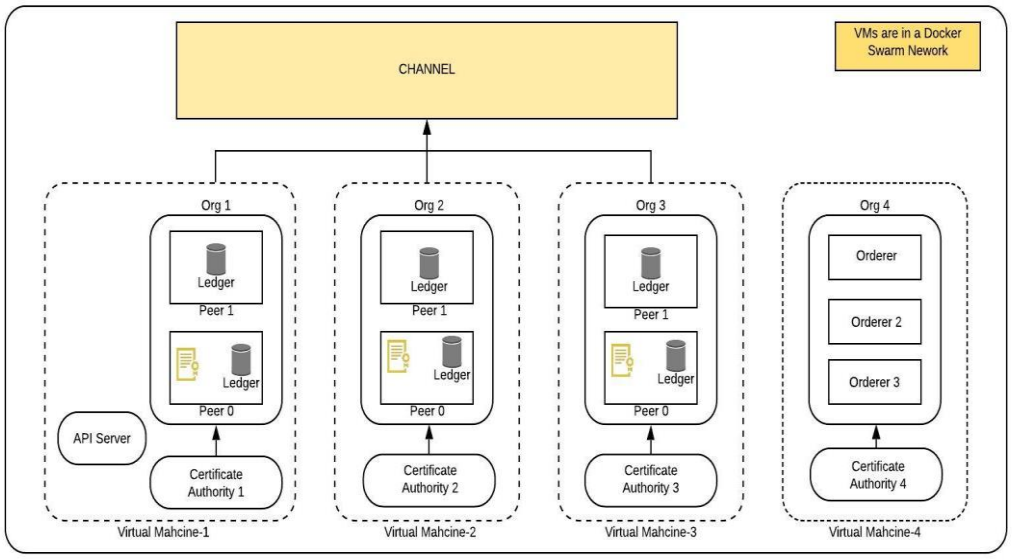

Figure 7. Setup of Fabric on virtual machines.

React is used for developing the front end that serves to invoke the functions of chaincode with the use of API endpoints, as illustrated in Figure 8. The users and organization's websites are hosted on AWS S3. This will aid in the website's high performance because it is easily scalable. 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 this, 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 the functionalities of the 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 Users can upload files to an S3 storage bucket, which can be used to store them. As 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;
- Insufficient and complicated information is present regarding Fabric SDK usage;
- Managing the development of the system's front end, such as CORS or cross-origin resource sharing.

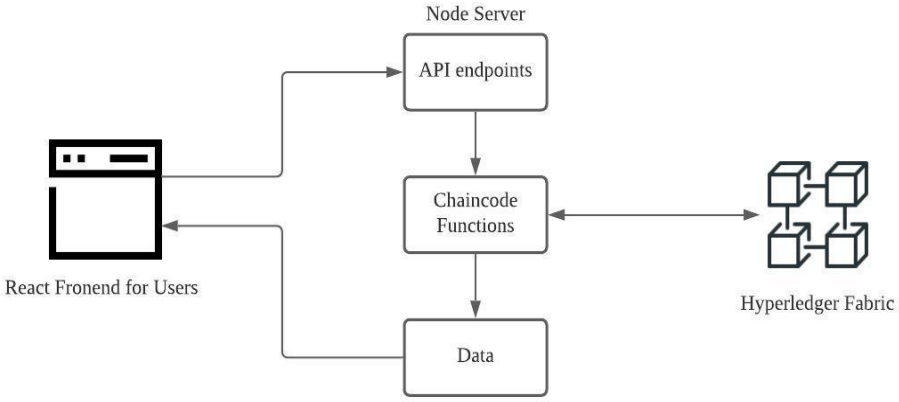

Figure 8. Interaction between the front end and the blockchain network.

4.2. Use Case—Healthcare Research

This section details the functionality of the proposed system and includes figures to help with understanding. After registering and storing data in the cloud, consent can be added to the network in the method illustrated in Figure 9.

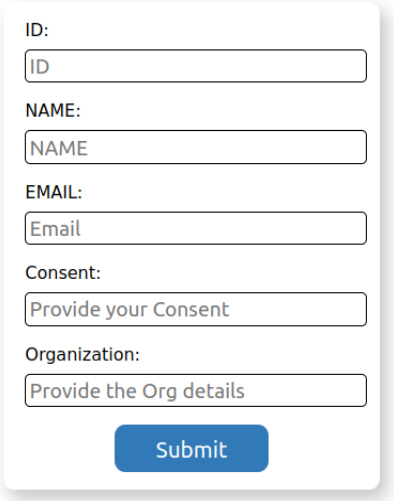

Figure 9. Adding consent details to the network.

Once the consent is added to the network by the users, the organizations can view the user's consent, as shown in Figure 10. The user has granted Org 1 partial consent to use their data for disease prevention studies solely. Additionally, users can modify their consent through the front end of the system. They have the opportunity to present further consent or revoke previously granted consent. Finally, users can check the access log, which allows them to track the organizations they have granted consent to. The access log for a user's consent information is depicted in Figure 11. The user initially consented to Org 1, but later retracted consent.

Figure 10. Reading user's consent details from the network.

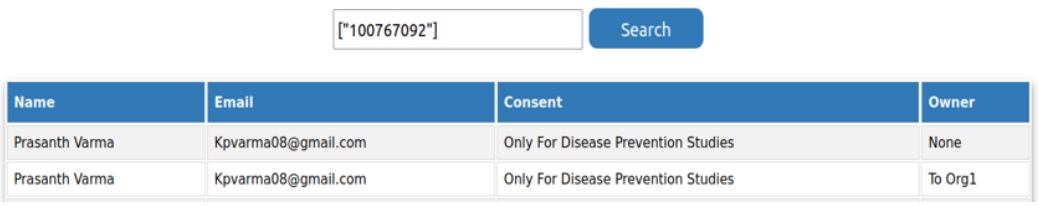

Figure 11. An access log of a user's consent information.

Overall, the proposed system is traceable and transparent. This will encourage users to contribute relevant data to healthcare research. The following section discusses the evaluation results for the system's local and cloud implementations.

5. Evaluation Results

First, results of the early locally implemented system are provided, followed by the results of cloud-based implementation. The permissioned blockchain network is running

using four virtual machines, meaning four organizations are set up in this configuration, one of which is the admin organization and the other three of which are research organizations. The admin and the users can access the blockchain using the admin organization. Amazon Cognito was used to create admin and user sign-ins to access the blockchain. By adding another virtual machine and connecting using Docker, more organizations can be added to the blockchain network. The database is built on the cloud, and the user can post data to the database through the front end.

5.1. Early Local Experimental Results

The configuration of the local virtual machine is given in Table 2. Peers for each organization are installed on separate ports within the same virtual machine in this configuration.

Table 2. System configuration of the local virtual machine setup.

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

The network's performance was initially analyzed utilizing J-meter. With a ramp-up speed of one second, 100 threads were taken for evaluation. Ramp-up speed is the rate at which new concurrent users attempt to access the system during a load test [32].

Experiments

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. This experiment uses 100 threads with a one-second ramp-up period. This test was repeated four times to ensure that there were no significant differences in the final output. Figure 12 depicts the combined results of all four experiments. The results are distinguished from one another using a different color (1, 2, 3, 4).

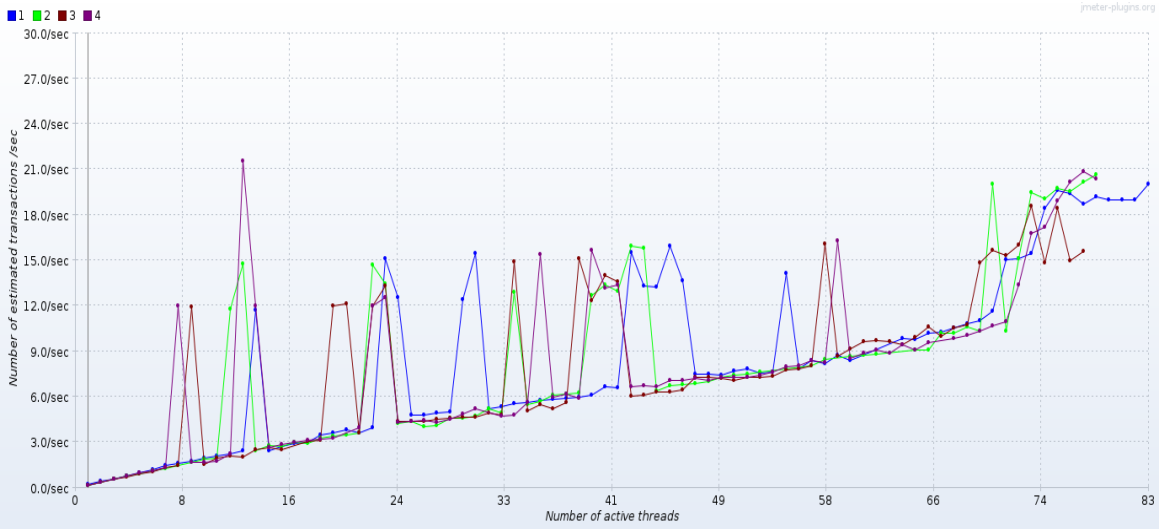

Figure 12. Overall results of the throughput in the local machine.

The y-axis represents the number of transactions per second, while the x-axis represents the number of active threads. As illustrated in Figure 9, there was always an average of 20 unsuccessful transactions per 100 users. Thus, the success rate of local implementation is approximately 80%.

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 Figure 13.

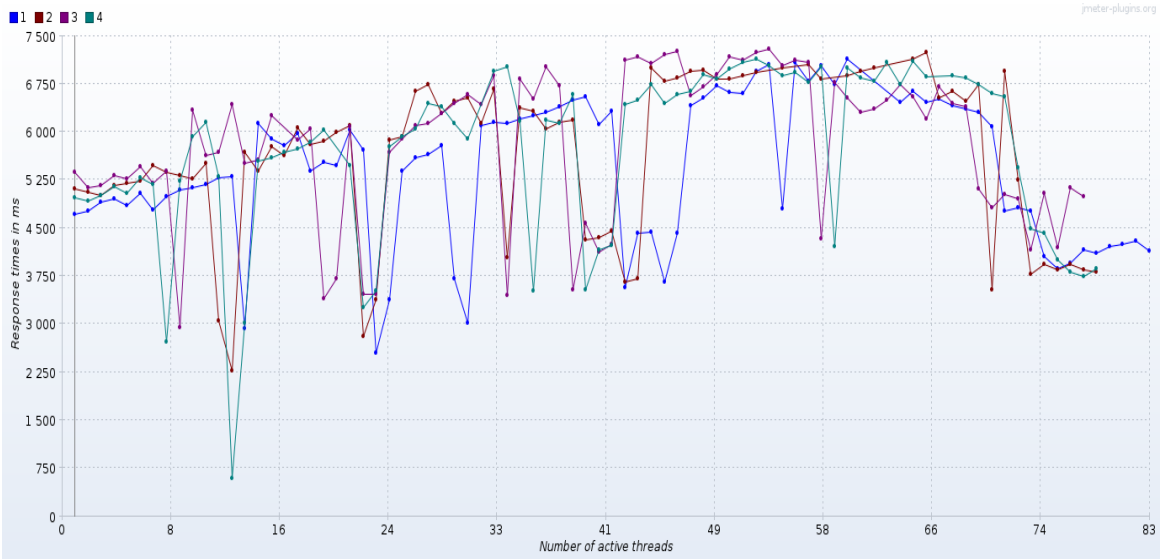

Figure 13. Combined results of the response times in a local machine.

The average response time for all results is approximately 5302.4 milliseconds (almost 5.3 s) 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 Experimental Results

In this study, a permissioned network was created, to which only specified organizations could be added. All virtual machines were configured identically and were in a private virtual private cloud (VPC) on AWS (Amazon Web Services) [33]. Similar to the local setup, four organizations were configured on four EC-2 instances. An admin and three research organizations are deployed on the cloud. Additionally, the database is deployed on the cloud within the VPC. To test the system's performance, experiments were run using Hyperledger Caliper [34]. The details of our system's configuration are included below in Table 3. The Docker file used to test the system is shown in Figure 14.

Table 3. System configuration of the cloud setup.

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

```yaml
version: "2"


services:
  caliper_2.2:
    container_name: caliper_2.2
    image: hyperledger/caliper:0.4.1
    command: launch manager --caliper-fabric-gateway-enabled
    environment:
      - CALIPER_BIND_SUT=fabric:2.1.0
      - CALIPER_BENCHCONFIG=benchmarks/scenario/simple/hf-v2.2/config.yaml
      - CALIPER_NETWORKCONFIG=networks/fabric/hf/network-config.yaml
    volumes:
      - ./caliper-benchmarks-hf:/hyperledger/caliper/workspace
    network_mode: host
```

Figure 14. Docker file used for setting up the Hyperledger Caliper.

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 working directory;
- Set the binding and run parameters that are required, as shown in Figure 14.

The Fabric version that is used in our implementation is 2.1.0. 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 Figure 15 is a snippet of the network configuration used to connect to the Caliper. Once the configuration is completed, the Docker container is started. To begin, two test cases were created to test the throughput and latency of the system: 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.

```yaml
name: Fabric
version: "1.0"
mutual-tls: false

caliper:
  blockchain: fabric
  # command:
  #   start: export FABRIC_VERSION=1.4.4;export FABRIC_CA_VERSION=1.4.4;docker-compose -f networks/fabric/docker-compose/2org1peercouchdb_s
  #   end: docker-compose -f networks/fabric/docker-compose/2org1peercouchdb_solo_raft/docker-compose-tls.yaml down;(test -z \"$(docker ps

info:
  Version: 2.1.0
  Size: 3 Orgs with 6 Peer
  Orderer: Raft
  Distribution: Multiple Host
  StateDB: CouchDB

clients:
  client0.org1.example.com:
```

Figure 15. Docker file used for setting up the Hyperledger Caliper.

5.2.1. Experiment—1

The throughput and latency of the system were evaluated in this experiment, which began by experimenting with a minimal number of transactions with a send rate of 1 TPS The throughput and latency of the system, by executing ten transactions, was measured. A throughput of 1 transaction per second (TPS) with ten transactions was achieved. The transaction processing speed (TPS) was 1.1 transactions per second. The most considerable latency was 2.27 s, and the minimum was 0.17 s. The average latency was approximately 1.38 s. The experiment's outcome is depicted in Figure 16.

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 16. Caliper results (Create consent) of experiment 1.

Figure 17 illustrates the findings from the evaluation of Read Consent measures. The average latency was 0.02 s, while the maximum and minimum values were respectively 0.02 and 0.01 s. The overall throughput of the Read Consent experiment is 1.1 TPS. Consequently, the transaction volume was increased 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 17. Caliper results (Read Consent) of experiment 1.

5.2.2. Experiment—2

The number of transactions for the subsequent experiment was increased to 1000 and 2000 for the Create and Read Consent experiments, respectively, with a 40 and 220 TPS send rate. This increased the throughput of the system. The results of this experiment are shown in Figure 18.

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 18. Caliper results for experiment 2.

The highest latency, in terms of results, is 13.87 s for the Create Consent experiment and 12.54 s for the Read Consent experiment. The most negligible latency is 0.49 s, and the max-

imum delay is 0.57 s, 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 s for Create Consent and 7.69 s for Read Consent, as shown in Figure 18. With 2000 transactions, the throughput was 133.2 TPS, with an average latency of 7.69 s. The maximum latency was 12.54 s, whereas the minimum latency was 0.57 s.

5.2.3. 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 was increased from 2000 to 2500, while the number of transactions requesting Create Consent remained constant. Figure 19 illustrates the outcomes of Create Consent, whereas Figure 20 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 19. 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 20. Caliper results (Read Consent) of experiment 3.

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

In general, the network was able to handle a higher volume of requests without experiencing any performance concerns. One failed transaction was observed 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 more 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 and scalable than a local virtual machine-based one. Table 4 presents the comparison between the proposed system and other healthcare-related systems.

Table 4. Comparison between systems.

Paper #	Implementation/Performance Evaluation	Comments
[9]	Prototype implementation details are given.	Does not provide implementation details.
[10]	Implementation with few performance analyses.	Does not cover the details of data sharing.
[11]	Implemented with Hyperledger Composer.	Does not have performance analysis. The composer is now depreciated.
[12]	The prototype is mentioned in this paper.	Does not provide implementation details
[13]	Uses multichain to implement private data management.	Performance evaluation of the system is not reported.
[14]	Implementation with the performance analysis is covered in this paper.	The composer is used in the system, which is now depreciated. Discussed only a few metrics of Fabric.
Proposed System	The implementation details and performance analysis were included.	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 4, the research papers [9] and [12] address only prototypes in the healthcare domain. The paper [10] does not examine the specifics of data exchange between users and other healthcare departments. It also does not involve any evaluation of the implemented prototype. 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 [35]. The prototype discussed in [11] involves the use of Hyperledger Composer, which is depreciated. The depreciated Hyperledger Composer is also used in [14], and the response time for retrieving patient data was 5683 ms; no other metrics are discussed. In the prototype, Caliper was used to measure the system. Additionally, metrics were demonstrated with varying quantities of data records.

6. Conclusions and Future Work

The design and implementation of a consent management system for private data were examined in this paper. The implementation details of the proposed system from the perspective of a healthcare case study were discussed. The proposed system is intended for use by individuals and organizations. In the 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. This system was created with the security of sensitive data in mind. The prototype capitalizes on blockchain's core benefits, such as immutability, to give users traceability and transparency.

Additionally, the technology improves the present consent collection process with blockchain by informing the user of the purpose for data collection. The methods for sharing sensitive data that are included in the process of sharing via an IPFS and Amazon S3 were covered. The AWS access policy that will be used to ensure that the data is not accessible after the agreed-upon period was reviewed.

Future Work

The implementation has shown that blockchain is a viable technology for developing a consent management system. While features of blockchain provide users with a new level of trust, the evaluation of the proposed system focused on scalability and throughput with an evaluation of enhanced privacy protection for future work. In addition, there are few limitations of the proposed prototype, which 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 I.D., consent can be updated on the network. I.D. 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 I.D. loss;
- The cost of the production-ready application will require a certain amount, compared to traditional systems, as it depends highly on the resources allocated, such as processing units, memory, storage, etc. This will also affect the network speed of the blockchain in reading and writing data.

Blockchain can also be used for data validation. It is intended to incorporate this feature into the proposed system by making a few minor design changes. The second challenge will be addressed by utilizing the attribute-based access control (ABAC) technique to implement the smart contract. This will also increase the users' trust. Finally, the application will be deployed in a practical situation that benefits both users and organizations.

Author Contributions: Writing—original draft preparation, P.V.K.; supervision and writing—review and editing, Q.H.M. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the support of the Office of the Privacy Commissioner of the Canada Contributions Program 2020–2021.

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

References

1. Participating in Health Research Studies. Available online: https://guides.library.harvard.edu/c.php?g=389023&p=2639499 (accessed on 22 May 2021).
2. Gostin, L.O.; Levit, L.A.; Nass, S.J. *Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research*; National Academies Press: Washington, DC, USA, 2009; p. 114.
3. Haber, S.; Stornetta, W.S. How to timestamp a digital document. In *Advances in Cryptology-CRYPTO' 90, Proceedings of the Conference on the Theory and Application of Cryptography, Berlin/Heidelberg, Germany, 18 May 2001*; Springer Nature: New York, NY, USA, 1990; pp. 437–455.
4. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Decentralized Bus. Rev.* **2008**, 21260. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 22 May 2021).
5. Using Blockchain to Improve Data Management in the Public Sector. Available online: https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/using-blockchain-to-improve-data-management-in-the-public-sector (accessed on 22 May 2021).
6. Total Amount of Global Healthcare Data Generated in 2013 and a Projection for 2020. Available online: https://www.statista.com/statistics/1037970/global-healthcare-data-volume/ (accessed on 1 June 2021).
7. Hyperledger. Available online: https://www.ibm.com/topics/hyperledger (accessed on 1 June 2021).
8. Szabo, N. Formalizing and securing relationships on public networks. *First Monday* **1997**, *2*. [CrossRef]
9. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. Medrec: Using Blockchain for medical data access and permission management. In Proceedings of the 2016 2nd International Conference on Open and Big Data (OBD), Vienna, Austria, 22–24 August 2016.
10. Liang, X.; Zhao, J.; Shetty, S.; Liu, J.; Li, D. Integrating Blockchain for data sharing and collaboration in mobile healthcare applications. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017.
11. Rouhani, S.; Butterworth, L.; Simmons, A.D.; Humphery, D.G.; Deters, R. MediChain TM: A secure decentralized medical data asset management system. In Proceedings of the 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), Halifax, NS, Canada, 30 July–3 August 2018.

12. Swetha, M.S.; Pushpa, S.K.; Muneshwara, M.S.; Manjunath, T.N. Blockchain-enabled secure healthcare Systems. In Proceedings of the 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT), Hyderabad, India, 20–21 December 2020.
13. Al Asad, N.; Elahi, M.T.; Al Hasan, A.; Yousuf, M.A. Permission-Based Blockchain with Proof of Authority for Secured Healthcare Data Sharing. In Proceedings of the 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), Dhaka, Bangladesh, 28–29 November 2020.
14. Rajput, A.R.; Li, Q.; Ahvanooey, M.T.; Masood, I. EACMS: Emergency access control management system for personal health record based on Blockchain. *IEEE Access* **2019**, *7*, 84304–84317. [CrossRef]
15. Tith, D.; Lee, J.S.; Suzuki, H.; Wijesundara, W.M.A.B.; Taira, N.; Obi, T.; Ohyama, N. Patient consent management by a purpose-based consent model for electronic health record based on blockchain technology. *Healthc. Inform. Res.* **2020**, *26*, 265–273. [CrossRef] [PubMed]
16. Agbo, C.C.; Mahmoud, Q.H. Design and Implementation of a Blockchain-Based E-Health Consent Management Framework. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (S.M.C.), Toronto, ON, Canada, 11–14 October 2020.
17. Shah, M.; Li, C.; Sheng, M.; Zhang, Y.; Xing, C. CrowdMed: A blockchain-based approach to consent management for health data sharing. In Proceedings of the International Conference on Smart Health, Shenzhen, China, 1–2 July 2019.
18. 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. [CrossRef]
19. 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.
20. 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.
21. 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.
22. 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. [CrossRef]
23. Rantos, K.; Drosatos, G.; Kritsas, A.; Ilioudis, C.; Papanikolaou, A.; Filippidis, A.P. A blockchain-based platform for consent management of personal data processing in the IoT ecosystem. *Secur. Commun. Netw.* **2019**, *2019*, 1431578. [CrossRef]
24. 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.
25. Agarwal, R.R.; Kumar, D.; Golab, L.; Keshav, S. Consentio: Managing consent to data access using permissioned blockchains. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2–6 May 2020.
26. Aldred, N.; Baal, L.; Broda, G.; Trumble, S.; Mahmoud, Q.H. Design and Implementation of a Blockchain-based Consent Management System. *arXiv* **2019**, arXiv:1912.09882.
27. Esposito, C.; De Santis, A.; Tortora, G.; Chang, H.; Choo, K.K.R. Blockchain: A panacea for healthcare cloud-based data security and privacy? *IEEE Cloud Comput.* **2018**, *5*, 31–37. [CrossRef]
28. What is Chaincode? Available online: https://fabrictestdocs.readthedocs.io/en/latest/chaincode.html (accessed on 15 June 2021).
29. Hyperledger—Write First App. Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/write_first_app.html (accessed on 10 June 2021).
30. React. Available online: https://reactjs.org/ (accessed on 15 June 2021).
31. FileZilla. Available online: https://filezilla-project.org/ (accessed on 15 June 2021).
32. Glenn Lee. 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 on 15 June 2021).
33. Amazon Web Services. Available online: https://aws.amazon.com/ (accessed on 16 June 2021).
34. Hyperledger Caliper. Available online: https://hyperledger.github.io/caliper/v0.3.2/fabric-config/ (accessed on 12 June 2021).
35. Hyperledger Whitepaper Metrics. Available online: https://www.hyperledger.org/wp-content/uploads/2018/10/HL_Whitepaper_Metrics_PDF_V1.01.pdf (accessed on 15 June 2021).