electronics

MDPI

Article

Hyperledger Healthchain: Patient-Centric IPFS-Based Storage of Health Records

Vinodhini Mani [1,*], Prakash Manickam [2], Youseef Alotaibi [3], Saleh Alghamdi [4] and Osamah Ibrahim Khalaf [5]

1 SRM Institute of Science and Technology, Computer Science Engineering, Kattankulathur 603203, India
2 SRM Institute of Science and Technology, Data Science and Business Systems, Kattankulathur 603203, India; prakashm2@srmist.edu.in
3 Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia; yaotaibi@uqu.edu.sa
4 Department of Information Technology, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia; s.algamedi@tu.edu.sa
5 Al-Nahrain Nano-Renewable Energy Research Center, Al-Nahrain University, Baghdad 10072, Iraq; usama.ibrahem@coie-nahrain.edu.iq
* Correspondence: vm6461@srmist.edu.in

Abstract: Blockchain-based electronic health system growth is hindered by privacy, confidentiality, and security. By protecting against them, this research aims to develop cybersecurity measurement approaches to ensure the security and privacy of patient information using blockchain technology in healthcare. Blockchains need huge resources to store big data. This paper presents an innovative solution, namely patient-centric healthcare data management (PCHDM). It comprises the following: (i) in an on-chain health record database, hashes of health records are stored as health record chains in Hyperledger fabric, and (ii) off-chain solutions that encrypt actual health data and store it securely over the interplanetary file system (IPFS) which is the decentralized cloud storage system that ensures scalability, confidentiality, and resolves the problem of blockchain data storage. A security smart contract hosted through container technology with Byzantine Fault Tolerance consensus ensures patient privacy by verifying patient preferences before sharing health records. The Distributed Ledger technology performance is tested under hyper ledger caliper benchmarks in terms of transaction latency, resource utilization, and transaction per second. The model provides stakeholders with increased confidence in collaborating and sharing their health records.

Keywords: IPFS; health records; blockchain; privacy; security; scalability

Citation: Mani, V.; Manickam, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. Hyperledger Healthchain: Patient-Centric IPFS-Based Storage of Health Records. *Electronics* **2021**, *10*, 3003. https://doi.org/10.3390/electronics10233003

Academic Editor: Juan-Carlos Cano

Received: 18 October 2021
Accepted: 30 November 2021
Published: 2 December 2021

1. Introduction

Providing a secure and private access control model is one of the most critical aspects of healthcare. Big data are used to store and access a large amount of health information over the Internet in the big data age. Cloud networking plays an increasingly critical role in this process. Traditional electronic health record (EHR) systems present numerous privacy and security challenges despite their ease of use and reliability. [1] Health record (HR) contains a lot of sensitive information about patients and diagnoses; therefore, it is considered the most sensitive data collection method. HR data has, however, become more susceptible to breach due to the advancement of the internet and advancement in digital healthcare systems [2]. Therefore, the security and privacy of HR data need to be taken into account when assessing a decentralized and trust-based approach [3].

1.1. Motivation

The electronic medical records (EMRs), electronic health records (EHRs), and personal health records (PHRs), clinical images, and patient information such as doctor names, individual measurements, and home-checking gadget information are centralized in cloud

databases used by content organizations. A centralized database can expose us to cyber-attacks, which threaten the privacy and security of EHR. Health providers and other stakeholders have difficulty sharing health information due to varying standards and formats. The EHR can be permanently lost when it is deleted from the hospital's database. This is yet another issue to be addressed carefully. It is imperative that systems are tamper-proof so that only authorized individuals can gain access to them. A further problem is that current healthcare systems do not completely empower patients to manage their health records since they are managed by our service providers. As healthcare data continues to increase, medical records' security and scalability have become major concerns. The overview of the existing health record system architecture is shown in Figure 1.

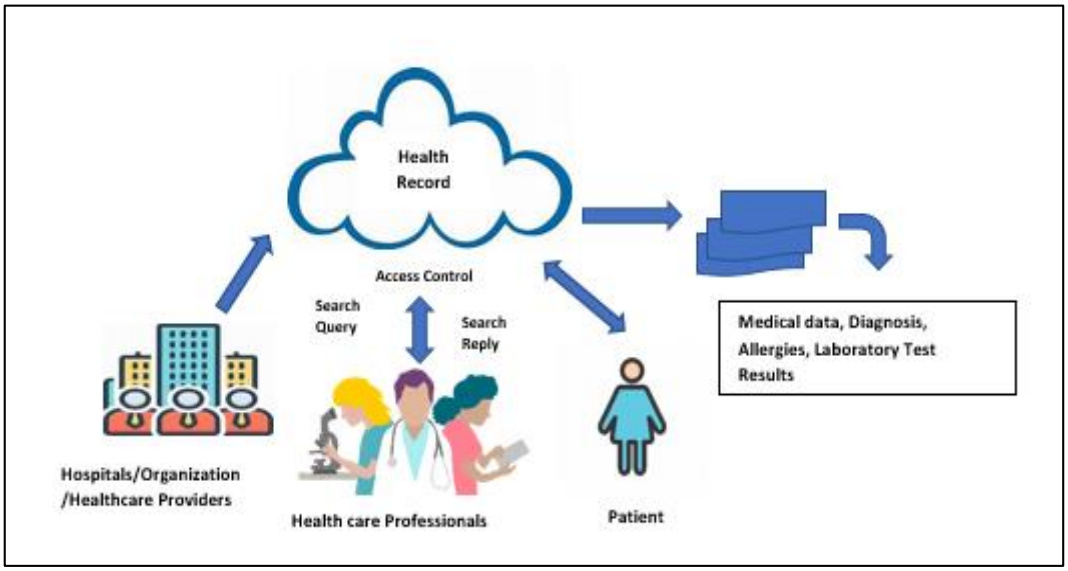

Figure 1. Existing health record system.

1.2. Contribution

Technology that stores data effectively is therefore required. The paper contribution is shown in Figure 2. To achieve this, an effective patient-centric distributed architecture for storing patient-centric data is simultaneously concerned with privacy, security, integrity, interoperability, and scalability. This research identifies security and privacy issues and examines a blockchain-based approach to conquer them. Second, we have created a novel algorithm for storing and securely gaining access to records using blockchains.

In this research, we are developing a permissioned network in a hyper ledger fabric-based PCHDM framework that ensures health record integrity, security, scalability, and privacy while providing patients with complete control. Health information is mainly stored on the blockchain as hashes, whereas the original vast quantities of data are maintained off-chain in IPFS to ensure scalability and efficiency.

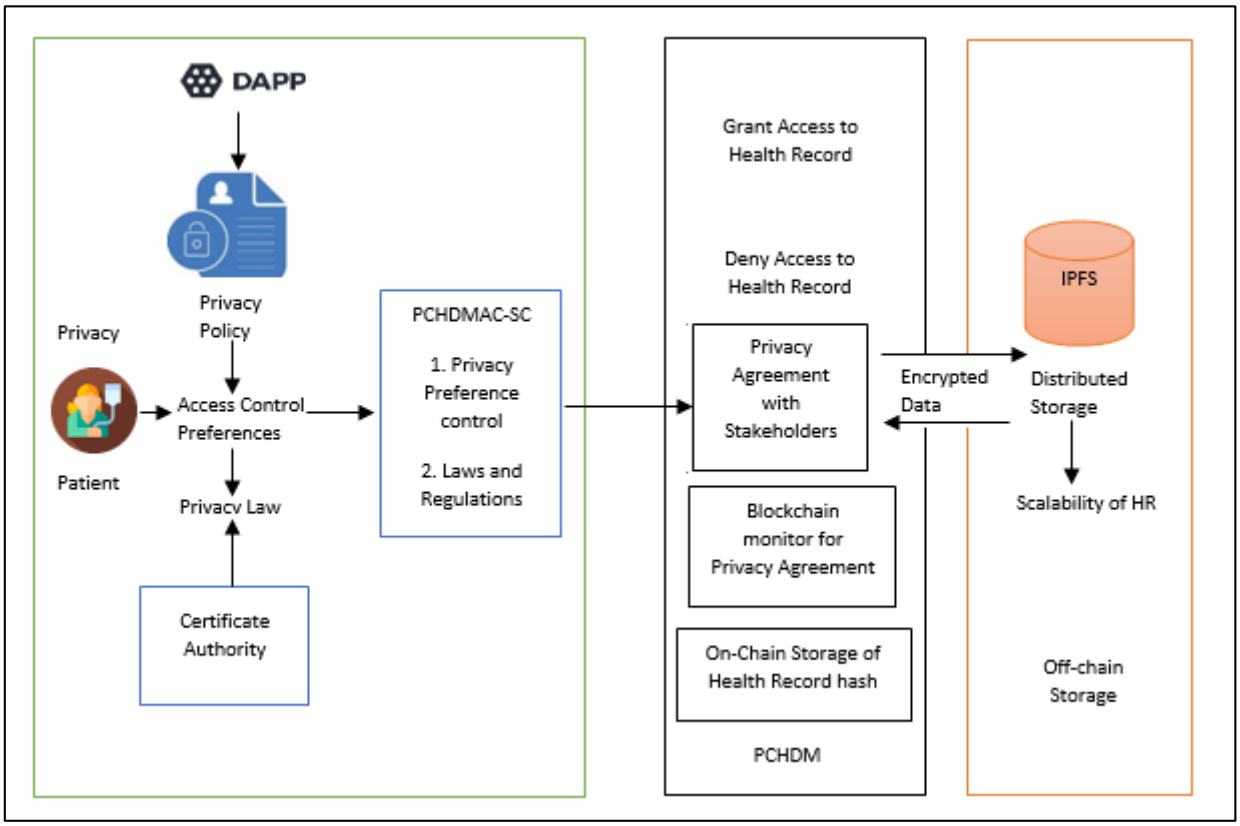

Figure 2. Contribution of the paper.

The smart contract chain code protocol in this research is named Patient-Centric Healthcare Data Management Access Control-Smart Contract, PCHDMAC-SC, where PCHDMAC-SC is the role-based access control chain code written using Access Control language for registered stakeholder groups and does not use any form of incentive mining beyond equitable access to the system. After the registration of stakeholders, the Fabric Certificate authority generates the role-based unique ID for the registered stakeholders. Each stakeholder is created with public and private key pairs for secure storage and sharing of the health records. A patient's health record can be created by a doctor. After creation, the patient commits the encrypted health record, which will be stored in IPFS permanently. The generated health record hash from IPFS is preserved via the Hyperledger blockchain.

The patient wants the doctor to be able to update his or her health record by granting access. The temporary view, called the patient-centric view of the health record, is generated from IPFS. The doctor can update the patient record using the patient-centric view, and then the patient commits to the update using their key pairs to store the updated health record again in IPFS. Hence, interoperability is achieved by developing this framework. A doctor session will expire before the hash value is committed to the ledger of the Hyperledger fabric using Couch DB. Therefore, no doctor has access to a patient's record without the patient's permission, which implies that the patient's privacy in the system has been protected.

To share the records with stakeholders, the patient wants to grant access that retrieves a patient-centric view of the record, i.e., partial information about the patient retrieved from the IPFS system. That means the privacy of patient health information is preserved.

Smart contracts PCHDMAC-SC are created at the back end for multiple healthcare processes, and then each patient manages the permissions to access data within the healthcare ecosystem.

Hence, this system protects the privacy of the patient using role-based access control, and its scalability is tested and evaluated under hyper ledger caliper benchmarks. The

performance evaluation results show our proposed system has improved scalability and interoperability compared with the existing system.

1.3. Organization

The remainder of the paper is structured as follows: Section 2 addresses the related work. Section 3 presents the framework of the proposed PCHDM. Section 4 presents the implementation of the proposed PCHDM. Section 5 discusses the results and performance evaluation. Section 6 concludes the paper and discusses future work.

2. Related Works

In this section, we discuss research related to the use of blockchain technology in e-healthcare to ensure secure data storage and efficient access control. The growth of electronic health systems is hindered by security issues. Blockchain technology has arrived and provides a cryptographic solution to the problem of security, but it has challenges, such as privacy, scalability, and interoperability. Throughout the past two decades, medical companies have experienced additional challenges caused by record breach episodes within large medical data centers [4]. In the early days of blockchain technology, MedRec [5] became the first suggestion for an electronic patient record management system that was implemented by blockchain technology. Ethereum blockchain and smart contracts store detailed accessibility data. The blockchain does not store medical records, but rather the third-party database operated by healthcare providers. Infringement or misuse of these records is therefore still possible. By recording data on the blockchain, healthcare management systems [6] encrypt patient keys. To decrypt data, hospitals and researchers obtain consent from a patient's public key to decode the data. This contrasts dramatically with our approach, in which patients are the only ones who have access control to their data. On the public blockchain, any node can join the network and their transactions are transparent [7]. The Medchain is a blockchain based application that permits hospitals, pharmacies, and patients to share healthcare data [8]. Data is stored on-chain, but they suffer from scalability and privacy problems. With blockchain-based smart contracts [9], the health of patients could be tracked using a blockchain-based IoT platform. A blockchain-driven system for tracking electronic healthcare records was proposed by the author in [10]. The author of [11] presented a blockchain-based architecture that integrates distributed health records across node models to maximize replication of health data. The author of [12] used smart technology called blockchain, which is a decentralized network with smart contracts, to store and share data with security. In [13], a privacy-preserving system was modeled for remote patient monitoring. The author [14] has proposed a permissioned blockchain with an access control audit log for storing the health record but it suffers from privacy issues due to the sharing of audit log information to all stakeholders. The author of [15] discusses the security of medical data using blockchain technology but faces privacy and scalability concerns. The author in [16] has proposed a ring structure-based access control that ensures privacy and the system is not stable due to scalability issues as the data is stored as an on-chain database. An intelligent data management framework was proposed for the cyber system in [17]. As described in [18], the decentralized storage and access of records effectively utilizes the network's power and resources. As in [19], the author used high-end privacy-enhancing technologies, such as homomorphic encryption, that allow data to be processed while remaining completely encrypted so that vulnerabilities can be prevented. As a privacy enhancement technique, the author [20] used zero-knowledge proofs along with proof authority consensus for mutual authentication to ensure nodes were not engaging in malicious behavior. A number of cryptographic mechanisms using blockchain technology are shown in Table 1, which can be used to prevent tampering, efficiently store and share data.

Table 1. Existing Blockchain-based healthcare techniques.

Ref.	Implemented Challenge	Problem to Be Addressed
[5]	Data Integrity and Interoperability	Privacy and Scalability
[8]	Sharing of Data and Integrity of data	Privacy and Scalability
[9]	Public Data Access and Integrity of Data	Interoperability and authentication
[11]	Interoperability	Security, Privacy and Scalability
[13]	Privacy and Security	Scalability and Interoperability
[14]	Scalability and Interoperability	Privacy
[15]	Security	Scalability and Privacy
[16]	Security and Privacy	Scalability

EHR systems are still not sufficiently interoperable for ensuring privacy, security, and effective access control. With the implementation of a permissioned blockchain framework and Practical Byzantine Fault Tolerance as the consensus algorithm, we are able to resolve most of the existing challenges in the eHealth environment to enable decentralized storing and sharing of health information while ensuring patient confidentiality, privacy, and scalability.

3. Framework of Proposed PCHDM

3.1. Preliminary Requirements of PCHDM Framework

3.1.1. Hyperledger Fabric Blockchain

The Hyperledger blockchain network is permission-based and requires users to sign up to use it. Permission on the network is controlled using Hyperledger modeling and access control languages. Hyperledger Fabric is a platform for distributed ledger solutions underpinned by a modular architecture delivering high degrees of confidentiality, resilience, flexibility, and scalability. Medical information is often highly sensitive, in both a social and legal sense, so a closed blockchain such as Hyperledger Fabric helps to retain the necessary privacy required for such an application. Hyperledger Fabric is a better solution for managing access to health records, as it accommodates for multiple layers of permission, meaning the owner of a set of data can control which parts of their data are accessed. Smart contract SC stores rules relating to contract negotiations. The framework designed for role-based access control chain code is named as PCHDMAC-SC.

For medical information sharing efficiently and reliably without having to rely on a single authority, we are employing Hyperledger Fabric, a permissioned blockchain based on pre-specified parties. Hyperledger Fabric offers the advantage of employing the Byzantine fault tolerance consensus protocol without requiring mining or an associated currency as a means of achieving consensus. Hyperledger blockchain uses a Merkle Directed Acyclic Graph tree structure as its state database, which can be replicated using IPFS objects. Therefore, IPFS can be used to model an off-chain and on-chain blockchain for health record storage. By implementing the PCHDMAC-SC protocol, we created a transparent, fine-grained access control system that prevented hacking without patient consent using a Hyperledger blockchain.

3.1.2. Distributed File System-IPFS

A cryptographic hash represents a unique fingerprint for each file within IPFS, a peer-to-peer (P2P) protocol. To make the contents immutable, the hash address is applied [21]. Merkle DAGs combine Merkle trees with DAGs in IPFS file storage. Rather than relying on location-based addressing, IPFS's key feature is that access to health record can be accomplished through content-based addressing. Due to IPFS, bandwidth costs can be reduced, record download speeds can be enhanced, and a large volume of data can be distributed without duplication, which can save storage space. The hash value of an IPFS file cannot be changed so IPFS is an immutable storage mechanism.

3.1.3. Services Provided to Members

A cryptographic procedure that validates identity, generates and verifies signatures, and generates and verifies certificates, as well as verifying the identity of the user is described in [22]. In this Hyperledger framework, Fabric-Certificate Authority (CA) of [23] acts as the interface for providing Services to Members who have registered on the network as shown in Figure 3.

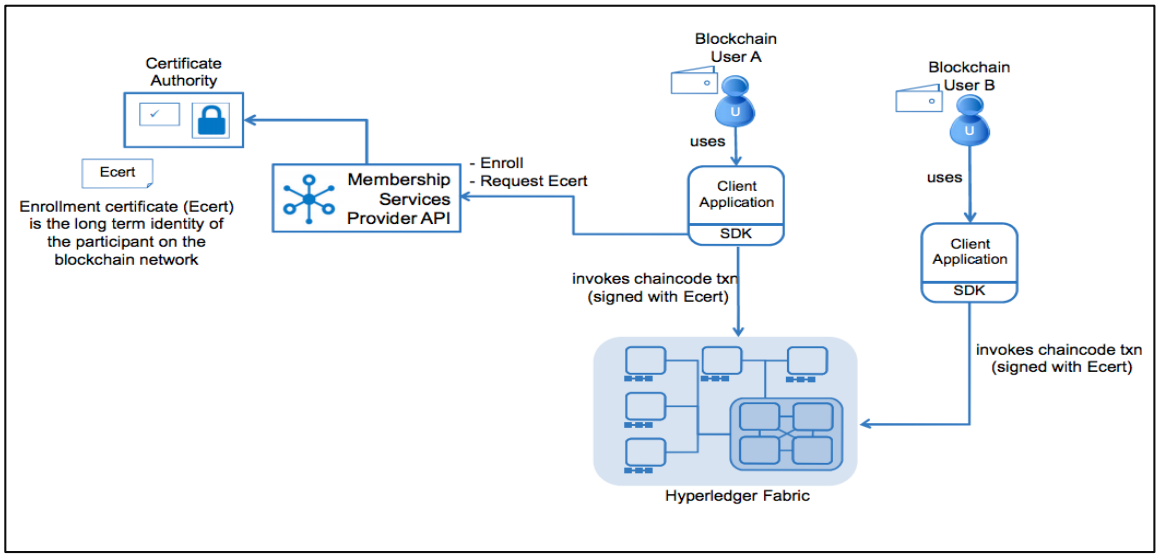

Figure 3. Certificate authority of Hyperledger Fabric.

3.2. A Background of the PCHDM System

An illustration of health chain's proposed framework can be seen in Figure 4. User requests are sent to the fabric network by the Dapp through an API called the composer, which is interactively handled by the Dapp admin. Through GET calls to the composer, the Angular framework can access the on-chain database and retrieve data based on the current state as returned by the API. Smart contract composer creates decentralized applications based on blockchain business networks. With Hyperledger Fabric [24] network stakeholder can validate medical data entries via smart contracts named chain codes. This technology was developed for distributed ledger solutions. Bitcoin [25] was created specifically for financial transactions, and Hyperledger is for storing health records.

The proposed system architecture as shown in Figure 5 has a Hyperledger Composer users permissioned blockchain based on Hyperledger Fabric to develop web applications for single organizations using three peer nodes. Three peer nodes are used by the organization, one serving as a validating peer node, while another serves as an ordering node that is used to register stakeholders. In this system, multiple peers access the corresponding database, IPFS for distributed storage of data, a Solo Order Node, a Data Certificate Authority, a Membership Service Provider, and smart contracts for blockchain connectivity. In order to verify the system's scalability, multiple peers can be added to multiple locations on different machines. Smart contractors have access to ledgers and have ledger access through this framework. Peer nodes are connected to the application, which then updates the ledger via smart. The three peer nodes in the organization are peernode0 (PE0), peernode1 (PE1), and peernode2 (PE2) each of which contains its own ledger and smart contract copies.

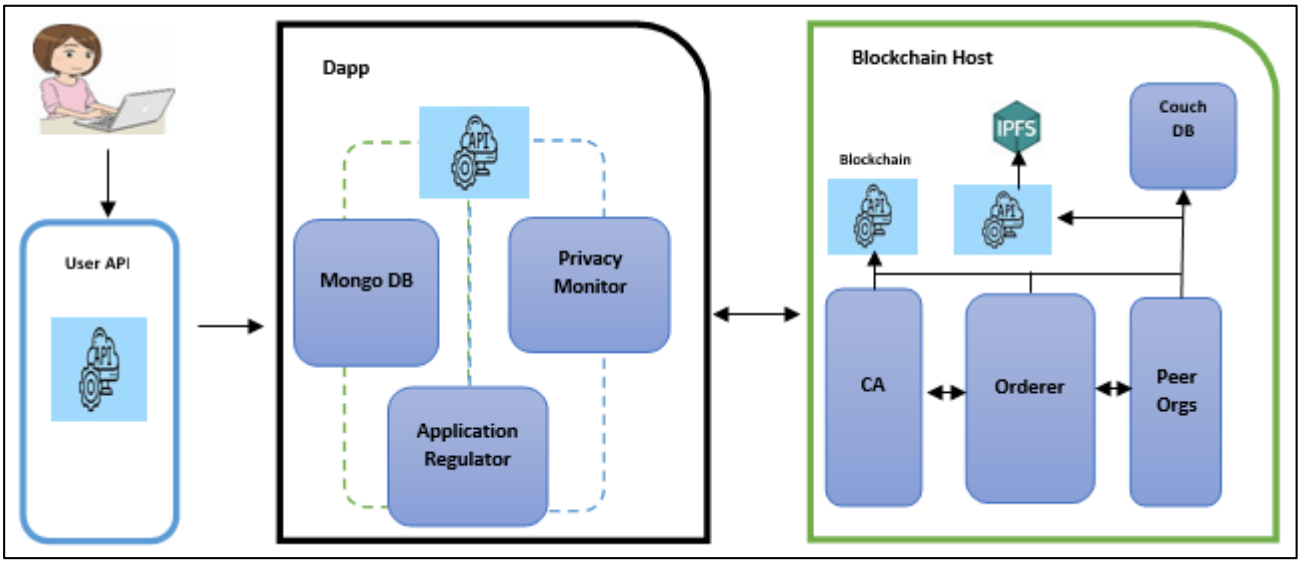

Figure 4. Proposed patient-centric health care data management.

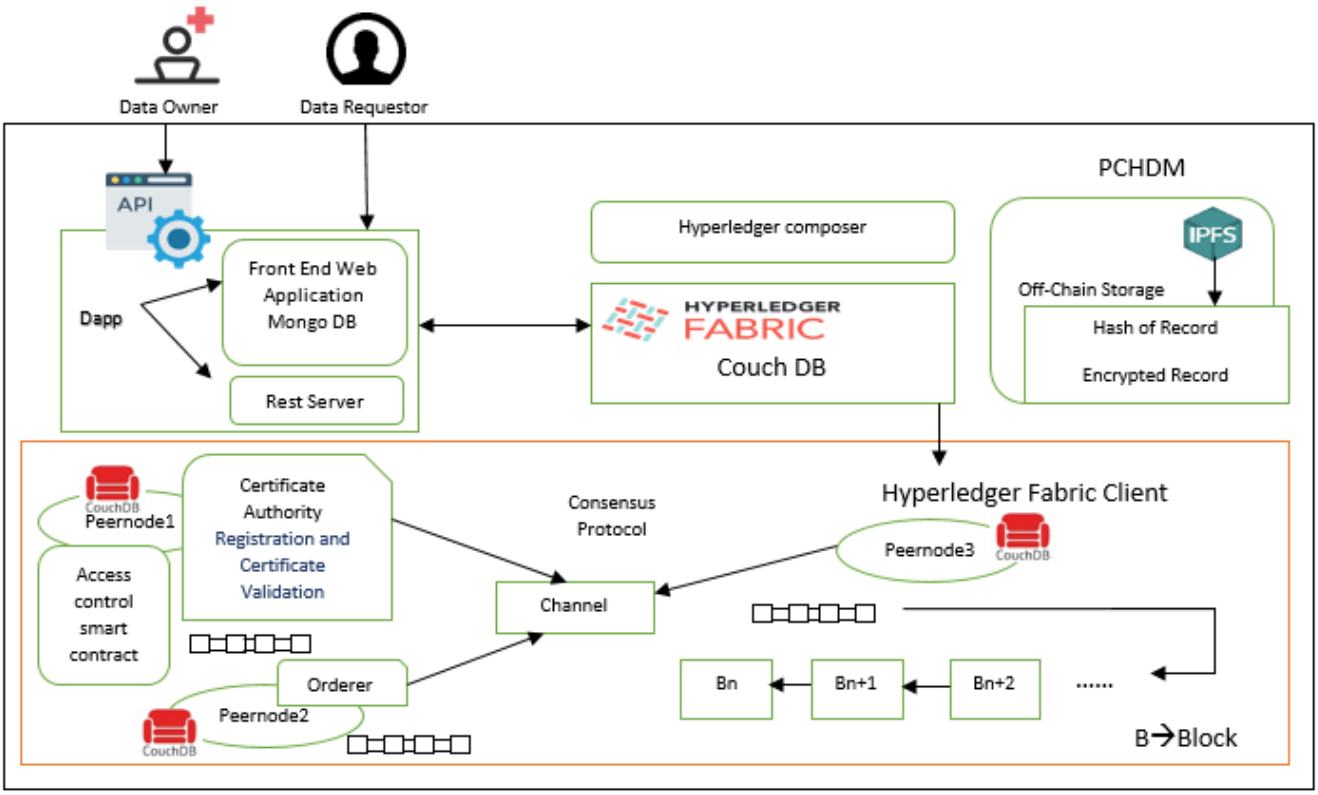

Figure 5. System architecture.

In Hyperledger Composer, a single channel (CH) facilitates communication with peers. This network generates a transaction T for our application and sends it to peernode0, peernode1, and peernode2. The chain codes are installed by the peers based on the execution of a transaction. When querying or altering the ledger, the application uses chain codes to interact with peers. The Health record chain network framework allows the histories of the changes that contributed to the framework to be viewed in the blocks as hash values in the blockchain.

In a ledger record, a block relating to the health record of a patient n is mainly comprised of that transaction's workload $WL_{tr(n)}$, hash of preceding transactions $WL_{ph(n),}$ and the current transaction hash $WL_{h(n)}$. The block workload can be calculated using $WL_{Tot(n)}$

$$WL_{Tot(n)} = WL_{tr(n)} + WL_{ph(n)} + WL_{h(n)} \quad (1)$$

The health record consists of patient's profile, Diseases Diagnosed, Address Location, Medicine, Doctor Suggestion, next Review Notes, doctors Name, Hospital ID, Scan and Test image reports.

PCHDM consists of the following stakeholders:

1. Owner of Record

Patients own their medical records. A patient will need to sign an agreement on PCHDMAC-SC in the Hyperledger blockchain and store it there. Health record chain networks allow patients to define access rights to their health data. Each PCHDMAC-SC defines this within its own context. Patient role is described in detail in Table 2.

Table 2. Role of patient in this proposed system.

Patient	Grant–Revoke–Commit–Deny Record, Read Record
	Revoke permission from Doctor/Service Providers
	Permission to a Doctor/Lab Technician to Read/Write portion of their HR
	Able to search available Doctor/Labs in network.

2. Data Uploader

Doctor may upload their medical data to Data Uploaders. Adding encrypted clinical data of the affected person to the IPFS community and confirming the preliminary transaction at the blockchain are the high responsibilities of the data uploaders. The Doctor/Lab technician role is described in Table 3.

Table 3. Role of Doctor/Lab Technician in this proposed system.

Doctor	Create/Read/Write on permissioned Healthcare system Able to search available doctor in network.
Lab Technician	Read/Write on permissioned Healthcare system Able to search available Labs in network.

3. Data Users

All those interested in obtaining clinical or medical data about patients, whether they are doctors, hospitals, insurance companies, or researchers, are considered Data Customers. PCHDMAC-SC contains role-based access control mechanisms that defines how patients can grant access privileges to data users.

3.3. Data Encryption

The integrity and confidentiality of blockchain data are ensured by cryptographic techniques. Figure 6 shows how patients and doctors interact with each other when accessing their health records. Bringing up the health records stored in the IPFS, the doctor requests permission to access them. Rather than sharing all the information about the patient, it creates a patient-centric view of the records based on the request. S_k is the Session key used to access records in a definite session, and the patient-centric view is encrypted and stored in IPFS with the session key. Doctors and patients receive encrypted patient-centric views and encryption session key S_k. Doctors can decrypt S_k and patient-centric views for an update on the health record.

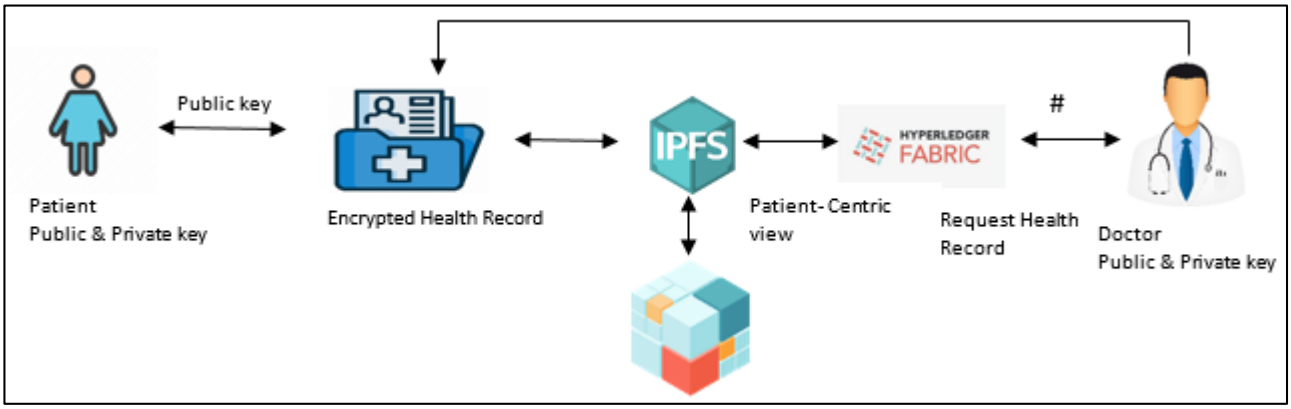

Figure 6. Patient Doctor Interaction in PCHDM.

Following the update of his record in IPFS, the patient has been notified. When a patient commits to his health record, the S_k and patient-centric view will be deleted automatically. This framework ensures the privacy of patients by preventing stakeholder access to health records without the permission of the patient. Then the hash value of the data is stored securely in Hyperledger blockchain using smart chain code running on the back end of the system. Hence, the ledger will intimate the patient about the successful addition or updating of the records.

3.4. PCHDMAC-SC

In order to obtain access to the IPFS health record of the patient, the doctor requests permission from the patients. Role-based access control permissions enable the patient to grant or deny requests for authorized users as shown in Table 4. After the approval of the patient, the doctor can create, write, and read patient records. After the write operation the patient wants to commit his record to permanent storage. In this health chain framework, other stakeholders such as researchers, pharmacists, and insurance agents can only access the patient-centric view of the health records for a specific session if their object ID matches the ownership ID and also the patient. After the approval of the patient and the doctor, the laboratory technician can update the health record. Certificate authority provides the privacy agreement, access control, and policies as a smart contract, and it is managed by the Hyperledger fabric blockchain. This approach has follows certain conditions such as

1. An access control policy defines the unique identity of the stakeholder who is allowed access.

2. The system assigns the authorized value to stakeholders, action types, resources, and environment attributes after the patient has granted access.

This system divides privacy into three levels:

Level 1: Health record is visible only to the patient.

Level 2: Health record is available to authorized stakeholders.

Level 3: During an emergency, the health record is accessible to an authorized patient caretaker.

A patient can control the privacy of his or her personal data by setting their own privacy level. In our model, the levels are configured to modify conditions during the transfer of authorizations to another authorized user prior to submission to the health record chain network.

Table 4. Role and rule-based access and authentication.

Stakeholders	Condition	Action	Operation
Patient	Authorized	Grant	Read–Grant–Revoke–Commit–Deny
Doctor	Authorized	Grant	Create–Read–Write
Pharmacist	Authorized	Grant	Patient Centric view Read
Health care Provider	Authorized	Grant	Read–Delete
Researcher	Authorized	Grant	Patient Centric view Read
Insurance Agent	Authorized	Grant	Patient Centric view Read
Referral Doctor	Authorized	Grant	Read–Write in Emergency
Lab Technician	Authorized	Grant	Read–Write

3.5. PCHDM Algorithm

The four stakeholders in our framework are Pa, D, Ph, and LT, where P is the patient, D is the Doctor, Ph Pharmacist, and LT is the lab technician. Notations used in algorithms are shown in Table 5. The n is the number of patients, doctor, pharmacist, health record, and lab technician where n = 1,2 . . . N. Hyperledger-CA issues public key certificates to n stakeholders including patients, doctors, lab technicians, and pharmacists. A key pair will be generated for all the stakeholders. The public and private key of patient and doctor are $Papk_n$, $Paprk_n$, Dpk_n, $Dprk_n$. As shown in Algorithm 1, the patient Pa_n grants access to their health record HR_n to the doctor, D_n, based on PCHDMAC-SC. Hence the system generates a patient centric view Pa_{cvn} of the health record HR_n. Based on the Doctor D_n request, the attribute-based data is retrieved from the Pa_{cvn} instead of sharing whole patient health records. A patient-centric view Pa_{cvn} of a specific health record allows users to see and modify only the data they need. In other words, the patient-centric view is a subset of the health record. In addition to this, the system generates a session key S_k shared by both doctor and patient within a particular session.

Table 5. Notation explanation.

Notation	Definition
Pa_n	Patient
D_n	Doctor
Ph_n	Pharmacist
LT_n	Lab Technician
HR_n	Health Record
$Papk_n$	Patient public key
$Paprk_n$	Patient private key
Dpk_n	Doctor public key
$Dprk_n$	Doctor private key
S_k	Session key
Pa_{cvn}	Patient-Centric View
UPa_{cvn}	Update Patient-Centric view
HR_n_hash	Hash

An encrypted session key such as Encrypted ($Papk_n$ (S_k)), Encrypted (Dpk_n (S_k)) is generated for the patient and doctor using their public key. The session key S_k is also encrypted with Pa_{cvn} which can be sent to doctors. To update the health record HR_n, Algorithm 1 calls the Create_Update () function of Algorithm 2. The updates are uploaded into the update patient-centric view UPa_{cvn} after decryption of doctor session key and patient-centric view session key. As soon as the patient system updates, the encrypted health record HR_n is decrypted by using the patient private key retrieved by decrypting the encrypted private key using the password of the patient and adds the Encrypted $UPa_{cvn.}$ Lastly, the patient commits the updates to the health record HR_n and stores in IPFS. The session key and Pa_{cvn} will becomes expired once the patient commits the health record $HR_{n.}$ The IPFS generates health record hash value HR_n_hash which is stored in blocks of Hyperledger blockchain.

Algorithm 1 SystemFunction ()

Creating and updating health record in Hyperledger blockchain
Input: A Doctor D_n with their Dpk_n and $Dprk_n$ with session key S_k of Health Record HR_n, A Patient Pa_n with their $Papk_n$ and $Paprk_n$ with session key S_k of Health Record HR_n
Output: Boolean (Success or Failure)

1. Procedure of storing and updating health records
2. For each user u having access permission to Health Record
3. Check PCHDMAC-SC
4. If (permission==" GRANT" && role==" DOCTOR") then
5. Create patient centric view Pa_{cvn} of HR_n in IPFS
6. $Pa_{cvn} \rightarrow$ Decryption (Encryption (HR_n))
7. Create S_k
8. send Encrypted ($Papk_n$ (S_k), Dpk_n(S_k), Pa_{cvn}(S_k)) to Pa_n, D_n and Pa_{cvn}
9. create_Update ()
10. $HR_n \rightarrow$ [(Decryption $Paprk_n$ (Encrypted $Papk_n$ (HR_n)) + Encryption (UPa_{cvn})]
11. $Pa_n \rightarrow$ Commit (IPFS (HR_n))
12. IPFS $\rightarrow$ HR_n_hash
13. HR_n_hash $\rightarrow$ HyperlegerFabric Blocks
14. Return True
15. Else
16. Permission=Deny
17. Return False
18. End if
19. End For ()
20. End procedure

Algorithm 2 Create_Update ()

Create and Update the Patient centric view of the Health Record
Input: A Doctor D_n with their Dpk_n with session key S_k
Output: Storage of health record

1. Procedure Doctor Dpk_n
2. For each Doctor having Dpk_n with session key S_k
3. $D_n \leftarrow$ Decrypt (Dpk_n (S_k))
4. $D_n \leftarrow$ Decrypt (Pa_{cvn} (S_k))
5. $Pa_{cvn} \rightarrow UPa_{cvn}$
6. IPFS Storage Encrypt (UPa_{cvn} (S_k))
7. End For
8. End procedure

4. Implementation of PCHDM Protocol

The proposed framework has two parts in terms of the development environment. This framework is built with network entities and smart contracts, IPFS storage is used, and smart contracts govern every transaction. This system consists of separate back-end and front-end development environments. Implementations and experiments were carried out on a Core i7-8765u processor and 8 GB of memory. We used a Linux Foundation project, Hyperledger Fabric 1.4v, for our research. The Fabric SDK requires Java and Node as prerequisites for client development. REST APIs made it possible to visualize back-end business logic, such as user requests, assets, searches, and transaction APIs. Front-end development was carried out using HTML5, CSS3 and JavaScript. To make our web application more efficient and user-friendly, we incorporate third-party frameworks such as jQuery and Bootstrap. Front-end programming is performed with a database, and back-end programming is performed with REST API servers. Clients use web applications to perform actions that trigger HTTP methods such as POST, GET, PUT, and DELETE. These methods cause the web service to respond with HTTP responses according to the requests

made by the client. Table 6 shows the machine configurations and main components used for the simulation environment.

Table 6. Environment configuration.

Component	Configuration
System under Test	Hyperledger Fabric 1.4v
CPU and Memory	Core i7-8765u and 8 GB
On-chain database	Couch DB
Off-chain database	IPFS
Test language	Node.js, Java

The protocol has been implemented in four steps.

4.1. Add Users

A step-by-step implementation of adding users to the network is shown in Figure 7. After the registration of stakeholders, the Fabric Certificate authority generates the role-based unique ID for the registered stakeholders.

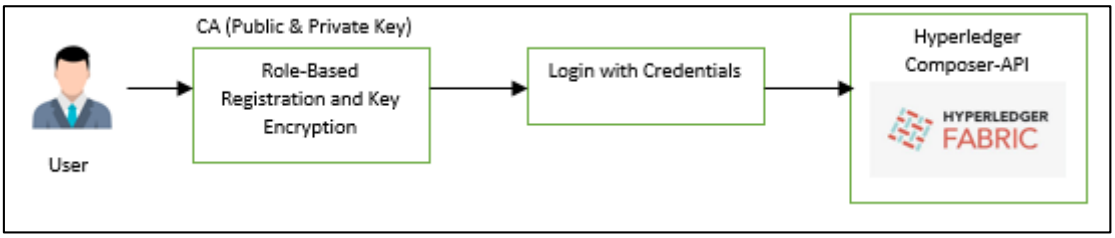

Figure 7. Add users.

4.2. Add Records and Update Records

Health records can be created by the doctor after the patient grants permission and stored in IPFS with encryption. The hash value is preserved via Hyperledger blockchain. The patient wants to grant access to the doctor's records in order to revise them. A temporary view called the patient-centric view of the health record is generated. The doctor may update the patient-centric view of the health record and, after the patient agrees, update the existing record permanently in both the IPFS and the Health record chain. The session key will expire after that, which means that doctors will no longer have access to patient records that contain confidential information. The step-by-step implementation of adding health records and updating records in the network is shown in Figure 8.

4.3. Assuring Authorized Users Have Access

Access permissions on the medical record are granted by the patient to stakeholders within a restricted setting, allowing them to read, write, and deny access as needed. The patient has complete control and ownership to grant read, write, and deny access permissions to stakeholders on the medical record, thereby maintaining restrictive access control. In addition, patients can authorize access to health records according to the role type and permission type of authenticated users approved by consensus. The patient can also deny specific doctors access to their medical records, and in that case, the records cannot be released to other doctors. When users interact with the system, smart contracts will identify requests, validate requests, update records, and grant access permissions. The step-by-step implementation of role-based access control and permission is shown in Figure 9.

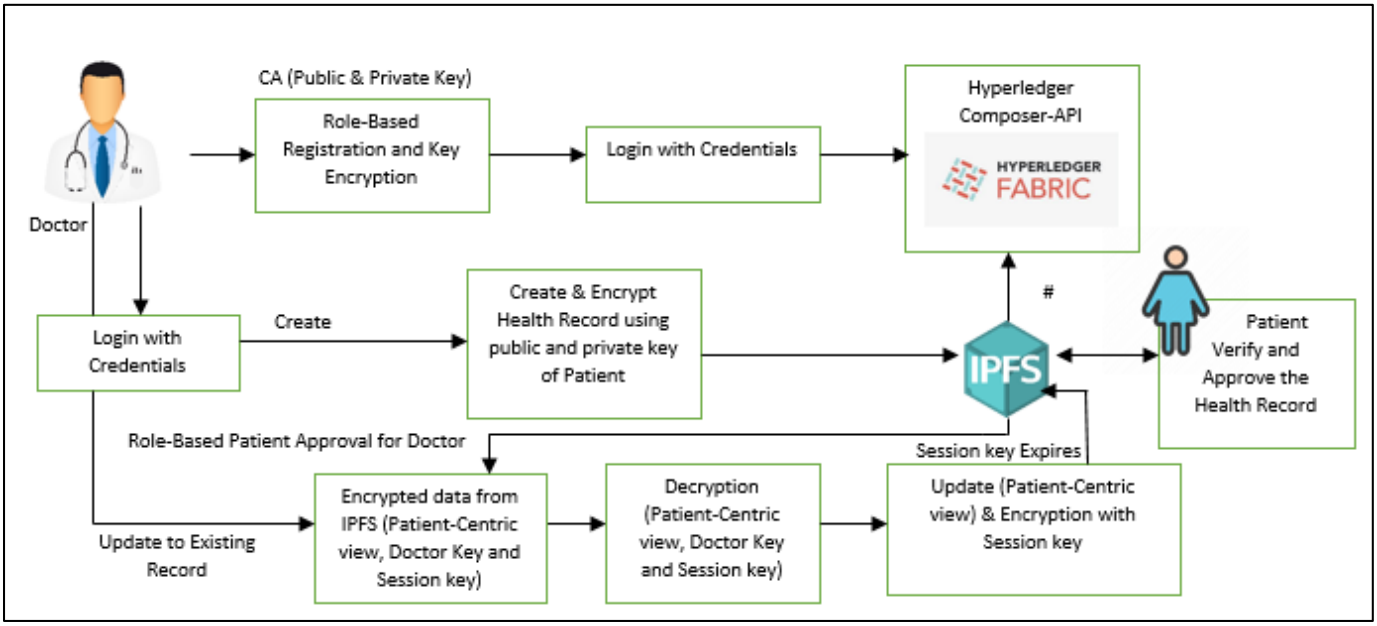

Figure 8. Add records and update records.

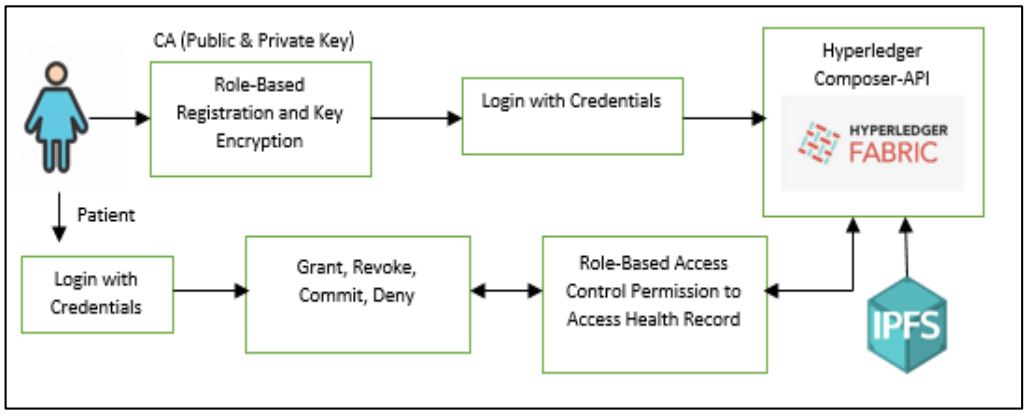

Figure 9. Assuring authorized users have access.

4.4. Records Retrieval

For all stakeholders to read a patient's record, the patient wants to grant access that retrieves partial attribute-based information about the patient, retrieved from IPFS using the hash value in the Health record chain network. The step-by-step implementation of the view record is shown in Figure 10.

4.5. Framework Implementation

The Health Chain Network Transaction Framework consists of smart contracts, chain code [26], IPFS storage, and network entities. In the Health Chain network user sign up system, doctors, pharmacists, receptionists, and other health care providers can register using their respective roles. Upon registration, the fabric certificate authority is generated. The certificate authority contains encrypted privacy policies as shown in Figure 11. Using their email address and password, the user can use their user type to sign in after registration. Receptionists can accept or reject appointments booked by the stakeholder using patient IDs and update the stakeholders based on the patient's information. The patient consults the doctor after the appointment has been approved by the receptionist, and the doctor creates the patient's medical record. IPFS allows doctors to upload medical notes or diagnosis results.

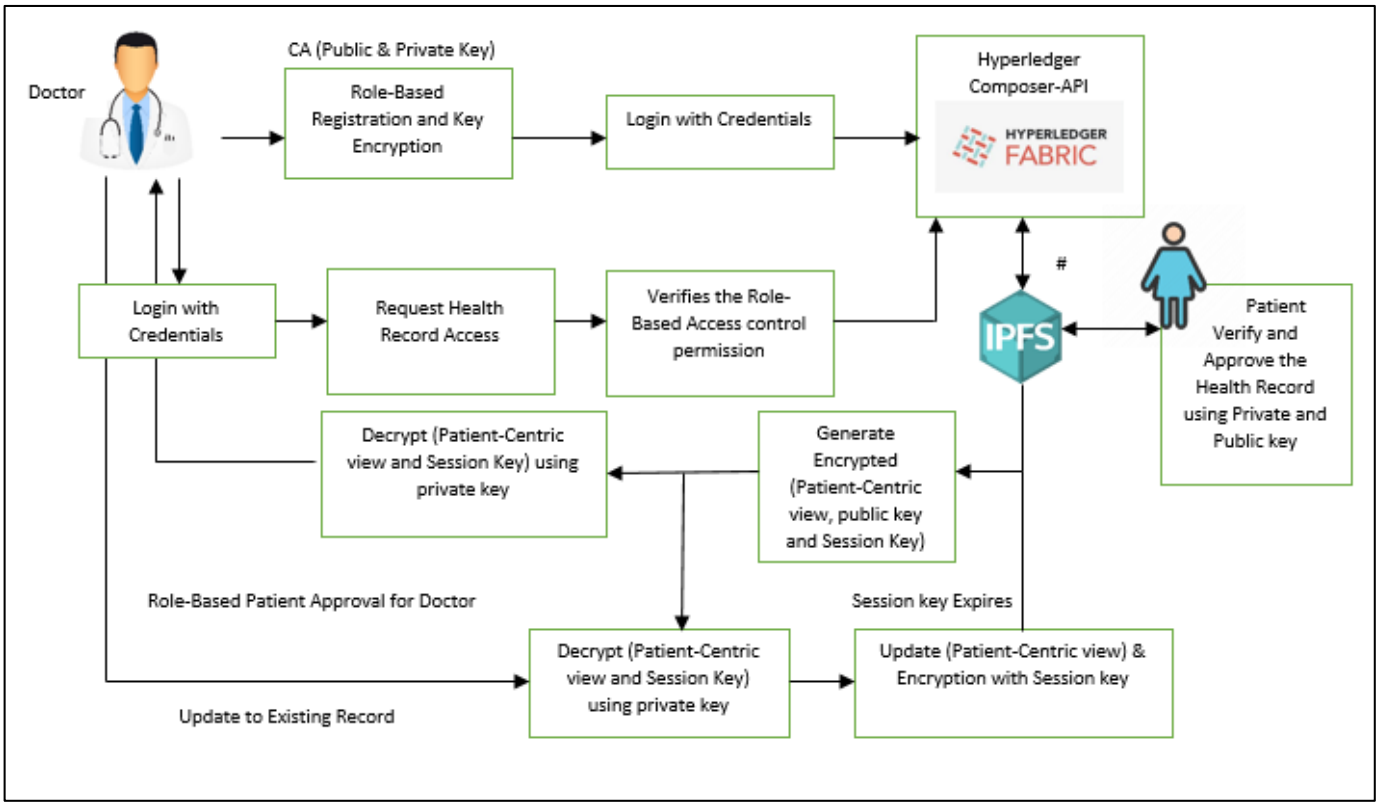

Figure 10. Record retrieval.

```
-----BEGIN CERTIFICATE-----
MIICKjCCAdCgAwIBAgIRAIDOrJjzkFRKVnKtr34xis4wCgYIKoZIzj0EAwIwXzEL
MAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNhbiBG
cmFuY2lzY28xDzANBgNVBAoTBmR1dm9yZzESMBAGA1UEAxMJY2EuZGV2b3JnMB4X
DTIwMDMzMDE0MjAwMFoXDTMwMDMyODE0MjAwMFowXzELMAkGA1UEBhMCVVMxEzAR
BgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNhbiBGcmFuY2lzY28xDzANBgNV
BAoTBmR1dm9yZzESMBAGA1UEAxMJY2EuZGV2b3JnMFkwEwYHKoZIzj0CAQYIKoZI
zj0DAQcDQgAEeS4405v1i685FbdO2hsa3xKTuTFapg5EiGkl1Ky9vsnUH565A8Xx
3v4EKwga5U22h3cktDuWMvKTSzCMJp9WfKNtMGswDgYDVR0PAQH/BAQDAgGmMB0G
A1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcDATAPBgNVHRMBAf8EBTADAQH/MCkG
A1UdDgQiBCCFzAeHiuYcdBbQWJ1axd9qV3oFchcHLRASjtD458IiwDAKBggqhkjO
PQQDAgNIADBFAiAIiHKHTnlDy7GZjwQHBecItSwFThlPIACBsx+kiLmv5AIhAKIc
15aO9siLLoDdHNi5RkIO/j5H/RrM+QuvfuVwYBbd
-----END CERTIFICATE-----
```

Figure 11. Fabric certificate generation for each stakeholder.

Our proposed business network consists of stakeholders, Assets and Transaction as shown in Figure 12.

The prototype has undergone a number of tests in order to validate its functionality and evaluate its performance. An assessment of the health chain framework systems is realized through the application of four case studies that illustrate efficiency, scalability, storage, and security.

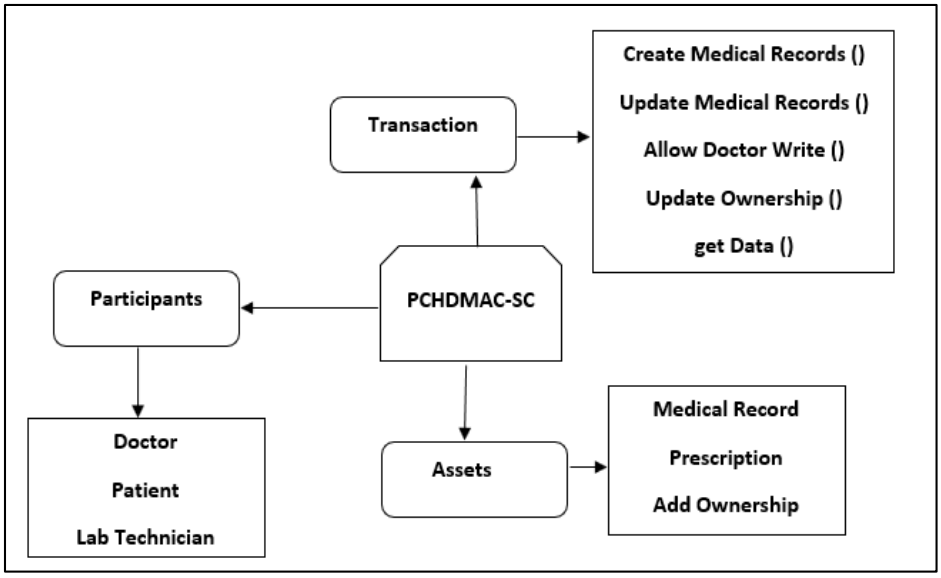

Figure 12. Business Network.

4.5.1. Efficient Storage

A few cases were tested to determine whether interplanetary database can adequately store health records such as:

1. Health records can be uploaded by doctors.
2. Health records can be viewed by a doctor with permission.
3. Health records can be viewed by patients.
4. Patients and doctors are able to identify health records based on their identifiers.
5. Retrieval of encrypted records efficiently.

As a result, doctors can encrypt their updated records for storage in IPFS with their session keys, and patients can decrypt them using their session keys.

4.5.2. Security

A few cases were tested to determine to verify the security such as

1. Encrypted User password.
2. Encrypted health records stored in IPFS.
3. A unique hash value is assigned to each health record.
4. All stakeholders are provided with public and private keys.
5. Assignment and expiration of session keys.

This test verifies successful generation of an encrypted health record before the data are stored in IPFS.

4.5.3. Privacy

The purpose of this test is to verify that stakeholders have been granted and have been denied access to health records in the system depending on their role. Some test cases are used to test the access control for health records such as:

1. Stakeholders can view their respective homepage based on their role.
2. Role-based access control.
3. Health records are read with an assigned session key.

Therefore, the system is capable of allocating access rights according to their levels and roles. Example code snippets for access control transactions are shown in Figures 13 and 14.

```
async getReportData(ctx, args){
      args = JSON.parse(args);
      let myAssetId = args.reportId;
      console.log("Inside getReportData");
      console.log(myAssetId);
      const exists = await this.myAssetExists(ctx, myAssetId);
      if (!exists ) {
          throw new Error({"error":`The report ${myAssetId} does not exist`});
      }
      const buffer = await ctx.stub.getState(myAssetId);
      const asset = JSON.parse(buffer.toString());
      return asset;
  }
```

Figure 13. Sample code snippet of transaction in Health record chain network.

```
rule PatientsCanOnlyAccessThemselves {
    description: "Patients can only read and update their own participant data"
    participant(p1): "nz.ac.auckland.Patient"
    operation: READ, UPDATE
    resource(p2): "nz.ac.auckland.Patient"
    condition: (p2.getIdentifier() == p1.getIdentifier())
    action: ALLOW
}

rule PatientReadTransactionAccess {
    description: "Patients should have only read access to the transaction registry"
    participant: "nz.ac.auckland.Patient"
    operation: READ
    resource: "org.hyperledger.composer.system.TransactionRegistry"
    action: ALLOW
}

rule PatientReadAssetAccess {
    description: "Patients should have only read access to the asset registry"
    participant: "nz.ac.auckland.Patient"
    operation: READ
    resource: "org.hyperledger.composer.system.AssetRegistry"
    action: ALLOW
}

rule PatientReadParticipantAccess {
    description: "Patients should have only read access to the participant registry"
    participant: "nz.ac.auckland.Patient"
    operation: READ
    resource: "org.hyperledger.composer.system.ParticipantRegistry"
    action: ALLOW
}
```

Figure 14. Sample code snippet of PCHDMAC-SC using access control language.

4.5.4. Data Scalability

The scalability of the proposed system is tested with a few test cases, such as:

1. Storing of large and small file size health records.
2. Computation of throughput and latency using Hyperledger caliper benchmarks.

We have verified that our proposed system is capable of handling large data sets and low latency through the above test cases using the Hyperledger caliper benchmarks.

4.6. The Comparative Analysis of Existing and Proposed PCHDM Model

In Table 7, we compare existing and proposed patient-centric health storage models with an emphasis on scalability, privacy, confidentiality, integrity, and security. Using blockchain technology, the author has explained recent healthcare management practices and their consequences in [27,28]. The existing frameworks considered are [8,13–16,29]. Each block contains a hash of a health record which will change if any changes are made to the record. As a result, tampering with the ledger is computationally difficult, ensuring that the medical record cannot be altered. The access control rules and levels prevent stakeholders from accessing health records without the patient's knowledge.

Table 7. Comparison of proposed and existing model.

Models	Ease of Scaling	Access Control	Confidential Information	Data Integrity	Data Security	Patient–User Preference
[8]	×	×	✓	✓	✓	×
[13]	×	✓	✓	✓	✓	×
[14]	✓	×	×	✓	✓	×
[15]	×	×	×	×	✓	×
[16]	×	✓	✓	✓	✓	×
[29]	×	×	✓	✓	✓	×
PCHDM	✓	✓	✓	✓	✓	✓

5. Results and Performance Evaluation

A dataset was gathered from US health records on a website called Kaggle. The dataset consisted of images and text of variable sizes. It was used to test the existing health records. Hyperledger Caliper was used to benchmark a blockchain-based application [30]. Caliper is designed to benchmark the performance of Hyperledger using many different metrics such as throughput, latency, and success rate (average, minimum, maximum, and percentile). Furthermore, it indicates how resources such as CPU memory will be allocated to the system. The result is calculated and generated from Hyperledger caliper reports benchmarks with the following metrics:

1. Success and fail rate.
2. Transaction/read throughput.
3. Transaction/read latency (minimum, maximum, average, percentile).
4. Resource utilization (CPU, memory, network (traffic in and traffic out)).

The scalability of the proposed application is extremely important. The proposed framework was therefore tested against ordering services against varying numbers of peers. Initially the performance evaluation of this network was carried out under one organization and three peer nodes. The benchmark report helped to make a comparison of peer among nodes and finally the system efficiency was determined.

This first experiment employed a Hyperledger fabric blockchain framework to calculate transaction latency. The latency of a transaction reflects how long it takes to commit. It is a distributed parameter across nodes in the network. If there are p number of nodes in the health chain network, TR_{Lap} is the transaction latency, TR_{Ctp} is the confirmation time in the network nodes, and TR_{Stp} is the transaction submit time in seconds then transaction latency is given as

$$TR_{Lap} = TR_{Ctp} - TR_{Stp} \quad (2)$$

The network ledger has been updated through the use of eight groups of transactions in organization 1 peernode1 ranging from 5, 15, 25, 30, 35, 45, 50 and 55 as shown in Figure 15. In this configuration, the first five transactions across the network were committed in 104 s, and the final 55 transactions took 161 s on average. A range of 55 to 400 transactions were then added to the experimental result to determine transaction time.

Figure 16 shows that when 400 transactions were committed on three peer nodes of an organization, the average time was 420 s. As a result, the transaction latency remained average when the number of transactions increased. The comparison of commit transaction latency with three stage peernodes such as 1 Organization 1 peernode, 1 Organization 2 peernode, 1 organization 3 peernode. From Figure 17, it shows a slight increase in latency with an increase in the number of peer nodes. The next set of experiments was conducted for evaluating throughput. Transaction throughput is the number of transactions which were valid and committed from the total executed transactions as

$$TR_{tpp} = TR_{vcp}/TR_{totalp} \quad (3)$$

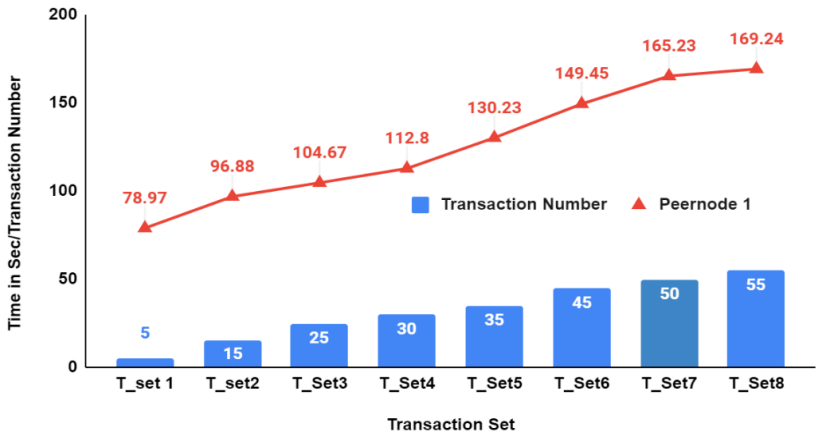

Figure 15. Latency of organization peernodes for sample set of Transactions.

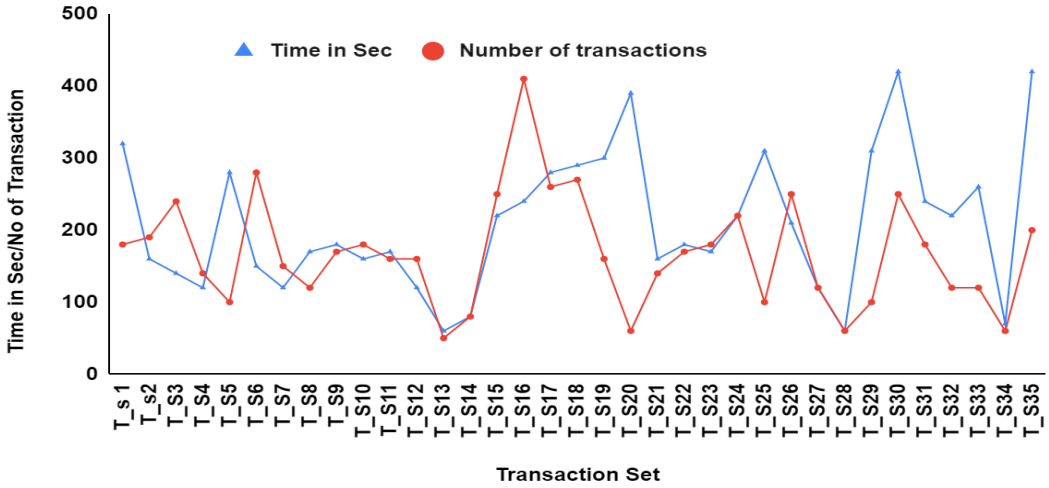

Figure 16. Simulation of transaction latency with a greater number of transactions sets.

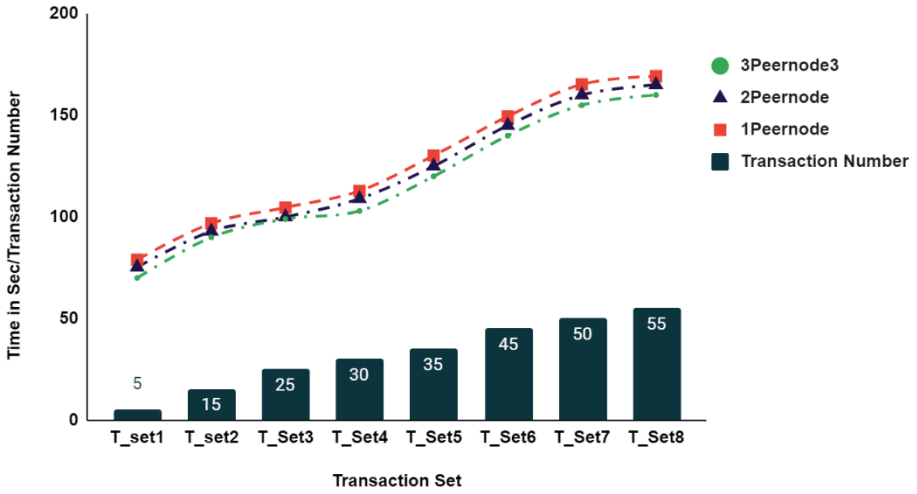

Figure 17. Latency comparison of organization 3 peer nodes.

Figure 18 shows transaction throughput of the proposed system using experimentation of eight sets of transactions. In this, the first five transactions have taken 79 s to commit into the network. Hence the valid transaction in system under test is three in the network. Likewise, the last transaction 55 takes 110 s to commit in the network with 28 valid transaction per minute. This same experiment is repeated for 1 organization 1 peernode, 1 organization 2 peernodes, 1 organization 3 peernodes. From the Figure 19 the successful transaction of all nodes in which 1 organization 1 peernode throughput is slightly higher than organization 2 peernodes and 1 organization 3 peernodes.

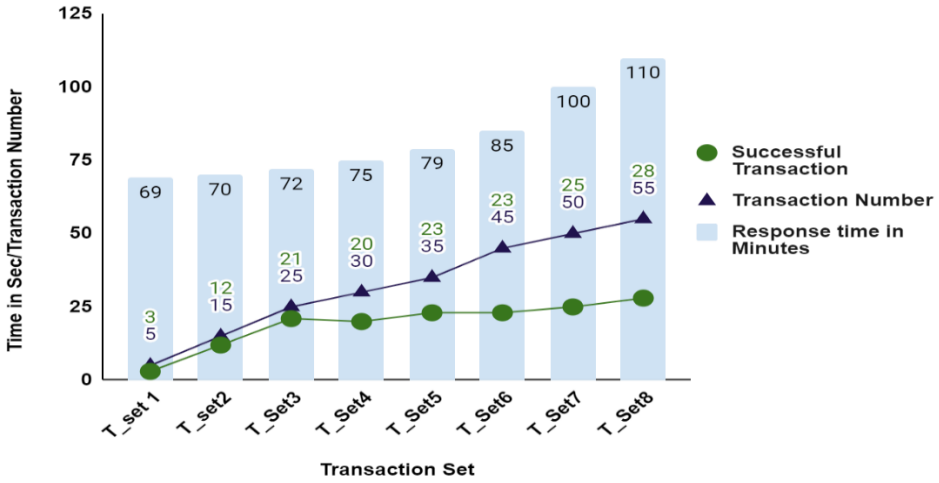

Figure 18. Throughput of Organization Peernode 1.

When assets are successfully loaded and written to a database, the asset latency is measured. In a blockchain network with P nodes, AS_Lp represents the Asset Latency. The response time TR_Resp is measured in milliseconds and the asset submit time TR_AS_Subp is measured in milliseconds.

$$AS\_Lp = TR\_Resp - TR\_AS\_Subp \quad (4)$$

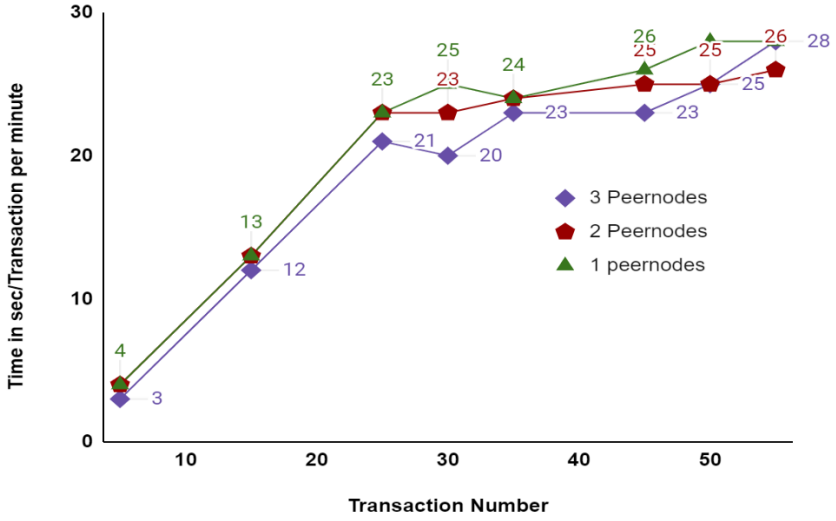

Figure 19. Throughput comparative analysis with 1 organization 3 peernodes.

Figure 20 shows the asset time to commit in the blockchain form sample set of transactions. As a result, this system can process a large dataset with low latency. In order to examine asset latency variability, the experiment was extended to include user numbers between 20 and 120, and data sizes between 200 k bytes and 20,574 k bytes. According to Figure 21, the average latency for updating assets in the ledger was 2.9 s.

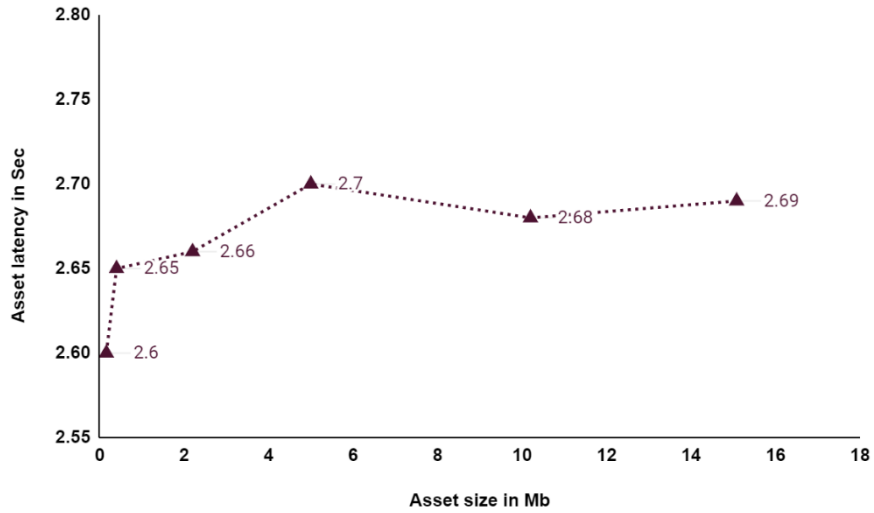

Figure 20. Average time taken for asset submission and response.

Considering configuration of the system, even with a 20-user increase to 120 users, the efficiency would still be higher and asset size would increase in the system, but the process of updating assets across the network would take slightly more time. As a result, this method can process data sets of large sizes with low latency. Further analysis such as upload of 140 mb of data and downloading of data in IPFS by the five concurrent users that take 60 sec to commit this upload and download operation. The average upload and download time of the medical image data in IPFS for the sample 100 MB is 34 and 43 s, as depicted in Figure 22. Then the average upload and download time of the sample 100 mb health record document file is 39.3 and 53.5 s, respectively, as depicted in Figure 23. From Figures 22 and 23, the average upload and download latency is maintained for all the transaction sets without any interruption. This shows the scalability of the system has improved using this proposed framework. Table 8 shows the Hyperledger caliper report

of resource utilization. Throughout the experiments, resource utilization remained steady. Hence it will not affect the system under test.

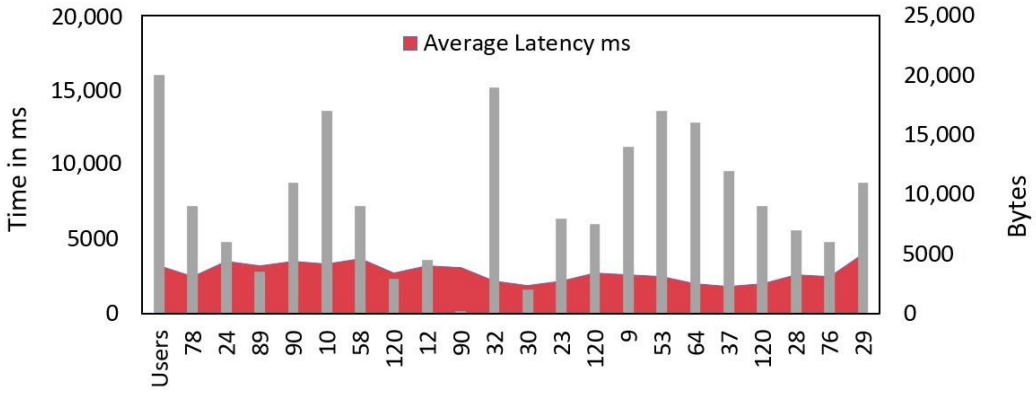

Figure 21. Simulation of assets latency with a greater number of users.

Figure 22. Average upload and download time of patient lab result as image using PCHDM.

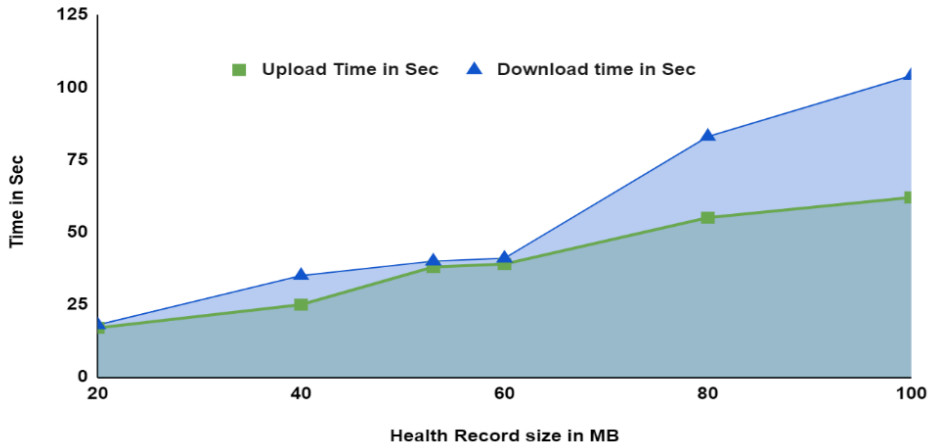

Figure 23. Average upload and download time of patient health record using PCHDM.

Table 8. An analysis of the proposed PCHDM system's resource utilization.

Type	Name	CPU% (Max)	CPU% (Avg)	Memory (Max) [MB]	Memory (Avg) [MB]	Traffic In [MB]	Traffic Out [MB]
Docker	1OrgLocalFabric-Org1Peer1-Patient centric health record-0.0.1	107.65	18.76	149.56	142.25	10.06	11.90
	1OrgLocalFabric_orderer.example.com	41.60	5.97	126.08	102.51	0.06	27.18
	1OrgLocalFabric_ca.orderer.example.com	0.30	0.01	7.67	7.24	0.00	0.00
	1OrgLocalFabric_peer0.org1.example.com	83.41	22.04	650.45	625.34	43.19	34.78
	1OrgLocalFabric_couchdb0.org1.example.com	72.34	15.08	149	139	2.09	3.96
	1OrgLocalFabric_ca.org1.example.com	0.09	0.00	5.90	5.50	0.00	0.00

6. Conclusions

In this paper, we describe the design, implementation, and evaluation of PCHDM, an end-to-end secure Health record chain network architecture based on PCHDMAC-Smart Contracts. By leveraging Health record chain networks, IPFS, and Smart contract, this framework ensures the safety of health records between stakeholders. In addition, it features an innovative access control scheme that adheres to compliance with privacy laws and patients' privacy levels. As a result of the analysis, the implemented system appears to be efficient and satisfies many security requirements. A high level of privacy, security, confidentiality and scalability can be achieved.

There are some limitations to this research that need to be addressed in future research. Multi-blockchain systems require a tremendous number of resources to be implemented. In the future, we will extend the framework integrated with Non-Fungible Tokens (NFT) to share audio and video as NFT data with the stakeholders. NFTs will allow patients to choose whose data they want to share and sell, as well as track how and by whom that data is being used. Patient data transactions and monitoring can be supported by trusted NFT management services.

Author Contributions: Conceptualization: V.M.; methodology: V.M.; validation: S.A. and P.M.; formal analysis: P.M., O.I.K. and S.A.; investigation: P.M. and S.A.; resources: V.M.; data Curation: Y.A. and V.M.; writing—original draft preparation: V.M.; writing—review and editing: V.M. and Y.A.; visualization: O.I.K. and S.A.; supervision: P.M. and Y.A.; project administration: Y.A. and S.A.; funding acquisition: S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Taif University, TURSP-2020/313.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We deeply acknowledge Taif University for supporting this study through Taif University Researchers Supporting Project Number (TURSP-2020/313), Taif University, Taif, Saudi Arabia.

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

References

1. Heart, T.; Ben-Assuli, O.; Shabtai, I. A Review of PHR, EMR and EHR Integration: A More Personalized Healthcare and Public Health Policy. *Health Policy Technol.* **2017**, *6*, 20–25. [CrossRef]
2. Idrees, S.; Nowostawski, M.; Jameel, R.; Mourya, A. Security Aspects of Blockchain Technology Intended for Industrial Applications. *Electronics* **2021**, *10*, 951. [CrossRef]
3. Sharma, A.; Tomar, R.S.; Chilamkurti, N.; Kim, B.G. Blockchain Based Smart Con- tracts for Internet of Medical Things in e-Healthcare. *Electronics* **2020**, *9*, 1609. [CrossRef]
4. Seh, A.H.; Zarour, M.; Alenezi, M.; Sarkar, A.K.; Agrawal, A.; Khan, A.R. Healthcare Data Breaches: Insights and Implications. *Healthcare* **2020**, *8*, 133. [CrossRef] [PubMed]

5. 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), IEEE, Vienna, Austria, 22–24 August 2016; pp. 25–30.
6. Ivan, D. Moving toward a blockchain-based method for the secure storage of patient records. In Proceedings of the ONC/NIST Use of Blockchain for Healthcare and Research Workshop, ONC/NIST, Gaithersburg, MD, USA, 4 August 2016.
7. Dannen, C. *Introducing Ethereum and Solidity*; Springer: Berlin/Heidelberg, Germany, 2017.
8. Shen, B.; Guo, J.; Yang, Y. Medchain: Efficient healthcare data sharing via blockchain. *Appl. Sci.* **2019**, *9*, 1207. [CrossRef]
9. Jamil, F.; Ahmad, S.; Iqbal, N.; Kim, D.-H. Towards a remote monitoring of patient vital signs based on iot-based blockchain integrity management platforms in smart hospitals. *Sensors* **2020**, *20*, 2195. [CrossRef]
10. Margheri, A.; Masi, M.; Miladi, A.; Sassone, V.; Rosenzweig, J. Decentralised provenance for healthcare data. *Int. J. Med. Inform.* **2020**, *141*, 104197. [CrossRef] [PubMed]
11. Roehrs, A.; da Costa, C.A.; da Rosa Righi, R.; da Silva, V.F.; Goldim, J.R.; Schmidt, D.C. Analyzing the performance of a blockchain-based personal health record implementation. *J. Biomed. Inform.* **2019**, *92*, 103140. [CrossRef] [PubMed]
12. Jha, N.; Prashar, D.; Khalaf, O.I.; Alotaibi, Y.; Alsufyani, A.; Alghamdi, S. Blockchain Based Crop Insurance: A Decentralized Insurance System for Modernization of Indian Farmers. *Sustainability* **2021**, *13*, 8921. [CrossRef]
13. Dwivedi, A.D.; Srivastava, G.; Dhar, S.; Singh, R. A decentralized privacy- preserving healthcare blockchain for iot. *Sensors* **2019**, *19*, 326. [CrossRef] [PubMed]
14. Rajput, A.; Li, Q.; Ahvanooey, M. A blockchain-based secret-data sharing framework for personal health records (2021) in emergency condition. *Healthcare* **2021**, *9*, 206. [CrossRef] [PubMed]
15. Jagadeesh, R.; Mahantesh, K. Blockchain-based knapsack system for security and privacy preserving to medical data (2021) in SN COMPUT. *Scientifur* **2021**, *2*, 245.
16. Egala, B.S.; Pradhan, A.K.; Badarla, V.; Mohanty, S.P. Fortified-chain: A blockchain- based framework for security and privacy-assured internet of medical things with effective access control. *IEEE Internet Things J.* **2021**, *8*, 11717–11731. [CrossRef]
17. Alsufyani, A.; Alotaibi, Y.; Almagrabi, A.O.; Alghamdi, S.A.; Alsufyani, N. Optimized intelligent data management framework for a cyber-physical system for computational applications. *Complex. Intell. Syst.* **2021**, 1–13. [CrossRef]
18. Singh, P.; Masud, M.; Hossain, M.S.; Kaur, A. Blockchain and homomorphic encryption- based privacy-preserving data aggregation model in smart grid. *Comput. Electr. Eng.* **2021**, *93*, 107209. [CrossRef]
19. Peng, C.; He, D.; Chen, J.; Kumar, N.; Khan, M.K. EPRT: An Efficient Privacy-Preserving Medical Service Recommendation and Trust Discovery Scheme for eHealth System. *ACM Trans. Internet Technol.* **2021**, *21*, 1–24. [CrossRef]
20. Piao, Y.; Ye, K.; Cui, X. A Data Sharing Scheme for GDPR-Compliance Based on Consortium Blockchain. *Future Internet* **2021**, *13*, 217. [CrossRef]
21. Tung, J.; Nambudiri, V. Beyond Bitcoin: Potential Applications of Blockchain Technology in Dermatology. *Br. J. Dermatol.* **2018**, *179*, 1013–1014. [CrossRef]
22. Tiwari, A.; Batra, U. IPFS enabled blockchain for smart cities. *Int. J. Inf. Tecnol.* **2021**, *13*, 201–211. [CrossRef]
23. How IBM Blockchain Can Solve the Problem of Continuity of Care-an Effort from LMS India. Available online: https://medium.com/@lmsin/ehr-solutions-67e199b8f596 (accessed on 29 November 2018).
24. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, ACM, Porto, Portugal, 23–26 April 2018; p. 30.
25. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, 21260. Available online: http://bitcoin.org/bitcoin.pdf (accessed on 18 October 2021).
26. Foschini, L.; Gavagna, A.; Martuscelli, G.; Montanari, R. Hyperledger Fabric Blockchain: Chaincode Performance Analysis. In Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [CrossRef]
27. Kumar, S.; Bharti, A.K.; Amin, R. Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions. *Secur. Privacy* **2021**, *4*, e162. [CrossRef]
28. Al-asmari, A.M.; Aloufi, R.I.; Alotaibi, Y. A Review of Concepts, Advantages and Pitfalls of Healthcare Applications in Blockchain Technology. *Int. J. Comput. Sci. Netw. Secur.* **2021**, *21*, 199–210.
29. Wang, H.; Song, Y. Secure cloud-based ehr system using attribute-based cryptosystem and blockchain. *J. Med. Syst.* **2018**, *42*, 152. [CrossRef] [PubMed]
30. Sukhwani, H.; Wang, N.; Trivedi, K.S.; Rindos, A. Performance Modeling of Hyperledger Fabric (Permissioned Blockchain Network). In Proceedings of the 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 1–3 November 2018; pp. 1–8.