BLOCKCHAIN® IN HEALTHCARE TODAY
PLATFORM APPROACHES IN HEALTHCARE

NARRATIVE/SYSTEMATIC REVIEWS/META-ANALYSIS

Efficient Health Information Exchange Automation System Based on Blockchain and IPFS

Rohit Kumar Jarariya, MTECH; Sri Khetwat Saritha, PhD; and Sweta Jain; PhD

Department of Computer Science Engineering (CSE), Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India

Corresponding Author: Rohit Kumar Jarariya; Email: jarariyarohit171198@gmail.com

DOI: https://doi.org/10.30953/bhty.v8.423

Keywords: Blockchain, distributed ledger, Ethereum 2.0, health information exchange, HIE, Interplanetary File System, IPFS

Abstract

In today's digital era, secure and efficient management of health information is a critical challenge. Centralized health data systems often expose sensitive information to security risks, enabling unauthorized access, modifications, and data sharing without patients' consent. These challenges require a patient-centric, standardized approach to managing health data. This article presents a decentralized health information exchange framework that leverages blockchain technology to address these issues. The framework combines the Interplanetary File System for scalable data storage with Ethereum (ETH) smart contracts to enforce secure and transparent access control. By integrating these technologies, the proposed solution presented here enhances data security, transparency, and interoperability while reducing costs and reliance on intermediaries. Experiments conducted on the ETH blockchain demonstrate the framework's efficiency, with smart contracts evaluated for transaction costs and accuracy. In addition to examining scalability and security, the authors discuss the framework's limitations and its potential for broader application. To foster further research and collaboration, the source code for the smart contracts is openly available on GitHub.

Plain Language Summary

Centralized health information systems are susceptible to breaches, unauthorized alterations, and non-consensual data sharing. This article presents a decentralized framework employing blockchain for patient-centric data management. It integrates the Interplanetary File System for scalable, distributed storage and Ethereum smart contracts for granular, transparent access control. This approach enhances security, interoperability, and cost-efficiency by eliminating intermediaries. Empirical evaluation on the Ethereum (ETH) blockchain confirms low gas costs (the computational fee, measured in gwei and paid in ETH, for executing transactions and smart contracts), high execution accuracy, scalability, and resilience against threats, addressing key limitations of traditional systems.

Submitted: June 30, 2025; Accepted: December 25, 2025; Published: January 21, 2026

Health information exchange (HIE) systems play a pivotal role in modern healthcare in sharing patient information between different healthcare providers with guaranteed security and efficiency. These systems provide improved patient care, a reduction in the occurrence of medical errors, and thus an enhancement in healthcare delivery.

Most of the HIE applications depend on the central architecture-based models, which introduce several critical challenges. These include vulnerability to data breaches, lack of scalability, and limited interoperability. In addition, centralized control often compromises patient data confidentiality, because it always remains at the mercy of one controlling authority that can misuse or provide unauthorized access to sensitive health information.

Recent advancements in blockchain technologies provide a decentralized solution to such problems. Because of its inherently immutable and transparent nature and its maintenance of a distributed ledger, blockchain provides a secure and trusted basis for the exchange of healthcare data. When combined with the Interplanetary File System (IPFS), blockchain frameworks can transcend the storage limitations native to on-chain systems, enabling scalable off-chain storage of large medical datasets.

In this article, the authors address these challenges by proposing a framework for HIE that leverages blockchain and IPFS in facilitating secure, scalable, and patient-centric data exchange. Specific contributions of the study are outlined as follows:

1. A distributed architecture, comprising blockchain and IPFS, is developed in order to meet both security and scalability requirements for the storage and exchange of data.
2. Efficient encryption mechanisms will be developed along with a key management framework in order to ensure confidentiality and also control access to a patient's data.
3. Extensive experiments will be carried out for the proof of the proposed system, which reduces the transaction costs as well as increases the scalability over the traditional frameworks.
4. This contribution will surpass the current state of the art, as it will provide insights into how blockchain technology can be leveraged to improve interoperability and efficiency in healthcare data.

While several blockchain–IPFS-based HIE models exist, most face limitations such as high gas consumption, fixed access control models, limited update mechanisms for patient records, and minimal compliance with health data standards such as Health Level Seven, Fast Healthcare Interoperability Resources (HL7-FHIR). In contrast, our framework introduces a dynamic, patient-driven permission model, an encrypted incremental update process for off-chain data, and a cost-optimized smart contract architecture. These features directly address the shortcomings identified in Section 2, as summarized in the comparative analysis (Table 1), clearly position our work as an advancement over prior designs.

The structure of this article is as follows: Section 2 examines existing research and determines limitations in the HIE systems in use today. Section 3 presents the proposed framework, including its architecture and key components. Section 4 details the implementation process and experimental evaluation of the system. Section 5 examines the scalability and performance of the framework under varying workloads. Finally, Section 6 concludes with key findings and directions for future research.

Traditionally, healthcare stakeholders have used HIE for the transmission of healthcare data. The COVID-19 pandemic accelerated technical changes in healthcare, with more organizations focusing on data interoperability. Today, in response to the pandemic, HIEs across the country are adapting to improve the flow of patient data, an essential aspect that will slow the spread of the virus. Research shows that more than 96 of the non-federal acute care hospitals in the U.S. have already integrated HIE.[1] The HIE ensures patient information is kept securely in a database and transmitted through digital media, thereby reducing the instances of prescription errors and data inaccuracies. It allows physicians to coordinate patient care and avoid redundant procedures and expensive errors.

Digitization of data storage eliminates the use of paper-bound processes, hence improving efficiency and productivity. It also increases patient involvement and

Table 1. Key contributions, limitations, and advancements of five related articles in the field, providing a clear comparison of existing solutions and the improvements introduced by this study.

Reference	Key contributions	Limitations	Advancements in this work
6. Zhuang, et al. (2018)	Developed a clinical trail system integrating blockchain for autoamtion	Relied on on-chain data storage, causing scalability	Introduces IPFS for off-chain storage to enhance scalability
7. Li & Han. (2019)	Enahanced Secure data sharing with cryptographic techniques	Did not address interoperability or scalability challenges	Leveraged HL7 standards for improved data interperability
8. Nguyen, et al. (2019)	Proposed a decentralized patient record system using mobile cloud computing	Lacked a robust mechanism for real-time data updateability	Added mechanisms for real-time data updates on IPFS
9. Sharma, et al. (2020)	Implemented zero-knowledge proofs for data privacy	High computational overhead due to encryption methods	Reduced costs with lightweight encryption techniques
10. Jabbar, et al. (2020)	Focused on improving shared EHR interoperability and integrity	Entirely stored data on the blockchain, leading to high costs and limited scalability	Combined blockchain with IPFS and reduce on-chain storage requirements

EHR: electronic health record; HL7: Health Level Seven; IPFS: Interplanetary File System.

Citation: Blockchain in Healthcare Today 2026, **8**: 423 - https://doi.org/10.30953/bhty.v8.423

education. In contrast, nearly all health-related information is centralized in computer databases, where data integrity and confidentiality are violated. Centralization enables a controlling authority to access and alter data without seeking permission from a patient, allowing identity theft and data breaches. Moreover, a central database is a point of failure wherein data may get lost and result in system crashes.

Utilizing blockchain technology can solve the problems of centralized servers or databases. Each block in a blockchain is cryptographically linked to the previous and subsequent blocks, ensuring data integrity. If there is any alteration of data, the blockchain authenticates this alteration with the network. If most peers in the network sense tampering, the affected node must update its blockchain. The blockchain-based algorithms that can be activated to fulfill specific criteria can be programmed over the blockchain network. Such contracts allow authorized parties to access and modify records safely. Patients can manage their personal health record information and grant specific organizations permission to update and access their records securely.

Decentralized apps, also known as "dApps," can be created on top of the open-source, decentralized blockchain technology known as Ethereum (ETH). Compared to Bitcoin, a more inflexible blockchain technology, ETH is more adaptive and versatile. The ETH network uses its coinage, ETH, to pay for transactions and computational services. A sizable and engaged developer community is continually striving to enhance ETH and develop new dApps. In addition, ETH is well-known for its smart contract features, which let programmers design self-executing contracts that may be used for a variety of situations. ETH initially operated on a proof-of-work consensus technique, which makes it slower and limits the number of transactions it can process per second to about 25 to 30. However, by 2023, when ETH has switched to Proof of Stack (PoS), it will be able to process 100,000 transactions per second,[2] and the size of the ETH block will also increase from 2 to 8 MB, which makes the network more effective and assists in the resolution of the blockchain scalability issue. Size is always the constraint of blockchain. The IPFS offers a similar decentralized storage way to store data and can solve scalability issues with the blockchain.

The IPFS is a peer-to-peer, content-addressed file system. The system is primarily designed to be used in decentralized, versioned file storage. Regarding applications such as websites, online businesses, and payment systems, IPFS can be utilized as a distributed web protocol with a fast and secure distributed database.[3] This concept implements data retrieval and file storage. Instead of giving directions to the data, you simply submit a request, and it is found and acquired on your behalf. The fact that the data are distributed across numerous computers allows each of them to feed your computer small amounts of the data concurrently, much like a torrent download.

The intention is to accomplish this to reduce latency, bandwidth, and bottlenecks caused by a single, central server.[4] A file is divided into smaller bits before being stored in IPFS, which is a distributed file system. A distinctive identifier called the content identification (CID) is produced using a cryptographic hash method. The IPFS employs a Distributed Hash Table—a distributed system that links CIDs to peers' Internet Protocol addresses and ports that carry the material via key-value mapping. Each node in IPFS has its own unique identity, which is the hash of its public key. Information can be pinned with the goal of seeding and preserving it indefinitely. Unpinned content will be removed after a specific period of time in order to save storage space. As a result, each node is in charge of the data it stores and seeds.

Related Work

The current medical information management systems have the potential for significant advancements and important changes due to blockchain technology. Several researchers created blockchain-based systems from scratch. To connect with patients and doctors, Zhuang et al.[5] employed two smart contracts, namely the clinical trial smart contract and the HIE smart contract. The hospital, any company, or any centralized data server stores actual health information. They demonstrated how blockchain technology might enable a third-party-free automated validation mechanism for HIE and clinical trials.

Blockchain enables executing HIE from distributed databases in a safe setting using smart contracts. Moreover, it could ensure data protection and authenticity. By encrypting the data and requests, the article's authors[6] have improved secure transmission. They also carried out load testing and data tampering success probability and applied their methodology. Nevertheless, because they employed on-chain storage, they also had scalability problems.

Nguyen et al.[7] proposed that an EHR sharing scheme is possible by mobile cloud computing and blockchain. They identified key challenges, for example, flexibility, availability, decentralized access, identity management, integrity of user authentication, and privacy, and devised a prototype implementation of their model. The authors are primarily concerned with creating a reliable access control system based on a single smart contract to regulate user access and

ensure effective and secure EHR exchange. To replicate the system, they employed the ETH blockchain on the Amazon cloud and performance tests. They used a peer-to-peer IPFS storage solution with blockchain to achieve decentralized data storage and sharing to address the scalability issue. The record updateability to the IPFS network, transaction cost analysis, and load testing are missing in their article.

Both Madine et al.[8] and Sharma et al.[9] suggested a cutting-edge healthcare structure. They recognized the obstacles to the "conventional" blockchain/cloud EHR sharing architecture's deployment in a country such as India and recommended a practical E-card-based alternative. They made use of the proxy re-encryption technique. To get around the drawbacks of proxy re-encryption, which serves as an access control mechanism, a hybrid encryption model that combines symmetric and asymmetric encryption is utilized.[8]

Sharma et al.[9] included two additional entities to safely retrieve, store, and distribute patient medical information, for example, trustworthy oracles and reputation systems. They also performed a cost investigation because they significantly reduced transaction costs compared to Madine et al.[8] The main disadvantage is that each hospital or organization must deploy a first controller each time they start the framework, and using that controller, patients and doctors must register. This causes a problem with standardization. Several controllers must be deployed for this system to be expanded worldwide.

To enhance shared HER interoperability and integrity with a solution and deploy a smart contract prototype on ETH's TestNet, Jabbar et al.[10] implemented their prototype into operation and did cost and time analyses for several functions. The main disadvantage of this architecture is that transaction costs are high, and the authors say nothing about data transfer safety. In addition, data are stored on the blockchain node, which creates a scaling problem.

In the model by Zhuang et al.,[1] the transmission and receiving records are processed using a blockchain adapter. A hashing technique is used to guarantee data consistency. The patient has full control over their data and also establishes a contact point for the doctor to access the records. The primary drawback is the setup needed in each healthcare facility. Each healthcare institution must finish the process of transforming servers into blockchain adapters and contribute at least one node to the blockchain. They must also use a centralized database to store health data.

Mani et al.,[11] deployed the framework over the Hyperledger blockchain platform. They design, implement, and evaluate patient-centered health data management. Numerous HIE problems, including compatibility, scalability, and privacy, have been resolved. Moreover, it features a decreasing access control system that meets patient privacy laws and regulations. However, the main issue is that this framework was implemented on the private Hyperledger blockchain. Moreover, a private blockchain is not considered as secure as a public blockchain. The speed and latency of Hyperledger are comparatively faster as compared to the ETH blockchain. Pawar et al.[12] also built on the Hyperledger blockchain. They implemented their Internet of Things (IoT)-based medical data collection approach. Moreover, the authors do not verify or analyze their approach.

An organized EHR in the healthcare network is proposed by Jayabalan et al.[13] It is effectively a blockchain-based framework, which integrates IPFS for off-chain storage. Implementation is not done for the proposed framework. Cost analysis is also missing. Another drawback of their work is that they have not created a method for record updateability back to the IPFS. In addition, standardization issues still exist. By utilizing HL7 FHIR standards and a blockchain-based HIE platform, Bae et al.[14] attempted to address the issue of HIE standardization. Their primary objective was to demonstrate how the HIE uses HL7 standards. Cost analysis and performance testing have not yet been completed. Moreover, no decentralized network stores real data.

The work discussed above has laid the foundation for utilizing blockchain in healthcare but is constrained by issues such as reliance on on-chain storage, high computational costs, and limited scalability. While several studies attempt to enhance data security and privacy, they often neglect critical aspects such as cost efficiency, real-time update mechanisms, and global standardization (Table 2).

This study addresses these gaps by providing a novel blockchain-based HIE framework that integrates IPFS for off-chain storage, implements robust encryption for patient data security, and incorporates HL7 standards for improved interoperability. The experimental results demonstrate substantial improvement over existing systems in terms of scalability, transaction costs, and system efficiency.

To further clarify the distinctions between our proposed HIE framework and existing blockchain-based systems, Table 3 presents a detailed feature-wise comparison. Unlike prior work that often relies solely on on-chain storage with limited scalability or provides only basic and costly privacy mechanisms, our framework leverages IPFS for scalable off-chain data storage combined with optimized lightweight smart contracts to reduce gas costs significantly.

Citation: Blockchain in Healthcare Today 2026, **8**: 423 - https://doi.org/10.30953/bhty.v8.423

Table 2. Comparative analysis of blockchain-IPFS HIE frameworks.

Feature / capability	Blockchain clinical trials	Crypto-secure sharing	Mobile cloud decentralized EHR	Zero-knowledge privacy	Blockchain EHR interop	This work (proposed framework)
Storage Scalability	On-chain only (limited)	Not addressed	Partial, some off-chain	On-chain (expensive)	On-chain (limited)	IPFS Off-chain (scalable)
Real-Time Data Update	No	No	Not robust	Computationally expensive	No	Efficient, encrypted incremental updates
Interoperability (HL7/FHIR)	No	HL7 basics	No	No	Partial, not explicit	Explicit HL7 FHIR compatibility
Privacy & Access Control	Basic, on-chain	Cryptographic, but limited	Basic, not robust	Strong, but high cost	Basic	Dynamic, patient-driven permissions; optimized encryption
Cost Efficiency	High gas cost	Not measured	Not measured	High computation costs	High blockchain costs	Low gas cost, lightweight contracts
Error Handling/ Recovery	Not addressed	Not addressed	Not addressed	Not addressed	Not addressed	Included (robust key management, recovery protocols)
Multi-Institution Workflows	No	No	No	No	Partial	Supported (proxy server integration)

EHR: electronic health record; HIE: health information exchange; HL7: health level seven; IPFS: InterPlanetary File System.

Table 3. System entities of the proposed framework.

Number	System entities of the proposed framework
1.	Patient
2.	Doctor
3.	Proxy Server

Furthermore, our approach supports efficient, encrypted incremental updates to patient records, addressing a key limitation in previous systems that either lacked real-time update capabilities or incurred high computational overhead. We also explicitly implement interoperability with HL7 FHIR standards, which remains only partially or superficially addressed in most related works. In terms of privacy and access control, our dynamic, patient-driven permission model coupled with optimized encryption techniques enhances data confidentiality beyond basic cryptographic or on-chain access controls found elsewhere.

Additionally, our system uniquely incorporates robust key management and error recovery protocols, often unaddressed in other blockchain HIE frameworks.

Finally, multiinstitution workflows—critical for practical deployment across diverse healthcare providers—are fully supported through proxy server integration, setting our framework apart from most existing solutions that offer no or minimal support for such collaborative environments.

Overall, this comparison underscores the technical advancements and practical benefits of our proposed framework, positioning it as a comprehensive and scalable solution for secure, interoperable, and patient-centric health data exchange.

Proposed Framework

Architecture Overview

We developed an ETH blockchain-based system with multiple smart contract functions to leverage blockchain technology for HIE. The system architecture, shown in Figure 1, includes two smart contracts: the "authorization contract" (main controller) and the "user smart contract." Only one authorization contract is needed to run the system, while the user smart contract is used by both patients and doctors for registration. The number of users in the smart contracts corresponds to the number of patients and physicians in the system. Doctors have a touchpoint to request patient records. To address the limited storage capacity of ETH blockchain nodes, we use a decentralized IPFS network for large file storage. Data are protected in IPFS using asymmetric public and private key pairs generated by the Rivest-Shamir-Adleman (RSA) technique. The same encryption method is used for data transfer from patient to doctor. Detailed descriptions and algorithms are provided in subsequent sections.

Environment Setup

The proposed framework requires the ETH blockchain for deployment. All smart contracts are deployed on the ETH blockchain, and a connection to the IPFS network is established. Upon application launch, an authorization smart contract is deployed to manage all entities. Patients register with the authorization

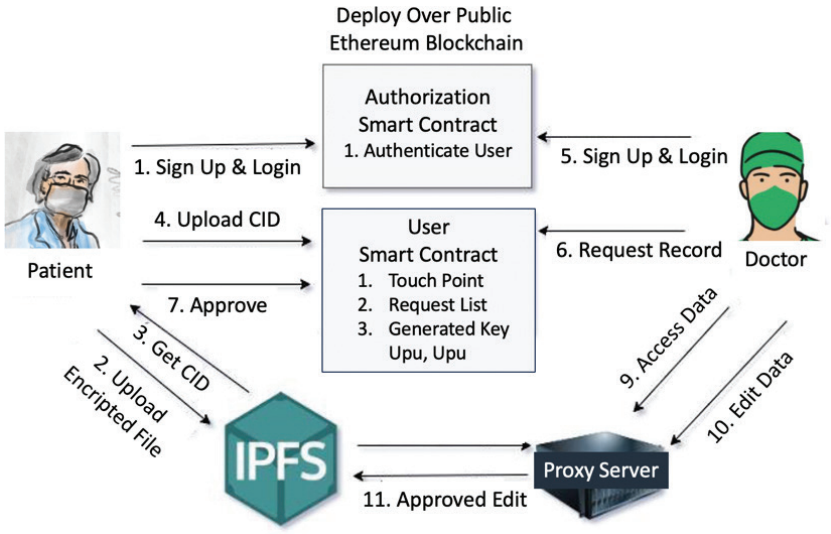

Fig. 1. Proposed HIE framework. HIE: health information exchange, IPFS: Interplanetary File System, Upu: parallel processing unit.

contract, and a user-smart contract is deployed for each patient. Doctors follow the same registration process using the user contract. Asymmetric public and private keys are generated for users and uploaded to the smart contract during deployment. Table 3 shows that the proposed system requires fewer entities compared to earlier frameworks.

Patients and doctors are the two entities using this system to securely exchange health information. The proxy server temporarily stores encrypted data after the patient approves the doctor's request for access. If a doctor wants to make changes, the updates are first saved on the proxy server. The patient is notified of the modifications and must approve them before the data are updated on the IPFS network.

Authorization Smart Contract

The system's primary controller is an authorization smart contract. It includes all of the user's personal data, including their addresses for smart contracts that will be saved in this smart contract. As soon as users log into the system, they must first get authorized using this authorization contract. Once authorized, the address of the user contract returns, allowing the user to access their contract and make updates, deletions, and approvals of doctor requests. Authorization smart contracts have various methods depending on the usage.

add_user(), getSC_hash(), getUser(), isUserExists(), and authenticate()

are the methods provided. The add user method is used to add new users, as the name suggests. This method can only be called by the administrator. The authenticate method is used to verify the user, which is also called by the admin. Get user is called immediately following user authentication, and the user exists method is used to determine whether or not the user exists. The doctor frequently uses the get smart contract hash technique to obtain the patient's contract address so that a subsequent doctor may ask the user for access to the record.

User Smart Contract

This user's smart contract is used by patients and physicians.

addData(), AddKeys(), GetPublicKey(), GetPrivateKey(), GetFileNames(), GetFileHash(), ChangeFileHash(), SendRequest(), GetRequest(), and ApproveRequest()

are the methods that are included in the user contract. The IPFS node hash and file name are added to the contract using the addData() method. The public and private key pairs were also kept in the user contract. The addKeys() function can add this and the user can retrieve these keys using the get methods. It is possible for doctors to access the public key using the getPublicKey method. This contract will also store the filenames, and these filenames can be accessed by getFileNames method. When the doctor is requested to the file using the sendRequest method, that request is stored inside a mapping variable. In addition, patients can get all requests using the getRequest method. And patients can approve the request using the approveRequest method. Once the patient has approved the request for the health data, then the encrypted data are sent to the proxy server. The doctor can update anything on the proxy server; once done, the patient can approve the latest changes and update the old IPFS hash to a new hash using the changeFileHash method.

Citation: Blockchain in Healthcare Today 2026, **8**: 423 - https://doi.org/10.30953/bhty.v8.423

Security Mechanism

The framework employs advanced encryption and secure key management to mitigate vulnerabilities.

- Data Encryption: Health records are encrypted using asymmetric cryptography before being stored on IPFS. Public and private keys are generated for each user to ensure secure access.
- Key Management: Keys are securely managed and stored on the ETH blockchain to prevent exposure. Access to private keys is restricted to the data owners.
- Access Control: Authorization smart contracts enforce granular permissions, ensuring only approved entities can access or modify medical records.

Design Enhancements

To address limitations in existing frameworks, the proposed system incorporates the following enhancements: (1) Decentralized off-chain storage using IPFS to overcome blockchain's storage limitations. (2) Elimination of plain-text passwords or key storage in smart contracts to enhance security. (3) Support for real-time data updates through seamless interactions between IPFS and ETH blockchain components. (4) Scalability to handle concurrent user operations with minimal performance degradation.

Implementation and Analysis

The implementation of our smart contracts uses Solidity. The code is created and deployed with Ganache, tested, and analyzed using Remix Integrated Development Environment. The graphical user interfaces (GUIs) and application programming interface (API) are developed with Node.js and other web technologies. We use the Web3 library to interact with smart contracts and the IPFS-CORE library to handle IPFS. Each component is detailed next.

Ganache

Ganache is a software that provides a local in-memory blockchain.[15] It simulates the functionality of an actual ETH network, including the availability of many test accounts that are financed with actual ETH. Ganache is available in two forms: a command-line tool and a desktop program with a user interface and command-line interface. To implement the framework, we used a Ganache provider with Web3. The gas limit of the Ganache ETH blockchain is set to 6,721,975. And other settings are also kept as default configurations.

Web3.js

We utilized the web3.js library[16] to connect the Node JS application to the Ganache provider. It is a JavaScript library that enables developers to use Hypertext Transfer Protocol (HTTP), interprocess communication, or a web socket to communicate with the local or remote ETH network. Hence, interacting with the blockchain is possible using this library client. We may deploy, send, or call any smart contract method or execute any blockchain transaction using this library.

IPFS-Core

The IPFS basic functionality is provided by the IPFS-core library collection for a variety of programming languages, including JavaScript, C#, VB, F#, etc.[17] Without the need to launch external processes or control API ports and servers, the fundamental objects of IPFS are designed to be utilized to run an IPFS node as a component of your application. A collection of composable peer-to-peer protocols called the IPFS Core API is used in decentralized file systems for addressing, routing, and transferring content-addressed data. It enables the development of fully distributed applications and seeks to speed up, secure, and open the web. We use this library to upload, read, and update documents to the IPFS network.

System Functionality

Users must enroll for the authorization smart contract in order to participate in the HIE blockchain. Many transactions are completed during the registration process, which makes it costly. Initial patient contract deployment involves some gases, and subsequent key addition and controller entry also require gas consumption. That has to be done once for all the new users.

We generate an RSA key pair with a 2048-bit modulus and a 0x10101 public exponent. The SubjectPublicKey Info (SPKI) and Privacy-Enhanced Mail formats are used for the public key, based on the public key encoding option. The private key should be encoded in PKCS#8 format, according to the private key encoding option. Appendix A[8] shows the sample-generated key pair; later it will be added to the patient contract. Algorithm A shows the complete step-by-step flow of registration. We created the user interface to register and log in (Appendix A). The user must choose their role—patient or doctor—and complete all other necessary information during registration. After successfully registering, the user can log in using their credentials. The dashboard of the patient and the dashboard of the doctor are shown in Figures 2, 3, 4. The patient is not required to remember their account identification (ID) or contract hash, which solves the issue with a previously suggested framework that requires users or patients to remember them.

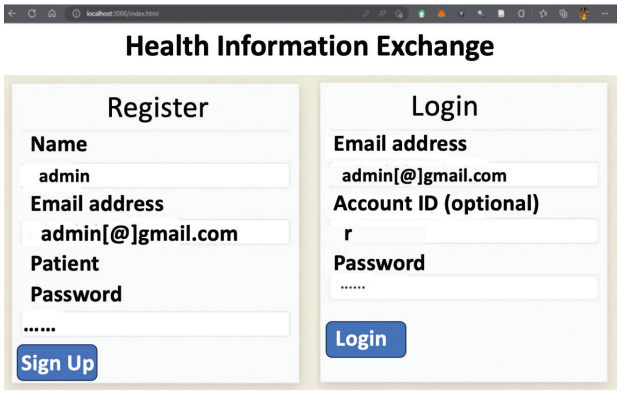

Fig. 2. User login/signup interface.

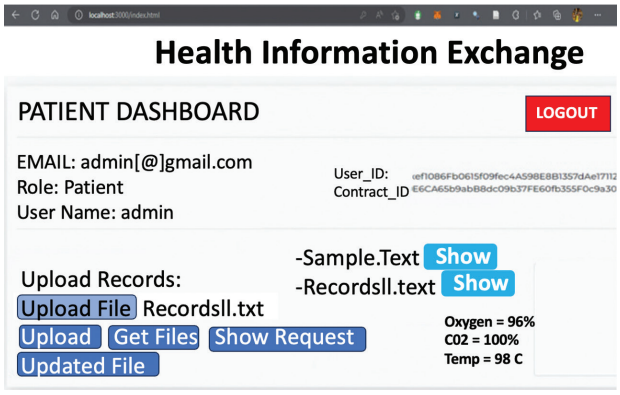

Fig. 3. Patient dashboard interface.

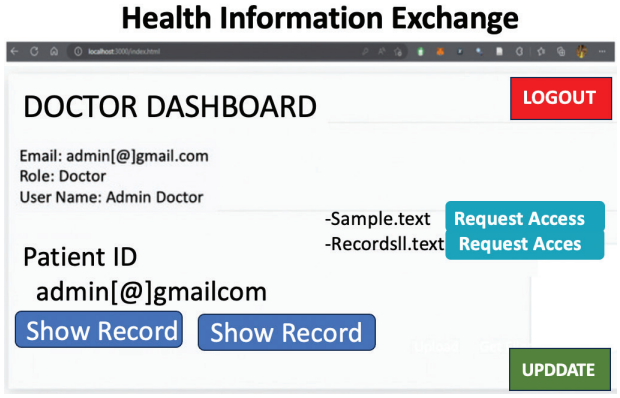

Fig. 4. Doctor dashboard interface.

When a user logs into the system, the authorization contract returns their account ID and contract address since everything is stored inside the authorization contract of the proposed model.

After successfully logging into the system, the patient has the option to upload their records to the IPFS network. The user can read and delete records from the IPFS network once they have been uploaded. In addition, the patient may view the doctor's request for a specific file. Doctors can only view the patient records when the patient approves their requests. Doctors can further request for their records only when they have access to the patient's ID. Once the patient ID is authenticated using the parallel processing unit, the doctor is granted access to the touchpoint, where they may choose the files they want to access. But at this time I am unable to access the file. Only with the patient's consent may they view the records. The record will be stored on the proxy server in an encrypted form after the patient authorizes the request. Once patients store the updated information, doctors can read and update records on the proxy server and then send update requests to patients. Only after the patient approves the request is the updated record stored on the IPFS network.

The core system workflows of the proposed HIE framework are formalized using Algorithms 1–5. Algorithm 1 describes the user registration and smart contract deployment process. Algorithm 2 presents the procedure for encrypting and uploading medical records to the IPFS network. Algorithm 3 outlines the record access request workflow initiated by doctors, while Algorithm 4 details the patient-side approval mechanism. Finally, Algorithm 5 illustrates the secure access of medical records by authorized doctors.

Algorithm 1. Registration user.

```
1: INPUT: Patient Detail
2: If User_ID Not Exists:
3:    Generate Patient Asymmetric public (Ppu) and private (Ppr) key pair;
4:    Deploy Patient smart contract for the user;
5:    Store the generated key pair to the patient contract;
6:    Call addUser method of authorization contract;
7: Else:
8:    return "User already Exist";
9: End If;
10: Return true;
```

Ppu: Patient public key, Ppr: Patient private key.

Algorithm 2. Upload Record.

```
1: INPUT: File (data).
2: REQUIRE: Owner of Contract (i.e., Patient)
3. If is_Logged_In():
4:    Get Patient Public Key (Ppu);
5.    Encrypt file using Ppu i.e., encData = Enc(data, Ppu);
6.    Upload Encrypted file to the IPFS network;
7.    ADD IPSH hash to the user contract;
8. Else:
9.    return "Invalid user";
10. End if;
11. Return true;
```

IPFS: Interplanetary File System, Ppu: Patient public key.

Citation: Blockchain in Healthcare Today 2026, **8**: 423 - https://doi.org/10.30953/bhty.v8.423

Algorithm 3. Request record.

```
1: INPUT: File Name, Patient_UserID, Doctor_UserID
2: If is_Logged_In(Doctor_UserID):
3:    If  is_user_exist(Patient_UserID):
4:       call getSC_hash();
5:       call SendRequest();
6:    Else:
7:       return "Invalid User";
8:    End if;
9: Else:
10:   return "Invalid User";
11: End if
12: Return true;
```

Algorithm 4. Approve request.

```
1: INPUT: File Name, Patient_UserID, Doctor_UserID
2: If is_Logged_In(Patient_UserID):
3:    If is_user_exist(Doctor_UserID):
4:       call ApproveRequest();
5:       Get Doctor Public Key (Dpu);
6:       Re-encrypt with the Dpu
            encData = Enc(Data, Dpu);
7:       Send re-encData to the Proxy Server.
8:    Else:
9:       return "Invalid User";
10:   End if;
11: Else:
12:   return "Invalid User";
13: End if
14: Return true;
```

Dpu: Doctor public key.

Algorithm 5. Access record.

```
1: INPUT: File Name, Patient_UserID, Doctor_UserID
2: If is_Logged_In(Doctor_UserID):
3:    If is_user_exist(Patient_UserID):
4:       Get EncData from the server
4:       Data = Decrypt(EncData, Dpr);
5:       return data
6:    Else:
7:       return "Invalid User";
8:    End if;
9: Else:
10:   return "Invalid User";
11: End if
12: Return true;
```

Dpr: Doctor private key.

All of the algorithms provided refer to the user ID, patient UserID, or doctor UserID as a single, distinctive identifier; in our implementation, this identifier is an email ID. The user cannot be registered again with the same email ID. Users can upload multiple files with different sizes. Doctors may also request multiple files from the user. Additionally, doctors may request multiple patients to share their records. However, a doctor can only access the touchpoint of a single patient. The same applies to patients, for example, patients cannot see more than one request from multiple doctors at a time. A patient can share single records with multiple doctors. However, if both doctors have modified the content of the files, it will save only the last doctor-modified data. This system is designed to help doctors and patients manage medical records more efficiently. By allowing users to upload multiple files with different sizes and enabling doctors to request multiple files from patients, this system makes it easier for doctors to access patient records and provide better care. Patients can also share their records with multiple doctors, which helps ensure that all healthcare providers have access to the same information. However, this system is designed to ensure that only one doctor can access a patient's touchpoint at a time, which helps to protect patient privacy and prevent unauthorized access.

For users, a GUI (a Graphical User Interface) is provided to interact with the DApp. While building an application, the AJAX (Asynchronous JavaScript and XML) framework is used to communicate data between the front end and the back end, and all transmission happens in the JSON format. The GitHub repository has the Smart contract for the suggested framework.

https://github.com/Rohit-dev-coder/HIE_SmartContracts

Analysis and Results

Two smart contracts have been created, deployed, and used. Table 4 lists the transaction costs for the operations and the miner's execution costs. All the transaction costs and the execution costs are comparable and less than the work of Madine et al.,[8] which was the last article based on the public ETH blockchain and IPFS network. A framework that was previously presented by numerous authors uses more gas than paper.[8] In the proposal framework, only the submit record function used considerably more gas; the second time the patient's record was submitted, the transaction cost and execution cost were about 71,445 and 50,007 INR, respectively. As we intended, adding a patient and a doctor imposes just execution costs. Updating files, getting

file hashes, and getting names are some new methods that are introduced in the proposed framework. With the use of blockchain technology, the proposed framework aims to make managing medical records more effective. This framework can assist in lowering expenses and enhancing security by utilizing smart contracts to govern interactions between patients and doctors. In comparison to earlier frameworks based on the public ETH blockchain and the IPFS network, this framework has lower transaction costs and execution costs. To make this framework more versatile, new methods such as updating files and obtaining file hashes and names have been included.

We used JMeter to test the execution time of the application. JMeter is a performance testing tool that is open-source and available as well. To evaluate the durability of servers, networks, or other objects or to assess overall performance under various load types, it may simulate significant loads on such items. We built a simulation of 50 users, calling the patient registration procedure at a specific moment. The simulation is done with the following system configuration: Intel® Core™ i5-7200U Processor at 2.50 or 2.70 GHz, with 8 GB of RAM. The execution time of the transaction is shown in Figure 5. Almost 12 to 14 s pass between transactions on average, which includes time of generating and storing keys, deploying the user contract, and adding details to the authorization contract. Only 1 of transactions took longer than 14 s; the majority were completed in less time. Since blockchain is a repository, the proposed blockchain system consumes less memory than traditional systems when comparing memory consumption.

Table 4. Gas of smart contract function.

Function name	2020 Cost (000, INR)[8]		Proposed Cost (000, INR)	
	Transaction	Execution	Transaction	Execution
Controller contract	872	621	457	376
Patient contract	1,193	862	705	609
Add patient	46	25	-	12
Add doctor	46	25	-	12
Submit record	81	57	89	67
Request record	141	119	91	70
Respond to request	38	16	29	8
Update file	-	-	27	6
Get file hash	-	-	-	3
Get file names	-	-	-	10

Gas: fee paid in cryptocurrency to compensate the network for computational resources.

The amount of time needed to upload the files to the IPFS network is shown in Figure 6. Obtaining the public key, encrypting the file, uploading it to IPFS, and storing the hash in the user contract all take time. Time is increased since several techniques are used when uploading. We evaluated the uploading speed of files with sizes of 13, 26, 103, 512, and 1,024 kilobytes. The time it takes to upload a file increases as its size does. Because larger files demand more computing power and bandwidth to transfer, the uploading process takes longer.

The HIE system analyzes if it is possible to communicate safe medical data using the suggested HIE system and successfully responds to security concerns. Medical data can be obtained by an external attacker via a sniffing or eavesdropping assault when they are exchanged across a network. If medical information is compromised, the patient's electronic medical record privacy may also

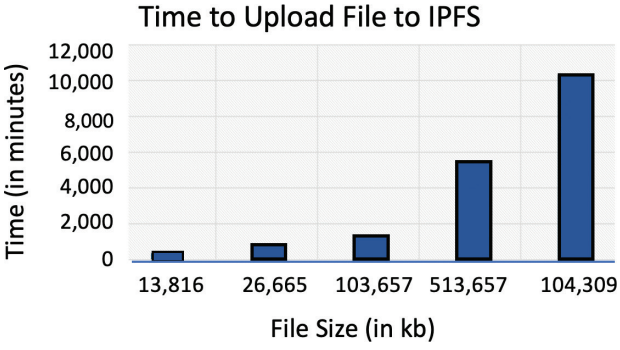

Fig. 6. Time to upload record to IPFS (Interplanetary File System).

Statistics

Requests	Executions			Response Time (ms)							Throughput	Network (KB/sec)	
Label	#Samples	Fail	Error %	Ave.	Min.	Max.	Median	90th pct	95th pct	99th pct	Tranactions	Received	Sent
Total	50	0	0.00%	12253.38	10657	14139	12223.50	13320.70	13951.10	14139.00	0.03	0.01	0.01
HTTP Request	50	0	0.00%	12253.38	10657	14139	12223.50	13320.70	13951.10	14139.00	0.03	0.01	0.01

Fig. 5. The execution time of the transaction (see text for a fuller discussion).

Citation: Blockchain in Healthcare Today 2026, **8**: 423 - https://doi.org/10.30953/bhty.v8.423

be compromised.[18] The proposed HIE system encrypts medical data using a unique encryption key to ensure that it may be shared safely. The information included in the encrypted health data is kept private even if the data are stolen since only the patient or a user authorized by the patient may decode the data. The attacker can try to attack directly to the IPFS network, but even if they get a file hash or get access to the content, they cannot read the file because the file is stored in an encrypted file using the cryptographic hash.

Security Evaluation

Our system employs a comprehensive security strategy combining cryptographic safeguards and smart contract-enforced permissions. The RSA encryption is applied to all patient data before off-chain storage in IPFS, ensuring that no plaintext data are exposed externally. Smart contracts govern access based on patient-driven authorization, and each access request is cryptographically verified, effectively preventing unauthorized data retrieval. The framework also incorporates coding best practices to harden contracts against known vulnerabilities such as re-entrancy and integer overflow. Testing of unauthorized access attempts confirmed that the system successfully denies all access lacking proper permissions, demonstrating robust protection mechanisms.

Error Handling and Recovery

To maintain data integrity and reliability, our framework integrates several error management features. ETH blockchain transaction failures trigger automated retry protocols to preserve synchronization between blockchain states and off-chain IPFS data. Additionally, a key management recovery process supports patients in restoring access in the event of key loss or compromise, utilizing proxy server intermediaries for secure credential restoration. Simulated scenarios of transaction interruptions and key loss verified that these mechanisms function effectively without risking data loss or breach.

Interoperability Testing

Explicit support for HL7 FHIR standards underpins our system's interoperability with existing healthcare infrastructures. We modelled patient data as standardized FHIR JSON bundles, which were transmitted and manipulated within our framework without loss of structural integrity, as validated in a controlled integration test environment. The modular API architecture allows seamless interaction with legacy EHR systems and federated provider directories, illustrating the framework's readiness for practical deployment and data exchange within heterogeneous healthcare ecosystems.

Security and Threat Analysis

Protecting patient data is critical in any healthcare system, and this is especially true for blockchain and IPFS-based HIE frameworks due to the sensitive nature of medical records. In designing our system, we carefully considered potential security threats and incorporated multiple layers of defense to safeguard data privacy and system integrity.

We recognize several key risks that our framework must address, including man-in-the-middle attacks where communication might be intercepted or altered; Sybil attacks where malicious actors create fake identities to manipulate the network; insider threats such as collusion among healthcare providers aiming to access data they shouldn't; denial-of-service attacks targeting blockchain nodes or IPFS storage to disrupt service; replay attacks that attempt to reuse valid transactions maliciously; and common smart contract vulnerabilities like re-entrancy and integer overflow errors.

To guard against these, our system encrypts all patient data off-chain using RSA encryption before storing it in IPFS. Every access or update request is cryptographically signed and validated on the blockchain, making it extremely difficult for an attacker to impersonate a legitimate user or tamper with data undetected. Access control is tightly managed through smart contracts that place the patient at the center of permission decisions, minimizing risks from collusion or unauthorized data sharing. Our contracts are written with security best practices in mind—protecting against well-known vulnerabilities such as re-entrancy attacks and ensuring safe arithmetic operations.

To reduce Sybil attacks, our system requires robust identity verification for network participants, ensuring that only verified healthcare providers and patients can interact with the system. We also incorporate gas limits and efficient contract logic to minimize the chances of denial-of-service attacks on the blockchain, while IPFS's inherent redundancy helps to resist node failures or overloads.

Replay attacks are prevented by embedding nonce or timestamp-based checks within transactions to ensure each request is unique and cannot be maliciously replayed. Furthermore, we regularly verify data hashes stored on-chain against the actual data in IPFS to detect any tampering attempts, with automatic synchronization protocols to maintain data consistency.

While comprehensive penetration testing is ongoing, our initial security assessments show that unauthorized access attempts are effectively blocked. Automated vulnerability scans detected no critical issues in our smart contract or key management setup, providing confidence in the robustness of our design. Considering the stringent privacy and regulatory requirements

in healthcare, the security of patient data is non-negotiable. Our multifaceted approach balances the openness needed for data sharing with the strict controls necessary for confidentiality, ensuring patients can trust the system while enabling healthcare providers to access data securely and efficiently.

Scalability and Usability

Scalability and usability are essential for the long-term success and practical adoption of any HIE system. While our experiments tested up to 50 concurrent users, the underlying architecture is designed to support much larger scales. By leveraging IPFS for off-chain data storage, the framework reduces on-chain bottlenecks, enabling horizontal scaling of storage nodes. The use of lightweight smart contracts further optimizes transaction throughput and cost, which are critical factors when extending to hundreds or thousands of simultaneous users.

From a usability perspective, the system's patient-centric permission model simplifies control over data sharing, allowing users to easily grant or revoke access without requiring specialized technical knowledge. The minimal transaction fees and moderate execution times help to ensure that responsiveness remains suitable for routine clinical workflows. Informal feedback from preliminary user interface tests with healthcare professionals indicated that the interface aligns with their day-to-day needs and reduces the burden of managing permissions manually.

Although blockchain transaction delays can be a concern, the average transaction time of around 12 to 14 s may be acceptable in many clinical contexts where immediate real-time response is not critical. Ongoing improvements in blockchain infrastructure and potential layer-2 scaling solutions offer promising avenues to further reduce latency, improving overall user experience. Future work will involve stress testing with larger user populations, as well as pilot deployments in clinical environments to gather detailed usability data and fine-tune system parameters.

Limitations and Future Work

Our proposed HIE framework moves us closer to secure and scalable sharing of medical data, but it is not without its shortcomings. Several areas will need further attention before such a system could be deployed at scale in real healthcare environments.

Security Threats

Even with strong encryption, patient-controlled access, and verified participant identities in place, no system can claim to be completely immune to attack. Large-scale denial-of-service attempts on the ETH network or IPFS nodes, coordinated insider misuse, and unforeseen vulnerabilities in blockchain protocols remain possible risks. Addressing these will require ongoing monitoring, periodic penetration tests, and independent code and security audits to keep pace with new threats.

Scalability and Performance

Our current evaluation, involving up to 50 concurrent users, offers only a glimpse of how the system behaves under load. Real-world health networks may involve thousands of transactions happening simultaneously, and handling that scale will demand further optimization—whether by refining smart contract code, batching transactions, or adopting layer-two scaling techniques. More extensive load tests with larger datasets will be an important next step.

Interoperability and Integration

While we have modeled patient records using HL7 FHIR resources, connecting seamlessly with the variety of EHR systems in use today brings its own challenges. Differences in how hospitals implement EHRs, varying patient ID schemes, and local data policies mean our APIs and data-mapping processes will need to be highly adaptable. Middleware that can bridge these differences may prove essential for practical rollout.

Usability and Adoption

Finally, a technically sound system does not automatically translate into one that patients and clinicians will adopt. Usability testing, refining consent workflows, and ensuring the system satisfies regulatory and compliance requirements will be critical in building trust. Training and outreach will also help stakeholders understand the benefits of a decentralized HIE platform.

By being transparent about these limitations and setting out a path to address them, we hope to encourage further development and real-world piloting of blockchain-enabled health data exchange systems.

Conclusion

This article presents a decentralized, transparent, and secure blockchain-based framework that empowers individuals to control their medical information. By leveraging ETH smart contracts, we enable automated and trustworthy authorization and user management for health data exchange. Our system integrates IPFS, proxy servers, and reputation mechanisms to safely store, retrieve, and share patient records while supporting broad interoperability across healthcare networks.

We evaluated the proposed solution within a HIE context, demonstrating reduced gas consumption compared to existing approaches. While recognizing some

Citation: Blockchain in Healthcare Today 2026, **8**: 423 - https://doi.org/10.30953/bhty.v8.423

limitations, our framework is designed with flexibility to function on both permissioned and permissionless blockchain platforms. This study contributes a novel model that addresses key challenges in managing healthcare data with scalable off-chain storage. Future work will focus on conducting real-world pilot deployments and comprehensive scalability evaluations to rigorously assess the system's effectiveness and operational viability across diverse clinical settings.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Contributors

All authors contributed to the conception, design, implementation, analysis, and manuscript preparation. All authors reviewed and approved the final manuscript.

Data Availability Statement (Das), Data Sharing, Reproducibility, and Data Repositories

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Application of AI-Generated Text or Related Technology

No AI-generated text or related technologies were used in the preparation of this manuscript.

References

1 Zhuang Y, Sheets LR, Chen YW, Shae ZY, Tsai JJ, Shyu CR. A patient-centric health information exchange framework using blockchain technology. IEEE J Biomed Health Inform. 2020;24(8):2169–76. https://doi.org/10.1109/JBHI.2020.2993072

2 What is Ethereum 2.0? (investopedia.com) [cited 2025 Jun 23]. Available from: https://www.investopedia.com/ethereum-2-0-6455959

3 Mohammed MK, Abdullah AA, Abod ZA. Securing medical records based on inter-planetary file system and blockchain. Period Eng Nat Sci. 2022;10(2):346–57. https://doi.org/10.21533/pen.v10.i2.612

4 What is the interplanetary file system (IPFS) and how do you use it? Available from: www.howtogeek.com/784295/what-is-the-interplanetary-file-system-ipfs/

5 Zhuang Y, Sheets L, Shae Z, Tsai JJP, Shyu CR. Applying Blockchain Technology for Health Information Exchange and Persistent Monitoring for Clinical Trials. AMIA Annu Symp Proc. 2018;2018:1167–1175. PMID: 30815159; PMCID: PMC6371378.

6 Li H, Han D. EduRSS: a blockchain-based educational records secure storage and sharing scheme. IEEE Access. 2019;7:179273–89 [cited 2025 Jun 23]. https://doi.org/10.1109/ACCESS.2019.2956157

7 Nguyen DC, Pathirana PN, Ding M, Seneviratne A. Blockchain for secure EHRS sharing of mobile cloud based e-health systems. IEEE Access. 2019;7:66792–806. https://doi.org/10.1109/ACCESS.2019.2917555

8 Madine MM, Battah AA, Yaqoob I, Salah K, Jayaraman R, Al-Hammadi Y, et al. Blockchain for giving patients control over their medical records. IEEE Access. 2020;8:193102–15. https://doi.org/10.1109/ACCESS.2020.3032553

9 Sharma B, Halder R, Singh J. Blockchain-based interoperable healthcare using zero-knowledge proofs and proxy re-encryption. In 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS) (pp. 1–6). IEEE; 2020.

10 Jabbar R, Fetais N, Krichen M, Barkaoui K. Blockchain technology for healthcare: Enhancing shared electronic health record interoperability and integrity. 2020;310–317. https://doi.org/10.1109/ICIoT48696.2020.9089570.

11 Mani V, Manickam P, Alotaibi Y, Alghamdi S, Khalaf OI. Hyperledger healthchain: patient-centric IPFS-based storage of health records. Electronics. 2021;10(23):3003. https://doi.org/10.3390/electronics10233003

12 Pawar P, Parolia N, Shinde S, Edoh TO, Singh M. eHealthChain—a blockchain-based personal health information management system. Ann Telecommun. 2022;77:1–13. https://doi.org/10.1007/s12243-021-00868-6

13 Jayabalan J, Jeyanthi N. Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy. J Parallel Distr Com. 2022;164:152–67. https://doi.org/10.1016/j.jpdc.2022.03.009

14 Bae YS, Park Y, Lee SM, Seo HH, Lee H, Ko T, et al. Development of blockchain-based health information exchange platform using HL7 FHIR standards: usability test. IEEE Access, 2022;10:79264–71. https://doi.org/10.1109/ACCESS.2022.3194159

15 Ganache—Truffle Suite [cited 2025 Jun 23]. Available from: https://trufflesuite.com/Ganache/

16 Web3.js. Javascript Ethereum API (web3js.org) [cited 2025 Jun 23]. Available from: https://web3js.org/#/

17 IPFS in JS. IPFS Docs [cited 2025 Jun 23]. Available from: https://docs.ipfs.tech/reference/js/api/

18 Lee S, Kim J, Kwon Y, Kim T, Cho S. Privacy preservation in patient information exchange systems based on blockchain: system design study. J Med Internet Res. 2022;24(3):e29108. https://doi.org/10.2196/29108

Appendix A. Asymmetric public and private key pair.

```
----------------------SINGUP METHOD---------------------

DeploySC: Default Account: 0xef1086Fb0615f09fec4A598E8B1357dAe171129c
DeploySC: Contract Address: 0xaB0E6CA65b9abB8dc09b37FE60fb355F0c9a30da
{
  publicKey: '-----BEGIN PUBLIC KEY-----\n' +
    'MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzDyD2fwgfLxJkyRRDFbd\n' +
    'QV8TIWHssQ3w0ddFPre4KHPo9XH5bJXsw9PDigMESeMbE0C8iYtEJ02WwxQNGa2m\n' +
    '4EbkrXiRk+QqhBwtGh8De+Lpu3aPLU85CvpU0MwjHKh1U1DmqZJ2NeFQ4gKu39hE\n' +
    'ChA1LHfkJ6lftCz4f2WVV7Qzaa+EnrFt/Y9d/IQ231CvWKzM/mHa4ierYCuo5bWP\n' +
    'P+HkOFZQfDoS4iJKvxrv7te2xSg2qa42NFVncJcE/UCtpR/y7DkeTGVFKPemX6g7\n' +
    '06H2ylkXf6kh8LU5JvWwuaMW5aoeJfK/aFic3U7P/vlePZkahy5xGyl6IYskUp6P\n' +
    'fQIDAQEB\n' +
    '-----END PUBLIC KEY-----\n',
  privateKey: '-----BEGIN ENCRYPTED PRIVATE KEY-----\n' +
    'MIIFLTBXBgkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQITClef4934PECAggA\n' +
    'MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBBFt0VYCzC3YxfcnKpGDDt1BIIE\n' +
    '0KvSjPOLRKOQIpkBLK+TJAFPwEfLv5XEdKp7NUv+Xn6BVlP2YjOSJIT6nrRqpzYd\n' +
    'DaxZeYd+nDXdgRDP69F8qIotjZCeyrilFmH41N/3fk7hgQsvpM00Qplml+Bi8BGw\n' +
    'TDs3duSj+oF/f44aoH55SYerWCNOVE3g10nGmmyNqlICA/I04HmcjzX/IHTFr/n8\n' +
    'RwAkgpixmZcBLMnu2hI+/c/FrrB6FEM8+xIJ/0+N1uhyAVaAL+/Xzd+Yb6A05QcF\n' +
    'c8p0xGpmtGVOkDOcMuPDQy3kqT71/dtyFBBSezQb8E2MFqdKy1Nxj6SC4hP/d5uf\n' +
    '1KlWbCFPFrvlf/Ls8ZDxIwUlbgpLb0I01Cv2+0rPSYE4yMWD73O/xZYxcRHAGFqn\n' +
    'WlrhePgs/D0H8Jz/ZiT0h+luscdUQAaiu6wQDBSdF9EU2f4Bf41QC22NLZBcp3SK\n' +
    'Umy4+w111xalNLxkUtTP3wRsASCsU351pnlZhISQsdkLY5qaKHIwWL7N8lnrxYOz\n' +
    'X8IWgCIm328aREt0QE3WqZNKmsw1BX/Lt7ypiQwWhoG7iNyn7XPlWoLo2bdW5bGM\n' +
    '4792rzkHuTIjreiFOag5M5H7tz6NDTiQHZ93tfxbly787t++8//DzS7TpFXCwjHW\n' +
    '1NUHH50kfVcifMWeonNOK1YFI2GFKmX0ZlYmxl5pcceMzk0Xa0RwOtrwo0a3cD9b\n' +
    'aJKYqyol9ysqXKh5O2g3sW1F3PT+1L7sPLIHp8/1qnQgqa1FwKB5YexCxnlUC3Wo\n' +
    'ibwi2a/bC4jk0M5/QVwr3MPOPwVKxFYaruVRlXSSy+8oEx+I/zghPcF+e9bJddc0\n' +
    'gJe0ZOV9wiLI6IcGEwXj15LHdUv8cBzulQhOroUUBIqosAxUVABebBAuLMC1gMqp\n' +
    '77UFJ9J4PJ0oL//hdorrGpLQHxGd3x0duS2cRrHSRoNUDzddMSWmJ46gGmqgu5Jf\n' +
    'WtQh0WozSnFycFiMeVPVdghumo3Nc7tg+XsfHnJP3Ci2esXPydL6Qgjm0/MMFvir\n' +
    '4FDHZomCtQYF0jio8hDDwU0E095//NYxDX7od75j2ng6nL4eE3elzFiu4PpxM4lB\n' +
    'PqIvdQTH9Rt6oOc6jmXFNyYU88Z7enjsZSobvolTPMsvhJChuh3q2BZhqn1eq1qy\n' +
    'CkzyDk1i7lLY/lRYmYNacCxk3UHzLLXKW9E40DlRAxOB0EYAvXq1+pdBldSVtEeK\n' +
    'BB/Qfq1vld3yNOyWRp8dYMbU//lK31xYBA+4E7AIFhnF5MVEM8W6dRhfw0GColAF\n' +
    'mlgg96VSh8D5ZlePzf++bSrx8eNqUzSU5Mc8rnr/m8edhqxzXmK6A5tgyQ8R7Jfz\n' +
    'zM0XwBVE38wySumKBJkpHWMaiHs58zRx1s/c2PCT70RPDDX/Cha0W740/Yugr1V9\n' +
    'AC3GVOhVYR6KXEvAeHdJaJicPAi1J9qxBbSn0LXuSDTxoCbp8kbHDu8cls4xxOHY\n' +
    'IjUxxIdhoDu/ksQgD7qt9A0h36noAsgWkKZCoStkV8agz78zCNX3C0p+8MEchmpj\n' +
    'aHL08qFE4qd+pFce7i8jcXKqBBzEhJ7d8izere0EMcJ5G+CWB5Mekj6O5TgHb1Gh\n' +
    '4SA/C1F+F2A/joVTQn4XA+G50tcHT7PHK9m1ENxXnpLv\n' +
    '-----END ENCRYPTED PRIVATE KEY-----\n'
}
Keys Added
```


Citation: Blockchain in Healthcare Today 2026, **8**: 423 - https://doi.org/10.30953/bhty.v8.423