applied sciences

MDPI

Article

Generic Patient-Centered Blockchain-Based EHR Management System

Alaa Haddad [1], Mohamed Hadi Habaebi [1,*], Fakher Eldin M. Suliman [2], Elfatih A. A. Elsheikh [2], Md Rafiqul Islam [1] and Suriza Ahmad Zabidi [1]

1 IoT & Wireless Communication Protocols Laboratory, Department of Electrical & Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia
2 Department of Electrical Engineering, College of Engineering, King Khalid University, Abha 61421, Saudi Arabia
* Correspondence: habaebi@iium.edu.my

Abstract: Accessing healthcare services by several stakeholders for diagnosis and treatment has become quite prevalent owing to the improvement in the industry and high levels of patient mobility. Due to the confidentiality and high sensitivity of electronic healthcare records (EHR), the majority of EHR data sharing is still conducted via fax or mail because of the lack of systematic infrastructure support for secure and reliable health data transfer, delaying the process of patient care. As a result, it is critically essential to provide a framework that allows for the efficient exchange and storage of large amounts of medical data in a secure setting. The objective of this research is to develop a Patient-Centered Blockchain-Based EHR Management (PCEHRM) system that allows patients to manage their healthcare records across multiple stakeholders and to facilitate patient privacy and control without the need for a centralized infrastructure by means of granting or revoking access or viewing one's records. We used an Ethereum blockchain and IPFS (inter-planetary file system) to store records because of its advantage of being distributed and ensuring the immutability of records and allowing for the decentralized storage of medical metadata, such as medical reports. To achieve secure a distributed, and trustworthy access control policy, we proposed an Ethereum smart contract termed the patient-centric access control protocol. We demonstrate how the PCEHRM system design enables stakeholders such as patients, labs, researchers, etc., to obtain patient-centric data in a distributed and secure manner and integrate utilizing a web-based interface for the patient and all users to initiate the EHR sharing transactions. Finally, we tested the proposed framework in the Windows environment by compiling a smart contract prototype using Truffle and deploy on Ethereum using Web3. The proposed system was evaluated in terms of the projected medical data storage costs for the IPFS on blockchain, and the execution time for a different number of peers and document sizes. The findings of the study indicate that the proposed strategy is both efficient and practicable.

Keywords: patient-centered; IPFS; blockchain; privacy; health record

check for updates

Citation: Haddad, A.; Habaebi, M.H.; Suliman, F.E.M.; Elsheikh, E.A.A.; Islam, M.R.; Zabidi, S.A. Generic Patient-Centered Blockchain-Based EHR Management System. *Appl. Sci.* **2023**, *13*, 1761. https://doi.org/10.3390/app13031761

Academic Editor: Gianluca Lax

Received: 21 December 2022
Revised: 20 January 2023
Accepted: 21 January 2023
Published: 30 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

One of the most important components of healthcare is providing a secure and discreet access control approach. In this age, big data is utilized to maintain and retrieve a significant amount of healthcare records over the Internet. For that purpose, cloud networking is becoming increasingly crucial in this process. Despite their simplicity and efficiency, conventional EHR systems pose a slew of privacy and security issues [1]. This is because HR are considered the most sensitive data being collected owing to the sensitive information about patients and diagnoses. HR data has, however, become more susceptible to breach due to the advancement of the internet and digital healthcare systems in current times [2]. As a result, while evaluating a decentralized and trust-based mechanism, the issues of security and privacy must be considered [3]. Table 1 shows the list of abbreviations and acronyms used in the article.

Table 1. List of abbreviations and acronyms used in the article.

Abbreviations	Meaning
EHR	Electronic healthcare record
HR	Health records
EMR	Electronic medical records
PHRs	Personal health records
P2P	Peer-to-peer
SC	Smart contract
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language

1.1. Motivation

In the conventional approach, cloud databases are used by content organizations to amalgamate EHRs, EMRs, clinical images, PHRs, and patient information; for example, doctor name, body measurements, and home-checking gadget information. It is worth noting that a centralized database is vulnerable to cyberattacks, jeopardizing the security and privacy of EHR [1]. Meanwhile, stakeholders and health providers face difficulty in sharing health information attributable to the discordant standards and formats.

The problem is further exacerbated if the EHR of a patient is erased from the hospital's database, resulting in a permanent loss of the record. Thus, it is imperative for the proposed system to be tamper-proof by unauthorized parties to avoid that issue occurring [4]. Another challenge posed by the current healthcare systems is patients do not have total control over their health records since they are maintained by the service providers [5]. As the healthcare data amount continues to multiply, the security and scalability features have become major concerns. Figure 1 shows the overview of the current system architecture for existing health records.

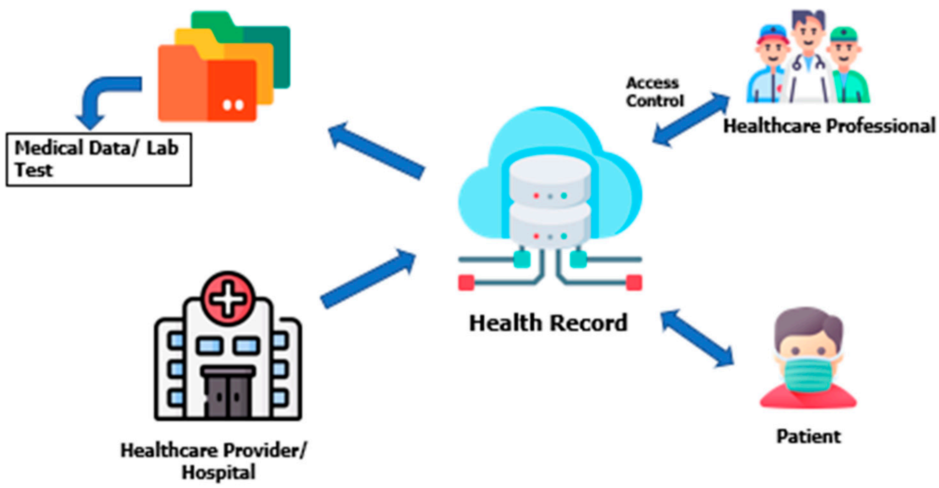

Figure 1. Overview of the current system.

1.2. Contribution

Considering the problems mentioned above, the healthcare industry would require a technology that is able to store data securely and efficiently. The contributions of this article are as follows:

- A proposed PCEHRM system's structure is introduced, and the different system's components interactions are demonstrated.

- The proposal of a patient-centric access control framework is based on a smart contract protocol to grant stakeholders access to the health records. It determines specific functions to be used to send information in and out of the Ethereum blockchain and to provide access privileges between stakeholders.
- The proposed patient-centric distributed architecture focuses on confidentiality, reliability, integrity, portability, and scalability through a blockchain-based approach.
- A novel algorithm is introduced for preserving and securely retrieving healthcare records using blockchains.
- The viability and coherence of the access process and system interaction amongst network participants was tested by putting the suggested framework into practice.
- The average time taken to download and upload health data, the cost of under-taking PCEHRM functions, as well as the efficiency of the transaction were examined and evaluated. Consequently, examining the test network permits us to validate and optimize the prototype prior to publishing it on a public blockchain.

2. Related Work

This section covers the study related to using blockchain technology in the e-healthcare industry to guarantee safe data storage and effective access management. Blockchain technology offers a cryptographic fix to the security issues that are hindering the growth of electronic health systems. Medical enterprises have encountered taxing challenges over the past 20 years as a result of incidents involving record breaches in major medical data centers [6]. MedRec became the first organization to suggest a blockchain-based electronic patient record management system in the early days of blockchain technology [7]. At that time, the Ethereum blockchain and smart contracts preserve detailed accessible healthcare records. However, the medical records are stored in third-party databases operated by healthcare providers, instead on the blockchain network. As a result, the violation or misuse of these records is still a possibility. Healthcare management framework encrypts patient keys during the process of creating or updating data on the blockchain [8]. To decrypt the record, healthcare personnel and lab technician would have to request authorization from a patient's public key. This is starkly in contrast to our approach, as patients are the sole party with access to and control over their data. Any node can connect to the network and conduct transactions on the public blockchain [9]. A blockchain-based application called Medchain allows patients, pharmacists, and hospitals to exchange healthcare data [10]. This architecture stores data on-chain, and thus, presents scalability and privacy issues. A blockchain-based IoT solution may utilize smart contracts to track patients' health [11]. The authors in [12] proposed a blockchain-driven system for maintaining electronic medical records, while [13] presented a blockchain-based architecture that incorporates distributed health records with node models to optimize the replication of healthcare records. The researchers in [14] used blockchain smart technology, a decentralized network with smart contracts, to save and transfer data in a secure manner. In a study by [15], a privacy-preserving system was developed for remote patient surveillance. A permissioned blockchain with an access control audit log was also developed to store health records, but it raises privacy concerns owing to the distribution of the audit log with all interested parties [16]. The authors of [17] argued about the robustness of blockchain technology in storing medical records and raised concerns about the scalability and privacy issues concerning the framework. In another study by [18], they proposed a ring structure-based access control that ensures privacy, but inadvertently leads to an unstable system as the data is maintained in an on-chain database. Ref. [19] also proposed an intelligent data management framework for the cyber system. The decentralized storage and access of records, as described in [20], effectively utilize the network's power and resources. In another study by [21], they employed high-end privacy-enhancing technologies such as homomorphic encryption, which permits data processing while maintaining complete encryption to avoid vulnerability problems. Meanwhile, the researchers in [22] adopted the zero-knowledge proofs approach, along with proof authority consensus for mutual authentication to enhance privacy by ensuring that nodes were not engaging in a malicious manner. Using blockchain technology, several cryptographic mechanisms for preventing tampering and becoming an effective data storage and sharing solution are shown in Table 2.

Table 2. Several solutions of the existing techniques EHR systems currently lack adequate interoperability to provide private, effective, and secure access control.

Ref	Implemented Problems	Problem to Be Solved
[7]	Data Integrity and Interoperability	Privacy and scalability
[10]	Sharing if Data and Integrity of Data	Privacy and scalability
[11]	Public data access and Integrity of data	Interoperability and authentication
[13]	Interoperability	Secure, Privacy, and Scalability
[15]	Privacy and Security	Scalability and Interoperability
[16]	Scalability and interoperability	Privacy
[17]	Security	Scalability and Privacy
[18]	Security and Privacy	Scalability

3. Framework Components

In this study, as shown in Figure 2, we established a permissioned infrastructure in an Ethereum Blockchain framework that allows complete control of records by patients while ensuring privacy, durability, and security. This is carried out by maintaining health information primarily on the blockchain as hashes, whereas the original bulk quantities of data are stored off-chain in IPFS to guarantee effectiveness and scalability.

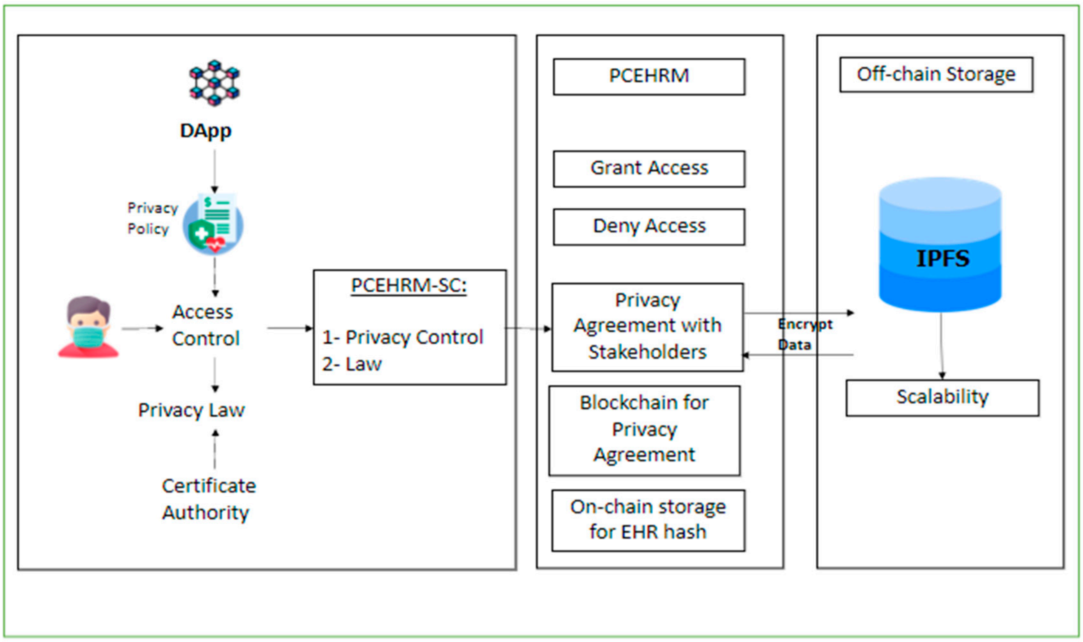

Figure 2. Framework of the proposed system.

The smart contract chain code protocol in this paper is termed a Patient-Centric Healthcare Data Management Access Control-Smart Contract, in which this SC is used to write the role-based access control chain code for recognized stakeholders involved and does not employ any incentive mining beyond ensuring that every party has equal access to the system. The role-based unique ID is initiated after the registration of stakeholders. Following this process, public and private key pairs are provided to each stakeholder for the storage and transfer of health information. In this architecture, a patient's health record is created by doctors. Subsequently, the patient would commit the encrypted health record to be stored in IPFS indelibly while the record hash from IPFS is conserved via the Ethereum blockchain.

In the event that the doctors need to update the patient's health record, one can allow or block the process by granting or revoking access, respectively. The temporary view, or patient-centric view, of the health record, is constructed from IPFS. In this view, the doctor is allowed to update the documentation before the patient commits to the update using their key pairs to save the updated files in IPFS. In this way, this framework would ensure the interoperability of the mechanism.

Additionally, patients have the right to grant access to share the records with only the relevant stakeholders in a patient-centric view of the record from the IPFS system. A doctor's session also would expire before the hash value is committed to the ledger so that non-concerned personnel would have gained access to a record without the patient's permission, hence, safeguarding the data privacy in the system. At the back end, smart contracts are generated for several healthcare procedures.

Hence, role-based access control employed in this framework protects the privacy of the patient. Moreover, our proposed system also has improved scalability and interoperability features as compared to the current system.

3.1. Ethereum Blockchain

Ethereum is designed based on the Bitcoin architecture and features a framework for programmable smart contracts (SC) [23,24]. Simply put, SC is a software program that holds guidelines for negotiating contract conditions. The cost of building and maintaining a centralized database can be minimized by allowing programs to autonomously validate and implement contract-related agreements. SC utilizes the Ethereum virtual machine to enable users to operate SC well within the blockchain network. The value of gas generally determines how the Ethereum system calculates fees [5,23]. A certain amount of gas, which in turn is purchased using digital currency, is required to perform SC and handles a transaction. Hence, the actual transaction cost can be defined as in the equation below:

$$\text{Ether} = \text{gas price} \times \text{gas used}$$

The Ethereum platform predominantly consists of two types of accounts: contract accounts commanded by the contract code, and externally owned accounts (EOAs) controlled by private keys. EOAs are used to undertake ether-sending transactions or to initiate SC execution. Account nonce, sender signature, recipient address, gas limit, ether values, gas price, and the endpoint of the medical data are just a few of the factors that are included in an Ethereum transaction. An affiliated state database for the Ethereum blockchain is built on an IPFS-like Merkle-Patricia tree structure [25]. Essentially, we can design a blockchain framework using IPFS for a more robust and secure off-chain and on-chain storage of medical records. We implemented the proposed system using the Ethereum blockchain's smart contract architecture, resulting in an unambiguous, fine-grained access control mechanism that prohibits hacking and unauthorized access without the patient's consent.

3.2. Distributed InterPlanFile System (IPFS)

Within the P2P IPFS system, a cryptographic hash serves as a distinct fingerprint for each file. In this regard, the hash address is employed to make the contents immutable [11,26]. In the IPFS file storage structure, Merkle DAGs combined Merkle trees with DAGs. The fundamental functionality of IPFS to access health information may be achieved by content-based addressing rather than location-based addressing. By harnessing the IPFS structure, lowered bandwidth costs can be achieved, file download speeds can be improved, and a substantial amount of data may be transmitted without duplication, which can free up storage space. Additionally, IPFS is an immutable storage solution since the hash value of an IPFS file cannot be modified.

3.3. A Background of the Proposed System

Our system structure, as displayed in Figure 3, has the organization use three peer nodes, with one acting as a validating peer node and the others as an ordering node for registering stakeholders. Multiple peers can access the same database in this system, which also uses IPFS for distributed data storage [27]. Multiple peers can be added to various locations on different machines to test the system's scalability. This framework, which contains its own ledger and smart contract copies, provides access to ledgers for smart contracts.

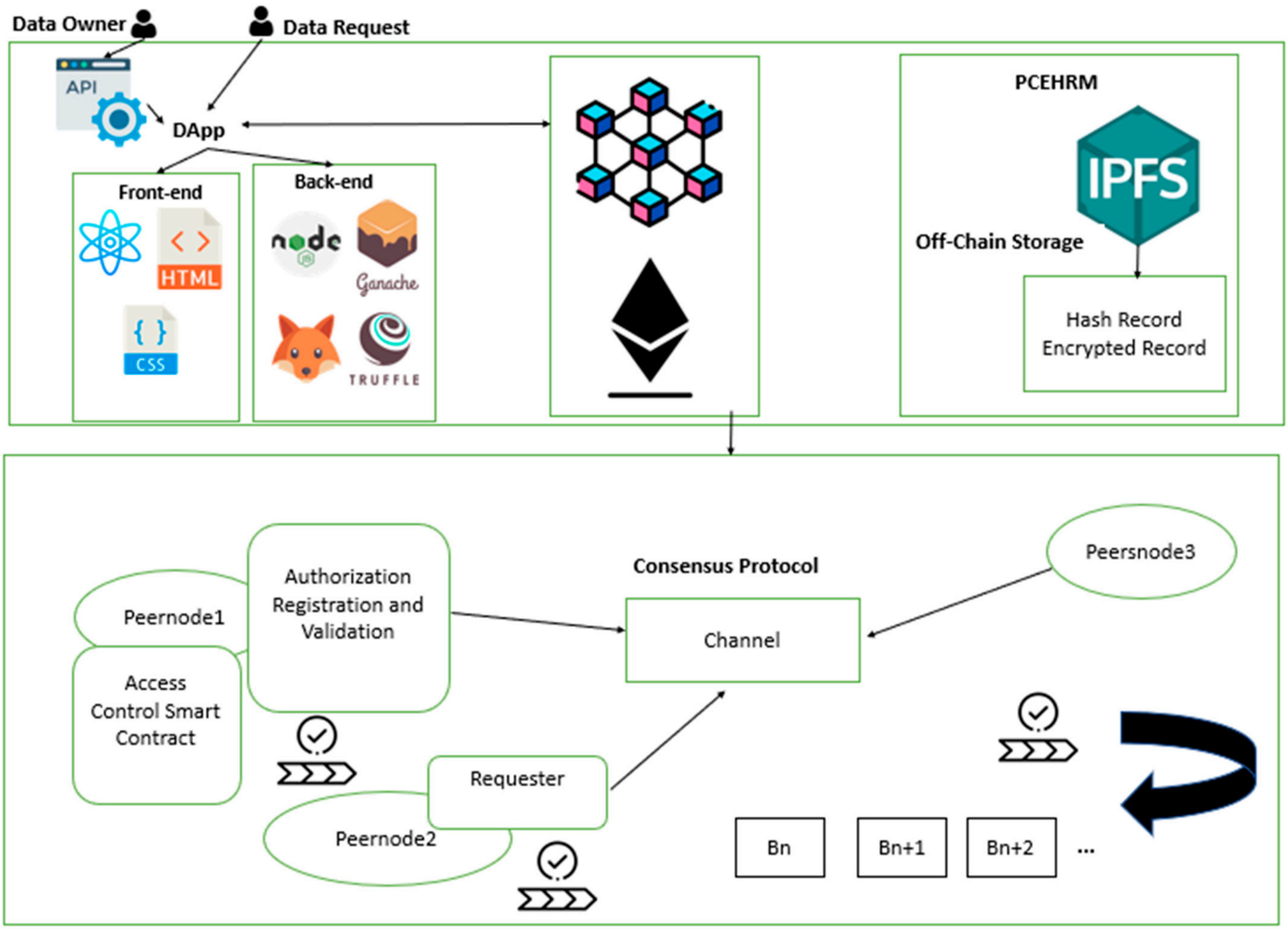

Figure 3. PCEHRM structure.

Peer nodes are linked to the application, which then uses smart to update the ledger. Figure 4 shows the data structure of the blockchain ledger after the integration of PCEHRM data fields, as it is intended to record only the information that patients wish to provide in a transaction. Peernode1, Peernode2, and Peernode3 are the three peer nodes in the organization, and each one has a copy of the ledger and a smart contract.

In this sense, the patient's profile, address and location, diagnoses, doctor recommendations, next review notes, physician's names, medication, scan and test reports, and hospital ID are all included in the healthcare records.

The following stakeholders make up the PCEHRM:

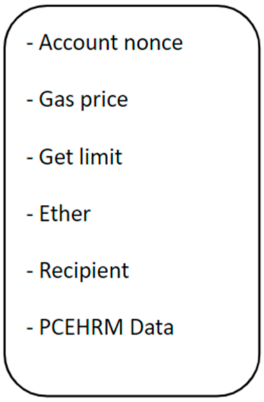

Figure 4. The data structure of the blockchain ledger of PCEHRM.

4. Record Owner

Patients retain their medical data, making them the owner of the record. To store the data, patients are required to acknowledge and sign an agreement on PCEHRM-SC in the Ethereum blockchain. The chain networks permit patients to specify access rights to their healthcare information, defined by each PCEHRM-SC within its context. Table 3 further describes the patient roles in detail.

Table 3. Patient roles.

Patient	Grant-Revoke-Commit, Read Record Revoke permission from Doctor/Service Providers. Permission to Doctor to Read/write of their her. Able to search for Doctor/Labs.

5. Data Uploader

The doctors/lab technicians may upload their patients' medical data to Data Uploaders. The primary responsibilities of the data uploaders include submitting the concerned individual's encrypted clinical data to the IPFS community and validating the preliminary transaction at the blockchain. This is further illustrated in Table 4.

Table 4. Doctors/lab roles.

Doctor/Labs	-Create/Read/Write on Permission for EHR -Search for Doctor in the network
	-Read/Write on Permission for EHR. -Search for Labs the network.

6. Data Users

Data users are defined as the parties that require patients' medical or clinical records for further action, namely, hospitals, researchers, insurance companies, and doctors. In this context, the role-based access control approach in PCEHRM-SC specifies the mechanism for patients to grant access rights to data users.

6.1. Data Encryption

Cryptographic methods guarantee the privacy and integrity of blockchain data. Figure 5 depicts the interaction between patients and their doctors while examining health records. The doctor initiates the process by requesting access permission to retrieve the information stored in the IPFS. Depending on the data fields requested, this generates a patient-centric view of the records rather than disclosing all the patient's information. The session key, Sk, is used to store the encrypted patient-centric view in IPFS and to retrieve records in a finite period. After doctors and patients have received encrypted patient-centric views and Sk, doctors may decrypt the Sk and patient-centric views for the process of updating the record.

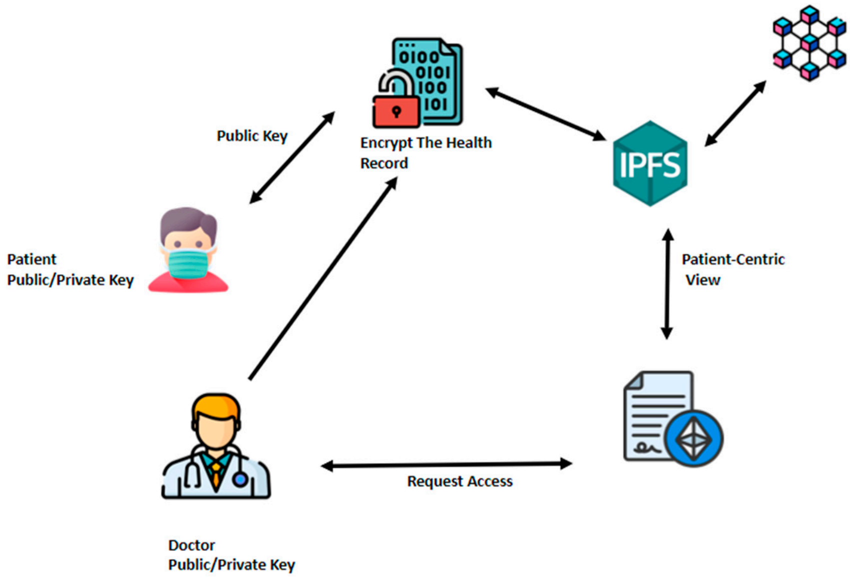

Figure 5. Collaboration diagram between patient and doctor in PCEHRM.

The patient would be notified after the revision of the record in IPFS. The Sk and patient-centric view are automatically erased when a patient commits to his health record. By prohibiting complete access to health records without the patient's consent, this framework is advantageous in terms of protecting patients' privacy. Then, utilizing smart chain code operating on the system's back end, the hash value of the data is safely preserved on the Ethereum blockchain. The ledger will then notify the patient after the records were successfully created or updated.

6.2. PCEHRM-SC

The process of role-based access is initiated once doctors request permission to the IPFS health record of patients, in which patients are in control to allow or revoke access for authorized users as shown in Figure 6. With the patient's consent, the doctor may create, view, and modify medical records before the patient commits the update to permanent storage. The patient-centric view of the health information can only be accessed by other participants in this health chain architecture for a given session, provided their ownership and object ID match the patients' [27]. These stakeholders include insurance agents, pharmacists, and researchers. As for laboratory technicians, they can only modify or update patients' medical records after the approval of the patient and the doctor. The regulations, access control, and privacy agreement are provided by authorization of smart contracts and are governed by the Ethereum blockchain. This methodology adheres to certain situations:

1. An access control regulation specifies the stakeholders' distinct profiles and defines their access rights.
2. After the patient allows access, the system defines the approved value to stakeholders, resources, action types, and environmental attributes.

In addition, the proposed solution segregates privacy attributes into three levels below:

Level 1: Patients are the sole party to access the health record.
Level 2: Approved stakeholders may be made available to the medical record.
Level 3: The permitted patient's caregiver has access to the health record in an emergency.

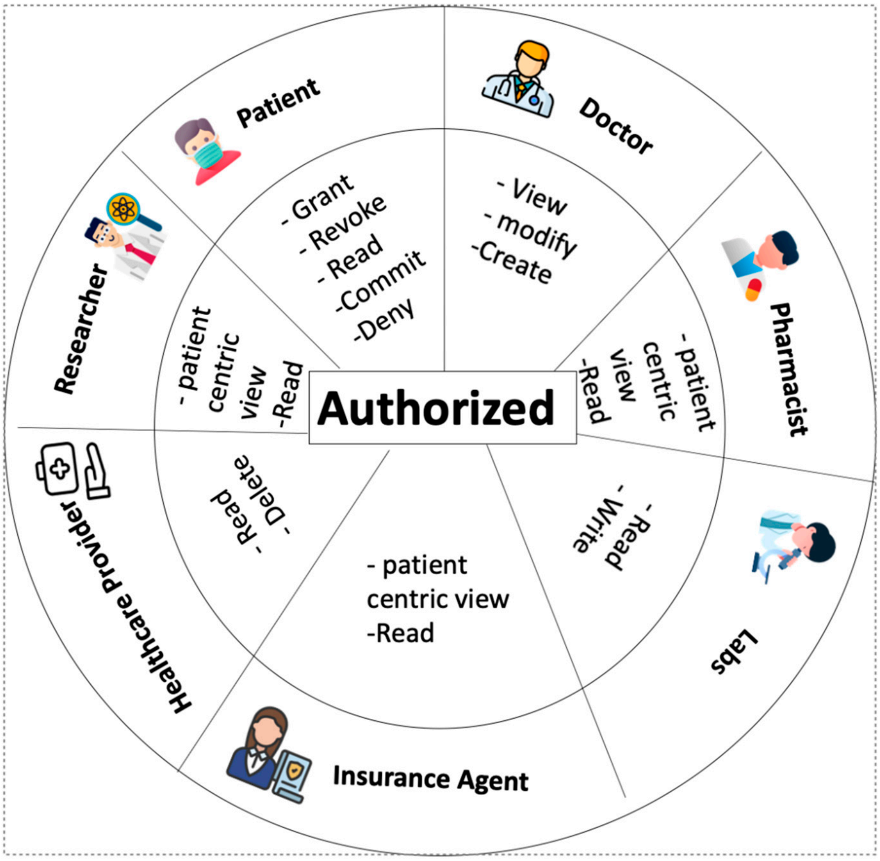

Figure 6. Stakeholders and rule-based access in PCEHRM.

By setting their privacy level, patients may manage the confidentiality of their data. Prior to submitting a modified medical record to the chain network, the levels in our model are calibrated to alter the terms during the transfer of authorizations to other authorized participants.

6.3. PCEHRM Algorithm

Our proposed framework consists of four stakeholders: Pa, D, Ph, and LT, which represent patients, doctors, pharmacists, and lab technicians, respectively. Table 5 shows these symbols in the algorithm functions. The n is the number of doctors, patients, health records, pharmacists, and lab technicians where n = 1, 2 . . . N. Patients, medical professionals, and pharmacists are among the n stakeholders provides public key. All stakeholders will each receive a key pair consisting of public and private keys. On that note, $Papubk_n$, $Paprk_n$, $Dpubk_n$ and $Dprk_n$ are the public and private keys of the patient and doctor, respectively.

In Algorithm 1, the patient allows the doctor access to their health record based on PCEHRM-SC. Hence, a patient-centric view of the health record is generated by the system.

Based on the doctor request, the patient-centric is queried for attribute-based data instead of granting unrestricted access to the patient's health record, as patient-centric can enable users to view and update only the specific fields of HR instead of sharing whole patient health records.

Essentially, the patient-centric view is a subset of the medical record. In accordance, the system creates a session key (S_k) used by both patients and physicians within a particular session. An encrypted session key for $Papubk_n$ and $Dpubk_n$ is derived using the public key for patients and doctors, respectively. The session key S_k which can be sent to doctors is also encrypted with a patient-centric view.

Algorithm 1 calls the *create_Update_HR()* function of Algorithm 2 to start updating the HR_n. After the doctor's and $Pacenv_n$'s session key have been decrypted, the modifications are uploaded into the updated patient-centric view (UP_Pacenv_n).

The encrypted HR_n is decoded once the system is updated, acquired by decryption of the encrypted private key using the patient's password, and appended to the encrypted UP_Pacenv_n. For the last step, the HR_n is saved in IPFS after the patient commits to the updates. Then, once the health record HR_n is committed, it immediately ceases the S_k and $Pacv_n$. In the Ethereum blockchain, the health record hash value HR_n_hash is generated by the IPFS and saved in blocks.

Table 5. Symbols in the algorithm functions.

Symbols	Definition
Pa_n	nth Patient
D_n	nth Doctor
Ph_n	nth Pharma
LT_n	nth Lab Technician
HR_n	nth Health Record
$Papubk_n$	nth Patient Public Key
$Paprk_n$	nth Patient Private Key
$Dpubk_n$	nth Doctor Public key
$Dprk_n$	Doctor Private Key
S_k	nth Session Key
$Pacenv_n$	nth Patient-Centric View
UP_Pacenv_n	nth Update Patient-Centric View
HR_n_hsh	nth Health Record Hash Value

Algorithm 1. System_Function()

Input: Doctor D_n, with their Public key $Dpubk_n$, with their Private key $Dprk_n$, with session key S_k of HR_n Health_Record. Patient Pa_n with their Public $Papubk_n$, and Private key $Paprk_n$.
Output: Boolean (True or False)

Function for storing and updating health records.
For user U have Access permission to HR
Check PCEHRM-SC
If (permission=="Grant" && role=="Doctor") then
    Create $Pacenv_n$ for HR_n in IPFS
    $Pacenv_n \rightarrow$ Decryption (Encryption (HR_n))
    Create S_k
    send Encrypted ($Papubk_n$ (S_k), $Dpubk_n$(S_k), $Pacenv_n$(S_k)) to Pa_n,
    D_n and $Pacenv_n$.
  create_Update_HR()
    $HR_n \rightarrow$ [(Decryption $Papubrk_n$ (Encrypted $Papubk_n$ (HR_n)) + Encryption (UP_Pacenv_n)]
  $Pa_n \rightarrow$ Commit (IPFS (HR_n))
  IPFS $\rightarrow HR_n\_hsh$
  $HR_n\_hsh \rightarrow$ Ethereum Blocks
  Return True
Else
  Permission=Deny
  Return False
End if
End For
End Function

Algorithm 2. **create_Update_HR ()**

Input: $\mathcal{D}_n$, $\mathcal{D}pubk_n$, $\mathcal{D}prk_n$, $\mathcal{S}_k$
Output: Storage of HR

1. Function Doctor $\mathcal{D}pubk_n$
2. **For** Doctor with $\mathcal{D}pubk_n$, $\mathcal{S}_k$
3. $\mathcal{D}_n \rightarrow$ Decrypt($\mathcal{D}pubk_n(\mathcal{S}_k)$)
4. $\mathcal{D}_n \rightarrow$ Decrypt($\mathcal{P}acenv_n(\mathcal{S}_k)$)
5. $\mathcal{P}acenv_n \rightarrow \mathcal{UP}\_\mathcal{P}acenv_n$
6. IPFS Encrypt($\mathcal{UP}\_\mathcal{P}acenv_n(\mathcal{S}_k)$)
7. **End For**
8. **End** Function

7. PCEHRM Protocol Operation

The proposed architecture consists of two parts of development environments, back-end and front-end. It is also developed upon network entities and smart contracts to handle each transaction by utilizing an IPFS storage solution. Performance evaluation was carried out on a intel®Core™i7 @ 3.38 GHz processor and 16 GB of RAM. For the back-end development, we used the Ethereum Platform and a Windows Foundation project, while the front-end development was implemented using JavaScript, CSS3, HTML5, and ReactJS. We also integrated third-party frameworks such as jQuery and Bootstrap to construct the web application in a more user-friendly and effective fashion. REST API servers were used to execute back-end programming while a database was employed for Front-end programming. On top of that, Truffle Framework, Ganache (Ethereum Blockchain), NodeJS and its libraries, as well as Metamask Browser extension (to connect Browser to Blockchain) were also adopted for this experiment. Solidity programming language was adopted for the implementation of PCEHRM in the remix IDE [28]. Using Kubo (go-ipfs), we established IPFS and transmitted an encrypted medical record to the network, returning a distinct hash value associated with the uploaded record. Then, by specifying the IPFS hash, the essential medical record attributes, and the patient Ethereum public key, we modified transactions on the blockchain. HTTP methods, for instance, POST, GET, PUT, and DELETE are triggered once clients consume web applications to call out actions. These methods, in turn, invoke HTTP responses by the web service based on the request by the client. The full PCEHRM-SC prototype code is available in our GitHub repository.

7.1. Add Users

Figure 7 details the successive steps to add users to the network. The smart contract may now create a role-based unique ID for the enrolled stakeholders upon their registration.

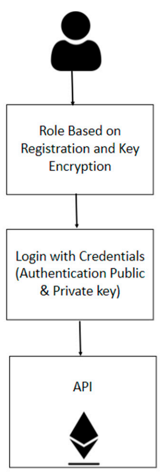

Figure 7. Add user in PCEHRM.

7.2. Add Records and Update Records

After the patient has been allowed access, the doctor may create a record before it is encrypted and stored in IPFS using the hash value that is maintained via the Ethereum blockchain. A temporary, patient-centric view of the medical record is supplied so that the physician may update the view and subsequently, the existing record incessantly in both IPFS and health record chains following the patient's authorization. Consecutively, the session key would lapse, denying doctors access to patient-sensitive information. Figure 8 specifies the step-by-step operation of appending and modifying health records in the network.

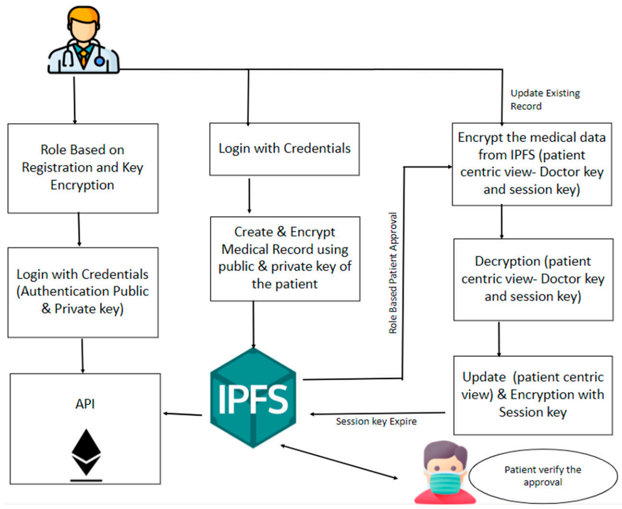

Figure 8. Add records and update records.

7.3. Assuring Authorized Users Have Access

Patients maintain restrictive access control to grant or deny access permissions to stakeholders within a restricted setting to allow them to view, modify, or create records, providing patients full authority and control over their health records. This is carried out by authorizing consent to medical records based on the role and permission type for authenticated users approved by consensus. In the event that healthcare personnel are denied access, the records would not be made available to them or other attending physicians. Upon interaction of users with the system, smart contracts identify and validate the requests, perform data updating, and provide access rights. Figure 9 describes the actions sequentially for the role-based access rights and permission.

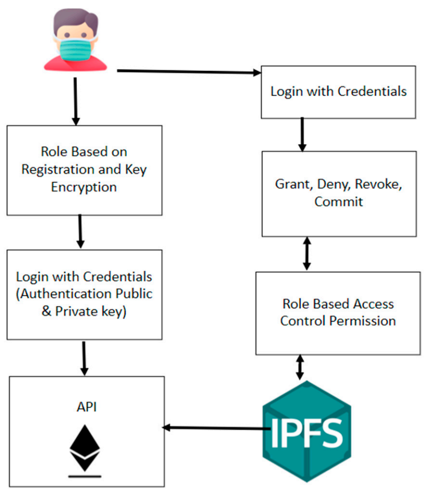

Figure 9. Assuring authorized users have access.

7.4. Records Retrieval

Patients must allow access for the retrieval of partial attribute-based data from IPFS using the hash value in the chain network, so that all stakeholders may access their records. Figure 10 delineates the view record step-by-step implementation.

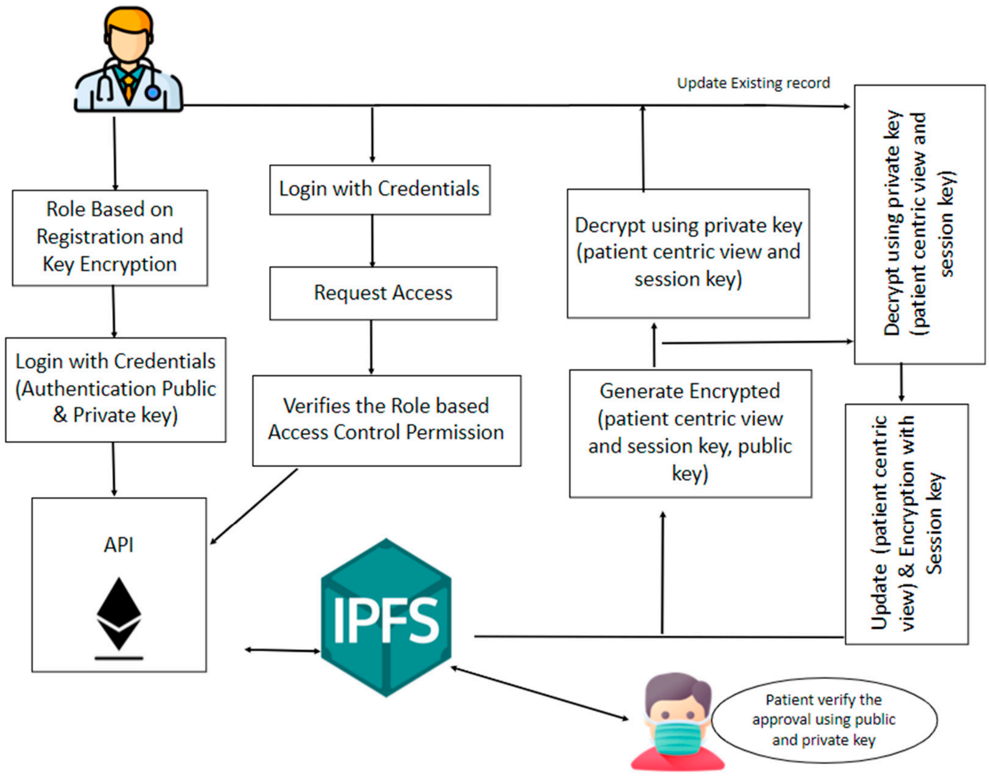

Figure 10. Records retrieval.

7.5. Framework Implementation

Chain codes, smart contracts, network entities, and IPFS storage are all components of the Health Chain Network Transaction Framework [26]. In the user sign-up module of the network, receptionists, pharmacists, physicians, and other medical personnel can enroll using their respective roles, upon which the certificate of authority is produced after the registration. Users can sign in using their email address credentials and password after enrolment. In this case, receptionists may approve or decline appointments made by the stakeholders using patient IDs and update them based on the patient's data. If the appointment proceeds, doctors may create or update the patient's medical record via notes or diagnosis results before being uploaded to IPFS.

Our recommended business ecosystem contains elements such as assets, stakeholders, and transactions, as depicted in Figure 11. Several tests were carried out for the health chain framework prototype to evaluate its performance and functionality through four case studies that describe its storage, scalability, efficiency, and security.

7.5.1. Storage Efficiency

To assess if the interplanetary database can sufficiently hold and retain health records, a few cases were performed in the following ways:

1. Doctors were able to upload health records.
2. With the patient's permission, a doctor could view the medical records.
3. Patients were capable to view their health data.
4. Patients and doctors were permitted to retrieve health records based on their credentials.
5. Encrypted health records were generated proficiently.

Subsequently, encryption and decryption of the modified records in IPFS could be accomplished by doctors and patients, respectively, using their corresponding session keys.

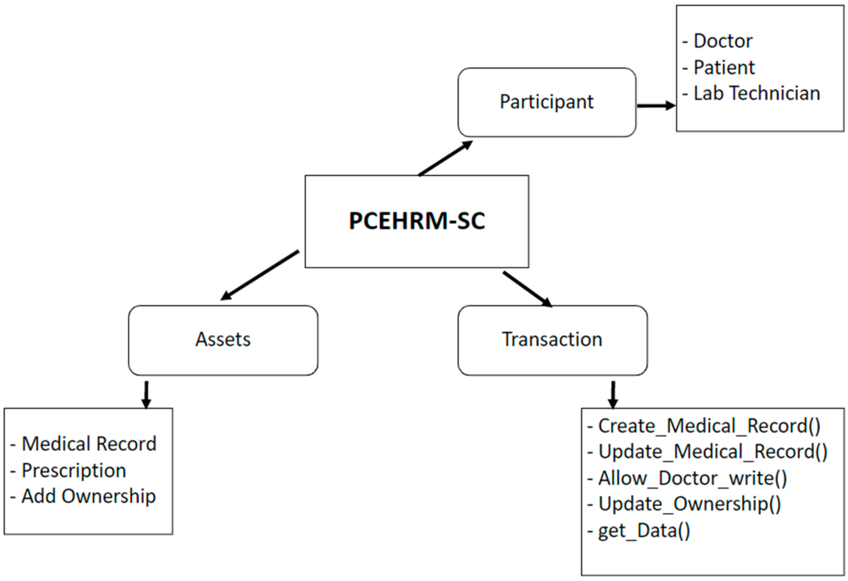

Figure 11. Proposed business ecosystem.

7.5.2. Security

Before the data is saved in IPFS, the following instances were verified to confirm security requirements for the effective implementation of encrypted health records:

1. Encryption of the user's password.
2. Health records were saved in IPFS with encryption.
3. Each record was assigned a distinct hash value.
4. Public and private keys were assigned to all stakeholders.
5. Session keys assignment and expiration.

7.5.3. Privacy

The objective of this test is to confirm whether or not stakeholders were granted or denied access to health records depending on their role. The access control was verified using test instances as listed below:

1. Depending on their role, stakeholders were able to view their respective home pages.
2. Access rights based on the stakeholders' role.
3. Session keys were allocated for viewing medical records.

This was performed to prove that the system is capable of assigning access permissions based on their levels and roles.

8. Evaluation

In this section, we tested the viability and coherence of the access process and system interaction amongst network participants by putting the suggested framework into practice. We investigated the average time taken to download and upload health data, the cost of undertaking PCEHRM functions, as well as the efficiency of the transaction. Consequently, examining the test network permits us to validate and optimize the prototype prior to publishing it on a public blockchain.

8.1. Verification of PCEHRM-SC

By testing two key functions for brevity givePermission(), removePermission(), we were able to confirm the access rights and interactions between entities. The results of these test are presented in Figures 12 and 13, depicting the summary of the transaction event log maintained in the blockchain once the functions were successfully executed.

{ "event" : "GivePermission" ,
"patient_id" : "0xc87656D593CD4e0178c7dbbE0D982a7d8dDFEA56"
"info" : "Approved" }
{ "event: "Approved" ,
"requester_id" :" 0x770a246c2CF57D25661Ef48C8D01b14D89264169"
"Info" :" Authorised access to HR" }

Figure 12. Result of give permission function.

{ "event" : "RemovePermission" ,
"patient_id" : "0xc87656D593CD4e0178c7dbbE0D982a7d8dDFEA56" ,
"info" : "Failed" }
{ "event: "Failed" ,
"requester_id" :" 0x770a246c2CF57D25661Ef48C8D01b14D89264169" ,
"Info" :" Need more details info" }

Figure 13. Result of revoke permission function.

8.2. Practical Applications and Costs

In the proposed system, we define the cost for the actual transactions and time taken to execute for each transaction. The cost is calculated by ether = gas used × gas price; "gas used" means the constant computational cost. The network adjusts the price of gas to compensate for changes in the value of ether [8]. Thus, the total transaction costs (ether) for the accessibility of health data remain relatively constant. Regarding the paying segment, all participants must pay gas for performing an operation in SC. Therefore, the automated SC process would result in significant cost savings for the patient. In the implemented PCEHRM-SC prototype, we have set a gas limit of 30,000, with each gas unit fixed at 2 Gwei.

Table 6 summarizes the time taken and cost of operations performed in SC. The addDoctor() function is implemented once at a cost of 0.00089442 eth with used time 10.316s. The addPatient() function costs around 0.093 eth, which is higher than other functions due to the additional input bytes that are included during function execution, such as the patient's blockchain address and usage notes. However, the used time for execute this function is not high compared to executing the getpatientdetails() function was 0.213s which requires fetch the EHR's data of the patient.

Table 6. PCEHRM time and cost analysis (database).

Function Name	User	Start Time	Time Taken	Gas Used	Total Cost
addDoctor()	0x0bCD3A9Fcc8EfC4B5B2c07BC1129454A9C1fc56D: (Admin)	10/27/22, 7:27 A.M.	10.316 s	44,721	0.00089442 ETH
addPatient()	0x770a246c2CF57D25661Ef48C8D01b14D89264169: (Patient)	11/30/22, 6:35 P.M.	13.669 s	148,329	0.00296658 ETH
addDoctor()	0x0bCD3A9Fcc8EfC4B5B2c07BC1129454A9C1fc56D: (Admin)	11/30/22, 6:37 P.M.	13,949 s	44,721	0.00089442 ETH
getpatientDetails()	0x770a246c2CF57D25661Ef48C8D01b14D89264169: (Doctor)	11/30/22, 6:38 P.M.	0.213 s	0	0 ETH
saveMedicalRecord()	0xa96C9CB8b1Ae46FEE4cC3d65877464dB72070e5e: (Doctor)	11/30/22, 6:38 P.M.	12.506 s	168,010	0.0033602 ETH

In general, the overall costs can be further reduced by keeping the size of the input data to a minimum. However, these costs are still lower than those associated with purchasing storage space from a third party or maintaining a database through a centralized system such as ISN [29], MedBlock [30], MeDShare [31], and MedChain [11].

8.3. Performance Evaluation

Storing large amounts of data on the blockchain is costly, so to minimize costs, IPFS hashes are stored on the blockchain. Figure 14 shows the predicted difference in cost events as the number of transactions on the blockchain increases. Primarily foresee the cost of writing data to the blockchain, as it is a more expensive transaction than reading data from the blockchain. As shown in Figure 14, storing his IPFS hash rather than the entire image significantly reduces storage costs.

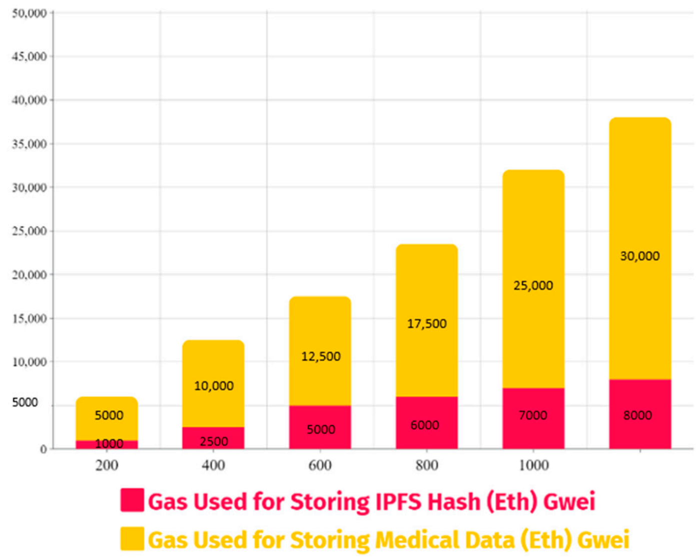

Figure 14. Medical data storage costs projected vs. IPFS on blockchain.

Our proposed system stores the media in IPFS, and IPFS stores media with the help of peers. Figure 15 shows that as the number of peers accessing the transaction or contributing to image storage increases, the execution time, or the time required to retrieve the media, increases. Depending on the previous section, the larger the image or report size, the longer it takes to access the same.

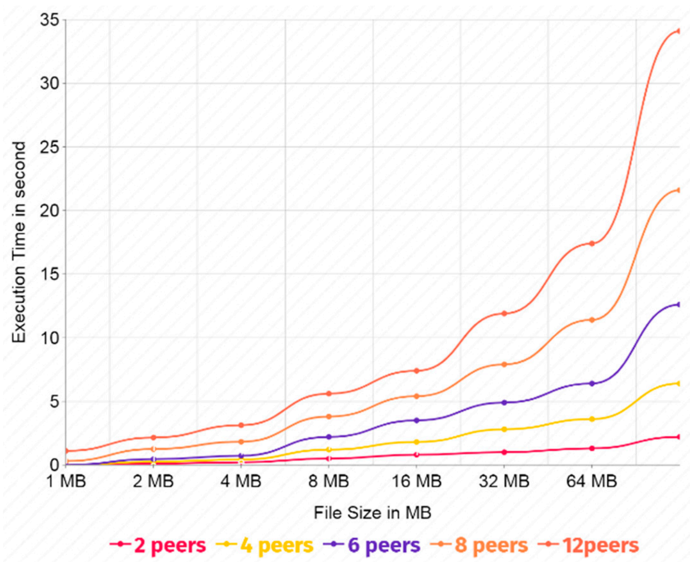

Figure 15. Execution time based on the number of peers and document size.

9. Discussion

To ensure the best possible health care experience for the patient, the patient should be able to access or grant access to his or her medical information as needed. This is currently not possible when the data is stored in a hospital's proprietary Electronic Health Record (EHR). As a result, there is a requirement for patient operated where the details of the patient's treatment are with the patient. This paper proposes a secure, interoperable patient-centric data access management system based on blockchain. In our proposed system, the patient has complete control over his or her health record-related data, which is stored securely using IPFS. Using a token, patients can grant controlled access to their medical data to health care providers for a set period of time. In the following, Table 7 explains the summary of the comparison between the existing and suggested patient-centric health storage models focusing on privacy, scalability, security, and integrity. The authors in [16,17] reviewed current strategies for healthcare management using blockchain technology and its effects. The existing architectures in [10,15,18,32] were examined. Each block includes a health record hash value that would transform whenever the record is updated. This ensures the records are immutable as it is computationally expensive to manipulate the ledger. On top of that, stakeholder access to patients' medical records is prohibited by access control rights and levels.

Table 7. Comparison between the existing and suggested patient-centric health storage models.

Ref	Ease of Scaling	Access Control	Confidential Information	Data Integrity	Data Security	Patient-Centric
[10]	✗	✗	✓	✓	✓	✗
[15]	✗	✓	✓	✓	✓	✗
[16]	✓	✗	✗	✓	✓	✗
[17]	✗	✗	✗	✗	✓	✗
[18]	✗	✓	✓	✓	✓	✗
[32]	✗	✗	✓	✓	✓	✗
PCEHRM	✓	✓	✓	✓	✓	✓

10. Conclusions

Medical records are the most precious commodity in the industry of healthcare intelligence. Often, this information is indeed dispersed across various platforms, posing a challenge in sharing them for an efficient and cohesive healthcare system. Meanwhile, a centralized hosting solution, for instance, a cloud-based platform, is vulnerable towards a single point of a cybersecurity attack. Decentralized designs and system interoperability have received more attention as the dispersed nature of healthcare services has come to light. In this study, we incorporated the PCEHRM system, a decentralized framework for sharing access and storing medical data that is built on the Ethereum blockchain and IPFS architecture. We also introduced a novel access control system termed PCEHRM-SC that provides authorized parties access to the pertinent blockchain data. By giving patients complete management over their healthcare records utilizing smart contract protocol, the PCEHRM system promotes a unique manner to enhance their rights. For example, patients have full control over their medical reports and possess the ability to allow or deny access to the records for use in clinical trials or research. We also analyzed and evaluated the feasibility, effectiveness, and rationality parameters of the proposed framework. 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.

By offering higher productivity, data integrity, and efficient audit, while allowing for shared access to medical data, the suggested scheme makes it easier for patients to access an immutable medical database. Since the data storage and exchange model are also decentralized, it is crucial to involve third-party intermediaries and eliminate administrative structures. Our long-term study objective is to implement the suggested system architecture on the public blockchain using real-world examples to establish a worldwide PCEHRM system and to assess associated laws and standards to integrate this cutting-edge technology within the healthcare system. Without a doubt, the artificial intelligence component would aid clinicians in the analysis of diagnostic medical data and communicate more effectively with patients.

Author Contributions: Conceptualization, methodology, and writing, A.H. and M.H.H.; software, A.H.; validation, supervision and revision: M.H.H., M.R.I., S.A.Z., E.A.A.E. and F.E.M.S.; funding: M.H.H., M.R.I., E.A.A.E. and F.E.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of scientific research at King Khalid University for funding this work through Research Group Program under grant number (RGP.1/147/43).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was conducted at the IoT and Wireless Communication Protocols Laboratory at the ECE department, International Islamic University Malaysia (IIUM).

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Madine, M.M.; Battah, A.A.; Yaqoob, I.; Salah, K.; Jayaraman, R.; Al-Hammadi, Y.; Pesic, S.; Ellahham, S. Blockchain for giving patients control over their medical records. *IEEE Access* **2020**, *8*, 193102–193115. [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. Saidi, H.; Labraoui, N.; Ari, A.A.; Maglaras, L.A.; Emati, J.H. DSMAC: Privacy-aware Decentralized Self-Management of data Access Control based on blockchain for health data. *IEEE Access* **2022**, *10*, 101011–101028. [CrossRef]
5. Makridakis, S.; Christodoulou, K. Blockchain: Current challenges and future pro-spects/applications. *Future Internet* **2019**, *11*, 258. [CrossRef]
6. 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]
7. 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.
8. 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.
9. Dannen, C. *Introducing Ethereum and Solidity*; Springer: Berlin/Heidelberg, Germany, 2017.
10. Shen, B.; Guo, J.; Yang, Y. Medchain: Efficient healthcare data sharing via blockchain. *Appl. Sci.* **2019**, *9*, 1207. [CrossRef]
11. 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]
12. Margheri, A.; Masi, M.; Miladi, A.; Sassone, V.; Rosenzweig, J. Decentralised provenance for healthcare data. *Int. J. Med. Inform.* **2020**, *141*, 104197. [CrossRef]
13. 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]
14. 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]
15. Dwivedi, A.D.; Srivastava, G.; Dhar, S.; Singh, R. A decentralized privacy- preserving healthcare blockchain for iot. *Sensors* **2019**, *19*, 326. [CrossRef] [PubMed]
16. 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]
17. Jagadeesh, R.; Mahantesh, K. Blockchain-based knapsack system for security and privacy preserving to medical data (2021) in SN COMPUT. *Scientifur* **2021**, *2*, 245.
18. 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]
19. 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]
20. 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]
21. 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]
22. Piao, Y.; Ye, K.; Cui, X. A Data Sharing Scheme for GDPR-Compliance Based on Consortium Blockchain. *Future Internet* **2021**, *13*, 217. [CrossRef]

23. Wood, G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Available online: https://gavwood.com/paper.pdf (accessed on 23 November 2022).
24. Buterin, V. Ethereum White Paper. Available online: https://github.com/ethereum/wiki/wiki/White-Paper (accessed on 25 November 2022).
25. Dhillon, V.; Metcalf, D.; Hooper, M. *Blockchain Enabled Applications*; Apress: Berkeley, CA, USA, 2017.
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.
27. 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. [CrossRef]
28. 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]
29. Langer, S.G.; Tellis, W.; Carr, C.; Daly, M.; Erickson, B.J.; Mendelson, D.; Moore, S.; Perry, J.; Shastri, K.; Warnock, M.; et al. TheRSNA Image Sharing Network. *J. Digit. Imaging* **2014**, *28*, 53–61. [CrossRef] [PubMed]
30. Fan, K.; Wang, S.; Ren, Y.; Li, H.; Yang, Y. MedBlock: Efficient and Secure Medical DataSharing Via Blockchain. *J. Med. Syst.* **2018**, *42*, 136. [CrossRef] [PubMed]
31. Xia, Q.; Sifah, E.B.; Asamoah, K.O.; Gao, J.; Du, X.; Guizani, M. MeDShare: Trust-Less Medical Data Sharing among Cloud ServiceProviders via Blockchain. *IEEE Access* **2017**, *5*, 14757–14767. [CrossRef]
32. Lee, W. Getting Started with Smart Contract and using the MetaMask Chrome Extension. In *Beginning Ethereum Smart Contracts Programming*; Apress: Berkeley, CA, USA, 2019; ISBN 978-1-4842-5086-0.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.