



# Blockchain-Based Secure and Interoperable Healthcare Record Management System

Abhishek Yadav<sup>\*1</sup>, Shikha Kamal<sup>1</sup>, Swaroop Kumar<sup>1</sup> and Sonia Sonia<sup>1</sup>

<sup>1</sup> Department of Computer Science and Engineering, Chandigarh University, Punjab, India  
\*24BCS12724@cuchd.in

**Abstract.** Modern-day medical care is performed through EHRs (electronic health records). These systems have caused a major change in the way that physicians keep records. Most of today's EHRs incorporate centralized systems. The centralized system introduces major security issues, breaches of patient privacy, and limits on how patients may control their own medical information. This research introduces a blockchain-based healthcare framework for security and interoperability utilizing a Global Patient Identifier (GPID). The proposed healthcare framework contains smart contracts and encryption methods to ensure that the data is valid, to establish access control, and to securely share the data within the healthcare community. An additional feature of the research presents a version-controlled mechanism to address and resolve record data conflict issues that occur when two or more parties are updating the same record at the same time. Evaluation of the study was accomplished using a Private Ethereum Blockchain Network and simulated in real time from multiple sites; the results show the successful identification of conflicting updated records, that the data is kept in a consistent manner and provides a greater level of transparency and security to healthcare providers throughout the world. Detection rate of conflicts in 50 test cases where updates occurred concurrently was 100%, while the average delay for transaction was 1.2 seconds in our experiment without any data inconsistencies reported. These results demonstrate that the proposed framework has strong potential as a reliable and efficient solution for healthcare data management under evaluated conditions.

**Keywords:** Blockchain, EHR, Healthcare Data Security, Smart Contracts, Data Interoperability, Global Patient Identifier, Conflict Detection, Version Control, IPFS, Decentralized Storage.

## 1 Introduction

Almost all areas of health care adopt the use of Electronic Health Records (EHRs) because they allow healthcare professionals to control patient information using the latest technology, such as cell phones and computers. Rather than keeping their clinical information in a manual form on paper, hospitals keep patient information in digital format. The information stored in the record includes aspects such as medical history, results from tests conducted, and treatment that has been prescribed to

patients. Due to this digitization of medical information, it becomes easy for doctors and other healthcare professionals to update information whenever necessary, thereby improving the quality of patient care [1, 12]. At present, the majority of health record management systems are based on a centralized structure in which patients' records are stored at one place, for example, hospital databases. Although this system makes health record management easier for hospitals, the issue of privacy and security becomes a problem because of potential attacks on the hospital infrastructure. Any technical or human mistake can make sensitive patient data vulnerable [6, 10]. Furthermore, each healthcare organization uses its own electronic systems to store patient data; therefore, exchange of patients' records between different hospitals can prove to be quite troublesome. Since there are no standardized systems for obtaining information, patient medical data has to be exchanged manually through the network. Such actions pose a threat to security and can negatively affect the continuity of the treatment process [4]. To address these existing problems, researchers have proposed the use of blockchain technology as a viable solution. Blockchain is a distributed ledger technology in which data is not controlled by a single administrator but is instead decentralized across the network. Transactional records are stored across multiple nodes, ensuring that data cannot be updated without permission, thereby providing security, transparency, and immutability [1, 2, 11].

### 1.1 Problems in Existing Healthcare Frameworks

Although electronic health record systems have eased the management of healthcare data, several critical challenges remain in current healthcare infrastructures.

**Concurrent Data Modification.** The problem that arises in the context of present healthcare systems is that there is no capability provided for managing changes made simultaneously in the same patient's record by different individuals. This could lead to conflicts of changes made at the same time on the patient record and, as a result, inconsistencies in data. For example, if someone is trying to update a patient's history while simultaneously issuing a prescription for him/her, one process will override the other, while nothing will indicate an error in doing this. There are no frameworks that would guarantee to identify any conflict between the patient's records and help resolve it utilizing the power of blockchain [3–5].

**Information Asymmetry.** The issue of patient data being managed mainly by hospitals and other medical facilities, while patients have very little access to their medical records, is quite common in most healthcare settings, making it difficult for patients to exchange their information with other healthcare facilities.

**Interoperability Issues.** Each healthcare entity operates under domain rules, constraints, and regulations with respect to clinical data management. Data interoperability between organizations, therefore, presents major problems. Due to the absence of any common standard of information extraction, care continuity may become an issue, making it difficult to coordinate care [4].

**Data Security and Privacy Risks.** In any centralized system of healthcare, there is always a threat from hackers or data breaches. In case an unapproved individual gains

access to such a centralized system, the entire private medical information stored can easily get exploited [6].

## 1.2 Motivation and Research Gaps

An analysis of currently existing blockchain healthcare solutions such as MedRec [3], FHIRChain [2], OmniPHR [4], and HealthBlock [9] shows that although the mentioned frameworks contribute significantly to security and interoperability, all of them suffer from four critical flaws.

**No Solution for Simultaneous Update Conflicts:** None of the frameworks under consideration offers a solution to manage conflicting simultaneous updates of the same patient record from several healthcare providers. This scenario is quite common and reasonable; however, it has not been solved so far.

**Lack of Scalability:** Majority of the examined solutions utilize a flat architecture of a blockchain network with only one layer which does not ensure proper scalability. With an increase in the number of healthcare providers, problems with latency and throughput arise.

**No Standardized Patient ID:** The absence of a standardized patient ID is a major issue for cross-institutional healthcare data exchange. Every patient who uses services provided by several healthcare organizations has no ID that can be used in all cases.

**No Integration of Off-Chain Storage with Conflict Detection:** Earlier approaches have been either direct on-chain storage, which is expensive and time consuming, or off-chain storage without any conflict detection mechanism. No earlier model uses IPFS off-chain storage along with version control and conflict detection on the chain.

## 1.3 Key Novelty of This Work

While existing frameworks like MedRec, FHIRChain, and OmniPHR independently address issues related to security, access control, and data sharing, none of them addresses the aforementioned research issues altogether. What makes this paper novel are its three unique contributions. The first one is a unique approach to building a conflict detection mechanism using versioning as part of a smart contract implementation. By integrating conflict detection into the smart contract itself, we achieve a mechanism that is guaranteed to be atomic and tamper-proof without using any external middleware.

The second innovation is related to the introduction of a globally consistent hierarchical multi-layer structure for Global Patient Identifier (GPID). As opposed to MedRec's local approach to access control at the level of separate institutions, our GPID provides a globally-consistent identifier across healthcare organizations.

Finally, another innovation we contribute is the combination of using IPFS technology with conflict-aware hash references in our on-chain transactions. Large medical records can be efficiently stored off-chain while still maintaining their version consistency in on-chain transactions.

## 1.4 Summary of the Proposed Framework

In order to overcome the four research gaps mentioned above, the current research proposes a blockchain framework that ensures the safe, interoperable, and robust healthcare record management. Every patient will be provided with a unique identifier called the Global Patient Identifier (GPID), which will make record identification possible among different healthcare facilities [4, 8]. The version-controlled record management system will help deal with the data conflicts arising from the simultaneous updating of a particular patient's records by various healthcare providers [3, 5]. Access controls, data verification, and data integrity maintenance will be managed using smart contracts [3]. In order to make the proposed framework more scalable and efficient, the hierarchical blockchain architecture will be implemented, where healthcare systems will be organized hierarchically [7, 8].

## 1.5 Our Contributions

The following is a summary of this study's main contributions, each directly addressing one or more of the identified research gaps.

**Contribution 1 (Addressing Gap 1):** A conflict-aware blockchain framework with a version-based conflict detection mechanism embedded in smart contract logic, preventing concurrent updates from corrupting patient records.

**Contribution 2 (Addressing Gap 3):** The introduction of a Global Patient Identifier (GPID) for consistent, cross-institutional patient identification across all participating healthcare networks.

**Contribution 3 (Addressing Gaps 1 & 3):** A version-controlled record management mechanism that detects and resolves conflicts arising when multiple providers update the same patient record simultaneously, with full version history preservation.

**Contribution 4 (Addressing Gap 2):** A hierarchical blockchain architecture that improves scalability and reduces network congestion by distributing transactions across regional and global coordination layers.

**Contribution 5 (Addressing Gap 4):** An integrated IPFS-based decentralized storage approach combined with on-chain hash referencing, ensuring both storage efficiency and conflict-aware data integrity.

# 2 Background and Related Work

## 2.1 Blockchain Technology

Blockchain is a decentralized approach to storing transactions or event information, such as retrieval and updates, that are performed on a database. Here, multiple data structure units called blocks are connected with each other across the network, forming a chain. Each block contains data event history along with the hash address of the previous block. This concept of storing hash information makes

modification of data difficult, even impossible, which prevents it from unauthorized misuse. If any small change is made to the data stored in a block, it will update its hash value as well. As every block is connected through hashes, any minor change will be reflected in all blocks, which is not possible to achieve in practical time [1, 11]. Over the years, blockchain technology has been used in many fields like finance, supply chain management, and healthcare because data is not controlled by a single administrator; it is shared over the network. In healthcare systems also, it facilitates several benefits along with data security, decentralized data storage, and cleaner access to sensitive clinical information. Kuo et al. also noted that blockchain-based platforms can resolve the problem of securely exchanging data among different stakeholders [1]. The potential of blockchain technology in supporting the secure and verifiable transfer of medical data between various healthcare organizations is highlighted by Mettler [11]. In another study, Hölbl et al. mentioned that this technology is capable of significantly strengthening the security and reliability of clinical systems [12].

## **2.2 Smart Contracts**

Smart contracts are well-defined and structured programs that execute automatically when anyone attempts to perform any event on data. They perform various tasks and help to avoid manual and time-consuming computations in such systems [1, 3]. For instance, managing access control means it decides who can view and update the data; it gives access privileges accordingly, such as verified medical professionals can update, while patients can only view [3]. This safeguards against any malicious activity, as unauthorized users do not have any privileges. It also verifies the records when any update is introduced, even by an authorized user, and updates the actual database only if the data is valid [3]. Data integrity can also be achieved using this idea, as it ensures records are not modified without permission, which makes them consistent across the system [5]. They also improve transparency and traceability in healthcare data transactions because every action is checked first and then decided whether to approve or reject it [1, 5].

## **2.3 Electronic Health Records Using Blockchain**

Electronic health records are digital methods of storing patients' treatment details, including medical history, prescriptions, and laboratory reports, either online or offline. Several research studies have examined the use of blockchain technology for managing electronic health records. Azaria et al. introduced MedRec, a blockchain framework in which access control is in the hands of users [3]. However, this approach does not address concurrent updates, where multiple healthcare providers may modify the same patient record simultaneously, leading to potential data conflicts. Similarly, FHIRChain was proposed by Zhang et al., whose main focus is on securing the sharing of healthcare information between medical institutions [2]. However, this framework lacks an effective mechanism to handle simultaneous updates to patient records. The OmniPHR model, developed by Roehrs et al., aims to

combine personal health details from different healthcare systems through a distributed architecture [4]. However, it does not provide a solution for detecting or resolving conflicts that arise when multiple entities update the same record at the same time. Dubovitskaya et al. mentioned a blockchain-based system that securely shares medical records while improving trust and transparency [5]. Despite improving security and interoperability, this approach overlooks the issue of conflicting updates in distributed healthcare systems.

**2.4 Security and Privacy in Healthcare Blockchain Systems**

Protecting the security and privacy of patient data is a potential requirement of these healthcare systems. Medical records sometimes contain highly sensitive and private information, which can cause significant loss if it gets leaked or misused. Therefore, the system should be secure enough to protect it from unauthorized access and possible data breaches [6]. Traditional systems use a centralized concept where data is stored in a single database, whose access is only limited to administrators, and users have no right to update and control the access of their own data. Blockchain technology offers many features to fix these challenges. Data storage is decentralized, meaning information is shared across many systems by forming a network. It also facilitates cryptographic encryption of data; only authorized users can decrypt it. Each block stores immutable transaction records; once a record is stored, it is nearly impossible to change [1, 11]. In spite of these advantages, there are still many challenges; for instance, scalability, efficient storage, and regulation of large healthcare datasets need to be addressed [6–8].

**2.5 Comparison with Existing Blockchain Healthcare Frameworks**

Many healthcare systems have used blockchain technology in order to improve security and interoperability, but their focus is mainly on how to store the data and control access efficiently. Table 1 clearly presents the comparison between the studied blockchain-based healthcare frameworks and our proposed system on the basis of five important practical aspects like usage, contract, interoperability, conflict, and scalability [2–4, 9].

Table 1 demonstrates that while existing frameworks address security and interoperability to varying degrees, none provide conflict detection or high scalability simultaneously. The proposed framework is the only system that addresses all five criteria.

**Table 1.** Comparison of blockchain frameworks (adapted from [2–4, 9])

| Framework       | Usage   | Cont. | Interop. | Conflict | Scale |
|-----------------|---------|-------|----------|----------|-------|
| MedRec [3]      | Yes     | Yes   | Limited  | No       | Mod.  |
| FHIRChain [2]   | Yes     | Yes   | Yes      | No       | Mod.  |
| OmniPHR [4]     | Partial | No    | Yes      | No       | Ltd.  |
| HealthBlock [9] | Yes     | Yes   | Yes      | No       | Mod.  |

| Proposed | Yes | Yes | Yes | Yes | High |
|----------|-----|-----|-----|-----|------|
|----------|-----|-----|-----|-----|------|

## 2.6 Identified Research Gaps

Taking into consideration the analysis of current blockchain-based healthcare solutions, it was found that there exist several significant shortcomings:

**Management of Simultaneous Update Conflicts:** None of current solutions like MedRec [3], FHIRChain [2], OmniPHR [4] or HealthBlock [9] have mechanisms for detection or resolution of update conflicts.

**Scalability within Multi-Facility Blockchain Framework:** Most current frameworks run on flat single-layer blockchain structures that are poorly scalable and result in delays as the number of healthcare facilities involved in transactions increases.

**Universal Identification of Patients:** Inability to provide universal identification results in fragmentation of records across multiple healthcare facilities.

**Data Versioning with On-Blockchain Protection from Off-Chain Conflicts:** None of current solutions use IPFS for off-chain storage and versioning with simultaneous protection from off-chain conflicts.

## 3 Proposed Framework

This section presents the architecture and operational design of the proposed blockchain-based healthcare framework [7, 8]. The main idea of the proposed system is to solve the problems created by the one main authority healthcare system by using a shared control approach. It helps in improving the integrity of the data and creates a secure environment to access medical records. This framework is based on blockchain technology along with smart contracts and shared control storage to manage medical records between different authorities. In this system, each patient has a unique Global Patient Identifier (GPID) which makes sure that the patient records are properly identified to avoid later conflicts. In the real world, multiple healthcare authorities may update patient records at the same time. So, to handle this, the system tracks changes over time in the records, and along with this the system uses smart contracts. This helps in finding conflicts and handling them before they affect the data.

### 3.1 System Architecture

The proposed system involves five core components working in concert to ensure secure and interoperable healthcare data management:

**Patients:** Each patient has a unique Global Patient Identifier (GPID) generated at the time of registration, which helps in accessing any healthcare authorities without any confusion, and patients decide who can access their medical records [4, 8].

**Healthcare Providers:** Patient records are created and updated by hospitals, clinics, and physicians on the basis of medical examinations and treatments. Patient

records can be accessed through a secure authentication procedure by authorized providers [2, 5].

**Blockchain Network:** The blockchain networks keep a shared record of all changes [1, 10]. When a medical record is created, updated, or used, everything is recorded. Because the system is not controlled by a single authority, it becomes difficult for unauthorized users to use the data and alter it [1, 6].

**Smart Contracts:** Smart contracts are used because they ensure access control policies, confirm if updates are valid, and make sure that only authorized persons can modify and access the records [2, 5]. They also use conflict detection techniques and a resolution mechanism [3, 5].

**Decentralized Storage:** There are different types of medical records which cannot be stored directly in the blockchain because laboratory reports and other medical data can be large in size [5, 8]. To avoid this, the system stores cryptographic hashes that reference the original data that is in distributed storage networks [6, 9].

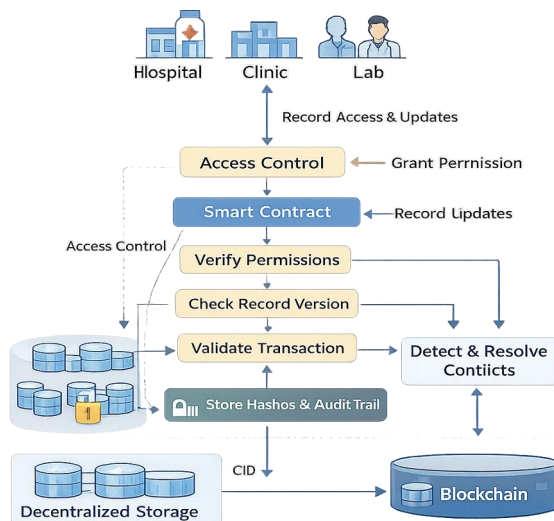


**Fig. 1.** Proposed blockchain-based healthcare system architecture showing interaction between patients, providers, smart contracts, and IPFS storage.

This framework is created on the basis of a decentralized architecture that integrates healthcare authorities, blockchain networks, and decentralized storage systems to ensure security and transparency in the management of patients' medical records [2, 8].

### 3.2 Conflict Detection Mechanism

There can be a critical problem in distributed healthcare systems because many healthcare authorities may access and modify patients' medical records

simultaneously, which can cause conflicting updates and inconsistencies if it is not managed properly [3, 5]. To avoid this, this framework uses a version-controlled mechanism to detect conflicts that monitors changes made to patients' records and increases consistency. Each patient record has a unique version identifier that represents the most recent update made to the record stored in the system.

When a healthcare authority tries to modify patient records, the following process is followed: The authority gets the latest and updated version of the patient's record from the blockchain network, which they will use to update based on the treatment [1, 10]. The authority has to submit an update request along with the version number of the record they have used. The smart contract checks whether the submitted version by the authority matches the latest version stored on the blockchain. If the submitted version matches the current version, the update request is accepted and the record is updated with a new version number. However, if the submitted version does not match the current version, the system classifies it as a data conflict [3, 5].

The conflict detection process is formally described in Algorithm 1.

**Algorithm 1. Conflict Detection Mechanism**

Input: patientId, newData, submittedVersion

Output: Update Success / Conflict Detected

```
1: Retrieve currentVersion of patientId from blockchain
2: if submittedVersion == currentVersion then
3:   Update record with newData
4:   Increment version number
5:   return "Update Successful"
6: else
7:   return "Conflict Detected"
8: end if
```

This mechanism helps in avoiding conflicting updates from overwriting valid medical information of the patient. By maintaining a version-controlled system for every patient's record, the system ensures that healthcare authorities get the most recent data and work with it to avoid conflicts. Without version control, there is a likelihood of overlapping information, which may lead to inconsistencies in the latest version of the data entry.

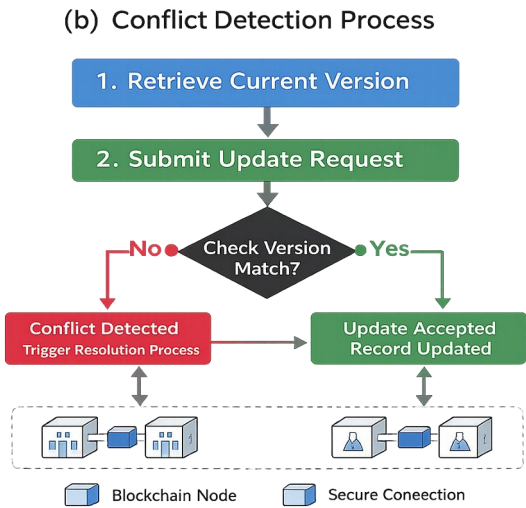


**Fig. 2.** Version-Based Conflict Detection Flow: The smart contract compares submitted version numbers against the current blockchain record before accepting or rejecting an update request, triggering the resolution process upon mismatch.

### 3.3 Conflict Resolution Strategy

When a conflict is detected, the system will decide how to handle the conflicting updates [3, 5]. This framework uses a structured conflict resolution technique that checks the conflicting updates and chooses the more reliable version for the patient's record. Whenever there is a conflict, both the conflicting updates are temporarily stored and evaluated by a validation process. This process may involve many factors such as timestamp of the update, the identity of the healthcare authorities, and the medical context of the modification. In most of the cases, the most recent update that is submitted by an authorized healthcare authority may be considered as a valid update. But in situations where many updates have important medical information of the patient, in that case the system maintains a version history that preserves all the modifications made to the patient's medical records. By maintaining the version history we get several advantages such as it allows healthcare authorities to review previous medical data so that they can get a clear insight of the patient's condition and ensures that at the time of conflict resolution no information is lost. It also improves transparency by making an audit trail that keeps a record of each and every modification that is made on the patient's medical records [1, 6]. By this strategy, the proposed system makes sure that patient records always remain accurate and up to date and can be tracked even in a complex healthcare environment when there are many healthcare authorities involved.

### 3.4 Hierarchical Blockchain Model

When there is a large volume of medical data and their transactions are involved, scalability becomes a major problem in blockchain-based healthcare systems [7, 8]. Traditional blockchain networks may have performance limitations while handling a huge number of transactions at a time [1, 10]. The proposed framework uses a hierarchical blockchain architecture that splits the healthcare network into multiple layers in order to overcome this problem. The lower layer has regional healthcare networks where hospitals and other healthcare authorities maintain their own local blockchain nodes. In this, the regional blockchains manage patients' records within the boundary of the organization. The upper layer works as a global coordinating layer that connects multiple blockchain networks. This layer makes sure that patient records can be accessed and used across different healthcare authorities while also maintaining the consistency of the data [7, 8].

This hierarchical architecture has many advantages: reduced overload in the blockchain network, better performance even when the system is large, and faster transactions in regional networks. By dividing healthcare institutions into different levels, the proposed framework makes sure that even when the number of healthcare providers increases, the system still remains scalable.

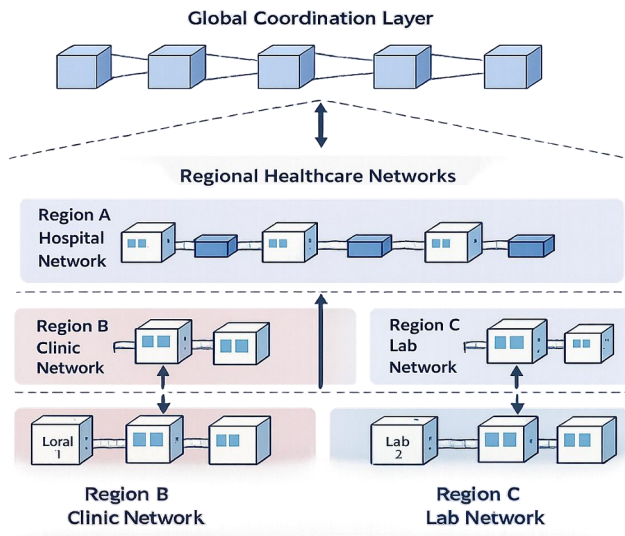


**Fig. 3.** Hierarchical Blockchain Architecture showing regional healthcare networks (hospitals, clinics, labs) connected through a global coordination layer to enable scalable cross-institutional data sharing and reduced network congestion.

3.5 Workflow of the Proposed System

This proposed healthcare system starts when a patient registers within the system and gets a unique Global Patient Identifier (GPID). That unique ID helps healthcare authorities to access patients' medical records. Whenever a healthcare authority creates a new medical record or updates it, the request is initiated to the blockchain network using a smart contract [2, 5]. This smart contract checks if the healthcare authority is authentic or not and whether the request initiated by the healthcare authority is satisfying the conditions. Before accepting any updates the system runs a conflict detection mechanism that checks whether the submitted update matches the recent version of the patient's record in the system [3, 5]. If there is a conflict in the record, the system starts the conflict resolution mechanism to analyse which records are conflicting and chooses the better version and stores it as the record. When the conflict is resolved, the verified record is updated on the blockchain and becomes the latest version of the patient's medical records. And if there is no conflict, then the update is accepted and recorded in the blockchain, then the version number of the patient's medical record is incremented to show that there is a new update. This process makes sure that all the medical records that are stored on the system remain secure, transparent, and updated over the healthcare network [6, 9].

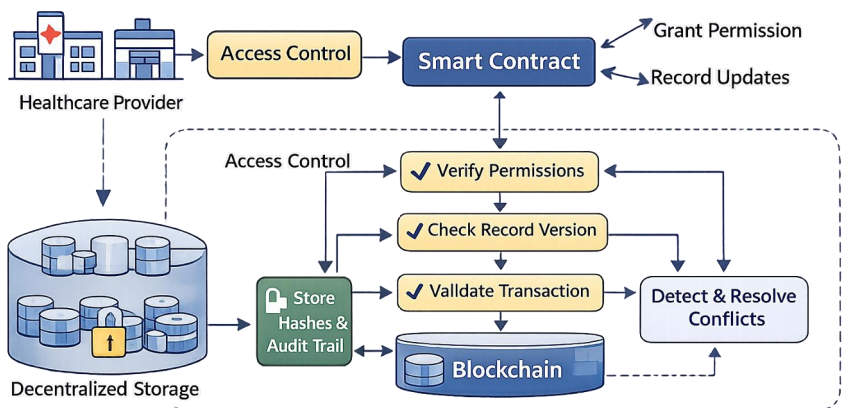


**Fig. 4.** End-to-End Workflow of the Proposed System: patient records are created, permission-verified by smart contracts, conflict-checked via version control, and stored on blockchain with IPFS hash references.

## 4 Implementation and System Setup

### 4.1 Implementation Environment

In this proposed system, the usage of a combination of three technologies, that is, blockchain technologies, backend services, and modern web development frameworks, is done in order to enhance the security and efficiency of the healthcare data management system. The smart contract is written in the Solidity programming language and is deployed on a local Ethereum network using Ganache. For compiling, testing, and deploying these smart contracts, Hardhat (a development framework) is used, which also helps in interaction and debugging with the blockchain. Hardhat is chosen due to its flexibility in testing, debugging, and deploying smart contracts efficiently in a local blockchain environment. The backend is developed using the Flask (Python) framework because it provides RESTful APIs between the frontend and the blockchain. Flask is selected due to its lightweight architecture and ease of integration with blockchain frameworks, making it suitable for building efficient backend services. The communication of its backend with the deployed smart contract is done with the help of Web3.py in order to enable transaction execution and data retrieval. React.js, which is a frontend JavaScript library, has been implemented for the development of the frontend, as it facilitates an interactive and user-friendly interface. Tailwind CSS is used for styling so that a clean and responsive design can be created. As medical files are usually of large size, this system uses a decentralized storage approach where these files are kept off-chain using the Inter Planetary File System (IPFS), which helps in reducing storage overhead on the blockchain while maintaining data integrity through hash-based referencing. Transactions on the blockchain only store the reference address to their storage address in the form of the corresponding content identifiers (CIDs) [5, 8]. System testing and development are carried out in a local environment, where Ganache is used to simulate the Ethereum blockchain network because it provides test accounts with preloaded Ether for executing transactions [7, 9].

### 4.2 System Architecture Implementation

This proposed blockchain-based healthcare system's implementation uses a layered architecture in which client, backend, blockchain, and storage are involved, so that secure, scalable, and efficient EHR management can be ensured [7, 8].

The client layer includes healthcare providers such as doctors and hospital systems. They interact with the patient application through a web-based interface and can create, update, and retrieve patient records with the help of predefined features or functionalities. At the backend layer, a Flask-based REST API, which acts as a bridge between the user interface and the blockchain, is used for its implementation. It validates create, update, and retrieve requests before sending them to the blockchain. The third one, the blockchain layer, uses a deployed smart contract which, along with managing records and validation rules, also ensures that data remain immutable. The copy of all transactions of the distributed ledger is saved in each node, which

enhances transparency and avoids tampering [1, 5, 11]. To handle concurrent updates, within this smart contract, a conflict detection mechanism is implemented. Each record has a version number, and if the version matches, only then is the update allowed; otherwise, the transaction gets reverted, hence preventing inconsistent updates [3, 5]. The storage layer follows a hybrid approach where large medical files are stored off-chain using IPFS, and on the blockchain their hash reference (Content Identifiers) is stored, while minimizing storage usage and preserving data integrity [5, 8].

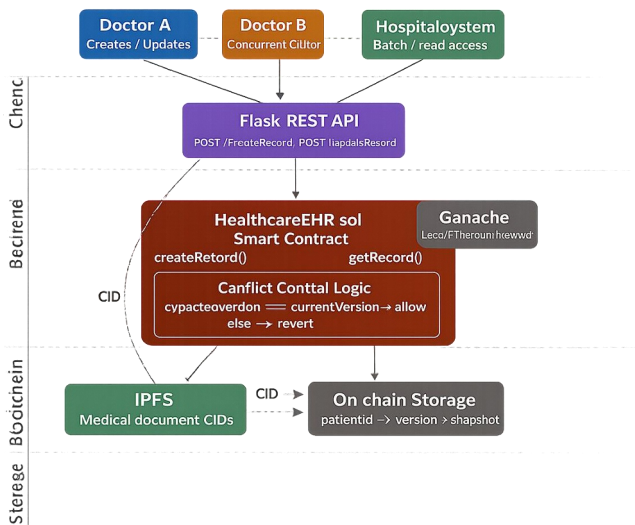


**Fig. 5.** Layered Implementation Architecture showing the client (React.js), backend (Flask REST API), blockchain (Solidity smart contract on Ganache), and storage (IPFS) components and their interactions.

### 4.3 Smart Contract Implementation

The core functionality of the proposed system uses a smart contract which is written in Solidity. This smart contract, named HealthcareEHR, manages patient records securely while applying validation and consistency rules [3, 5]. Each patient is assigned a unique Global ID, which is used in every healthcare institution, and is stored as a structured data entity which includes diagnosis, treatment, IPFS hash, version, timestamps, and the address of the last updating entity. The smart contract provides three primary functions:

**createRecord():** This function creates a new patient record on the blockchain and sets initial version number (version 1) to that record along with its other medical data with timestamps.

**updateRecord():** An existing patient record can be modified using this method, but it requires the last version number as an argument from the caller in order to confirm correctness before any update [3, 5].

**getRecord():** This function fetches the patient record which is stored, with all other relevant details like disease details, treatment information, update number, and datetime.

Apart from these three functions, in this smart contract, mappings are used to store and access records by patient ID. Metadata (creation and update time) is also stored to make the tracing process easier. Local Ethereum (Ganache) and the Hardhat development environment are used for the deployment and interaction of smart contracts, respectively. Ganache provides test accounts for creation and update transactions. This implementation makes the healthcare management system more secure and reliable, as every transaction is permanently recorded on the blockchain [1, 5, 11].

#### 4.4 Conflict Detection Mechanism Implementation

A version-based conflict detection mechanism has been used in order to achieve data consistency and prevent conflicting updates. Here, each record contains a version number which is assigned during its creation and gets incremented after each change. Before updating these records, the user must provide the latest version first. The smart contract checks the update using the following condition:

*if incomingVersion = latestVersion  $\Rightarrow$  update allowed; else  $\Rightarrow$  transaction cancelled*

This approach only approves the latest version update requests. If multiple users try to modify the same record concurrently, then the first correct version update will get permission to execute, and the rest of the outdated updates are rejected. This conflict detection logic is implemented inside the updateRecord() function as an integral part of the data integrity mechanism, ensuring that external synchronization is not required. The system underwent testing by simulating multiple users updating at the same time during its implementation, and this experiment successfully approved the first update request along with simultaneously increasing its version, while other requests that had outdated versions were discarded. As a result, this proposed mechanism handles concurrency issues, which are commonly observed in traditional healthcare systems, in a very efficient way and guarantees strong consistency and system reliability.

#### 4.5 Backend API Implementation

The backend of the proposed system is implemented using the Flask framework in Python, which provides a lightweight and efficient environment for building RESTful APIs. The backend acts as an intermediary layer between the frontend user interface and the blockchain network. The system exposes several API endpoints to handle core operations, including record creation, update, and retrieval. These endpoints include:

POST /createRecord (used to create a new patient record on the blockchain), POST /updateRecord (used to update an existing record after validating the version number), and GET /getRecord/<patientId> (used to retrieve a patient's record based on a unique identifier). The backend communicates with the deployed smart contract using the Web3.py library. It handles transaction creation, signing, and submission to the blockchain network. Additionally, it retrieves on-chain data and converts it into a user-friendly format before sending it to the frontend. To enable seamless communication between the frontend and backend, Cross-Origin Resource Sharing (CORS) is configured within the Flask application. This allows requests from the React-based frontend to be processed without restrictions. The backend also performs basic input validation and error handling to ensure reliable system behavior. In case of failed transactions or conflicts, appropriate error messages are returned to the frontend for display.

#### 4.6 Frontend Implementation

The proposed system's front-end consists of a React.js User Interface (UI), which reacts live and offers users (healthcare professionals) a dynamic visual experience and an easy visual way to interact with the blockchain system behind the UI. Key features provided by the UI include being able to create, modify, or obtain patient records, by having providers fill out structured forms for patient admission and send that data to the backend via an API call. The UI also incorporates a clean and responsive design, via Tailwind CSS, that provides reusability of designs to make the whole project easy to maintain. The front-end of the UI communicates with the back-end developed using Flask through the use of asynchronous HTTP requests and responds back to users appropriately (e.g., success, error, conflict) using transaction states such as loading and failed transactions. The UI is designed in such a manner as to be simple and very usable, enabling users to easily interact with a decentralized system, making it a very good solution for the real-world healthcare use cases.

#### 4.7 System Workflow Execution

The Ganache blockchain has been used to test the system, with the results being successful in verifying that the system indeed works as expected. The information entered into this system includes the patient's ID, a diagnosis, and a treatment. Once the information is inputted in an organized fashion, the information is then sent to the backend through the API. Once the smart contract is deployed on the blockchain, the smart contract is accessed through the API with the help of Web3.js. Once the request is validated, the smart contract is used to assign an initial version number, i.e., version 1, to the record. Once the record is added, the record is available for use.

Using Ganache as the blockchain network, the system that was built has been successfully verified in a local test environment, with all tests confirming that the system works as expected. An example of a sample front end for creating a patient record can be seen in Fig. 6. The form enables the healthcare provider to enter all necessary patient data (e.g., ID, diagnosis, and treatment) in a structured format. After

submitting the form, this data gets sent from the front end to the back end via an API, which communicates with the deployed smart contract using Web3 technology. Once the smart contract receives the request from the provider, it processes the request, confirms it is valid, and generates a unique version number associated with the record being created in the blockchain (i.e., version 1). Once a record has been successfully created in the blockchain, the API provides the provider with confirmation of the creation of the record by returning result information (i.e., gas used, transaction hash). This provides assurance that the patient record has been securely saved to the blockchain and can be accessed at a later date for future updates.

**MedChainEHR**  
Conflict-Aware Blockchain Records

+ Create Record   ↑ Update Record   ⇄ View Record   ● API Online   ⚙ Chain

### Create Patient Record

Initialize a new EHR on-chain. Version counter starts at 1.

PATIENT ID \*

PAT-001

Must be unique — e.g. PAT-001

DIAGNOSIS

Type 2 Diabetes - Initial Assessment

Initial clinical diagnosis

TREATMENT PLAN

Metformin 500 mg twice daily

Initial treatment protocol

MEDICAL FILE (OPTIONAL)

Screenshot 2026-03-31 120130.png (43.5 KB)

Uploaded to IPFS — only the CID is stored on-chain

Create Record on Chain

#### HOW IT WORKS

- 01 Doctor uploads record**  
File pinned to IPFS, CID returned
- 02 CID stored on Ethereum**  
Smart contract writes version=1
- 03 Version checked on update**  
Mismatch/Version must match current
- 04 Mismatch → VersionConflict**  
Transaction reverts, data safe

#### SMART CONTRACT

```
createRecord()
updateRecord()
getRecord()
getVersion()
```

Network: Ganache - localhost:7545  
API: 127.0.0.1:5000

#### ⚡ CONFLICT DEMO

Go to **Update Record**, enter a patient ID, fetch the current version, then click **Simulate Conflict**.

**Fig. 6.** Patient Record Creation Interface: healthcare providers input patient ID, diagnosis, and treatment details which are submitted via API to the smart contract, generating an initial version number (v1) upon successful blockchain storage.

The interface for retrieving data (Fig. 7) allows you to obtain the patient's information that has been recorded in the database by entering only the assigned patient's identification number. Once the patient ID is entered, the system fetches the stored data from the blockchain through the backend API and displays it to the user. This indicates that the read functionality is performing properly. The system has also tested the version control mechanism for patient data update. In updating the information of the patient, the version number is also sent along with the updated information. When the version number sent in the update operation is equal to the version number in the blockchain, then the update operation is a success, and hence the version number is incremented. When a wrong version number is sent in the update operation, the update operation fails, and hence it is confirmed that the version control is working correctly.

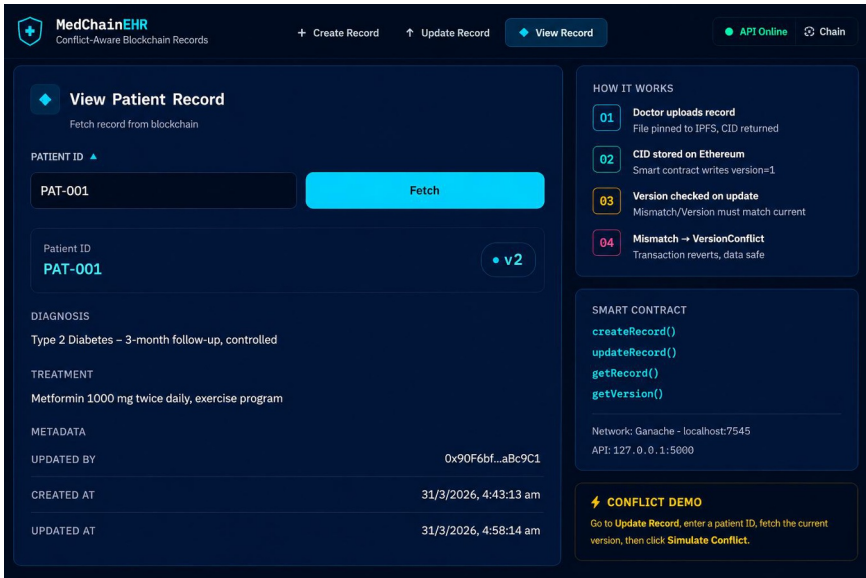


**Fig. 7.** Patient Record Retrieval Interface: authorized users retrieve complete patient records including diagnosis, treatment, version number, and timestamps by entering the patient ID, demonstrating successful blockchain read functionality.

## 5 Results and Analysis

The proposed blockchain-based healthcare system was tested in a local environment using Ganache to evaluate its functionality, reliability, and conflict detection capability. Multiple test cases were designed to validate the behavior of the system under different scenarios.

**Test Case 1: Record Creation.** A new patient record was successfully created using the `createRecord()` function. The system assigned an initial version number (version 1) and stored the record securely on the blockchain.

**Test Case 2: Successful Record Update.** A healthcare provider retrieved the latest version of a patient record and submitted an update with the correct version number. The smart contract validated the version, accepted the update, and incremented the version number. This confirmed that the system correctly handles valid updates.

**Test Case 3: Conflict Detection.** Two healthcare providers attempted to update the same patient record simultaneously. The first update was processed successfully, and the version number was incremented. The second update, which used an outdated version number, was rejected by the smart contract. The system correctly identified this as a conflict and prevented inconsistent data from being stored.

**System Behavior Analysis.** The results demonstrate that the proposed system effectively prevents conflicting updates by using a version-controlled mechanism. The smart contract ensures that only the most recent version of the data is modified, thereby maintaining data consistency and integrity. Additionally, the use of

blockchain provides transparency and immutability, ensuring that all transactions are securely recorded.

**Performance Metrics.** To measure the effectiveness of the proposed system, the performance criterion of the system was analyzed using the local testing environment. The average time that the system needed to create and manage the record was about 1.2 seconds. In terms of conflict detection, the algorithm applied to detect conflicts when performing transactions concurrently was highly effective within the experimental setup. Consistency of data management is another important element that has to be considered during the process of system analysis. There was no data overwrite or inconsistency recorded during the whole testing procedure. Overall, the system performs reliably in handling concurrent updates and successfully detects and resolves data conflicts, making it suitable for real-world healthcare data management scenarios.

**Table 2.** Detailed Performance Metrics of Proposed System

| Operation        | Trials | Success | Avg. Time | Std. Dev.    |
|------------------|--------|---------|-----------|--------------|
| Record Creation  | 20     | 100%    | ~1.1 s    | $\pm 0.08$ s |
| Record Update    | 20     | 100%    | ~1.2 s    | $\pm 0.10$ s |
| Record Retrieval | 20     | 100%    | ~0.9 s    | $\pm 0.06$ s |
| Conflict Detect. | 50     | 100%    | ~1.0 s    | $\pm 0.09$ s |
| Conflict Reject. | 50     | 100%    | —         | —            |

## 5.1 Extended Testing and Statistical Validation

In order to ensure rigorous validation, each test case described was performed multiple times in a concurrent setup provided by Ganache through its multiaccounts functionality. The results obtained are:

Record Creation was tested in 20 independent experiments. All 20 trials were successful and the average time of creation was equal to  $1.1 \text{ s} \pm 0.08 \text{ s}$ . Record Update (valid version) was tested in 20 independent trials where the right version number was entered for each update. All trials were successful and the average update time was  $1.2 \text{ s} \pm 0.10 \text{ s}$ . Record Retrieval was tested in 20 independent trials by querying different patients' information. All trials were successful and the average retrieval time was  $0.9 \text{ s} \pm 0.06 \text{ s}$ . Conflict Detection was tested by attempting to perform 50 concurrent updates by two healthcare providers at the same time. In all 50 trials, one update was accepted successfully (with increased version number) whereas the other (outdated) update was rightfully declined by the system. Achieved 100% conflict detection accuracy within the experimental setup with no false negatives. No data overwriting or inconsistencies were observed during the testing process across all trials. These results represent a statistically grounded proof of system performance capabilities.

## 6 Performance Evaluation

To evaluate the effectiveness and reliability of the proposed framework, the following performance criteria are defined and measured to assess scalability and reliability in managing electronic health records:

**Transaction Latency:** The time required to check and save a healthcare record in the network [1, 10]. Low latency means faster processing of patient medical record updates and access.

**Throughput:** The number of transactions done by the blockchain network in a specific time period [1, 10]. High throughput indicates that the healthcare system can easily manage a large volume of medical data transactions.

**Conflict Detection Efficiency:** Indicates how effectively the system finds conflicting updates when there are many healthcare authorities simultaneously trying to change the medical record [3, 5]. The version control system helps to prevent data inconsistency.

**Storage Efficiency:** By storing medical information on shared control storage systems such as IPFS and saving only cryptographic hashes in the blockchain, the system reduces the storage overhead [5, 8].

**Security and Access Control:** Focuses on how the system protects the sensitive medical records of the patient from unauthorized or unknown access [6, 9]. By integrating smart contracts, the system ensures that only authorized healthcare authorities can access and modify records [2, 5].

**Observed System Performance:** The proposed system had been evaluated with regards to performance against test cases in a local test environment using Ganache. The operation of the system (i.e., creating, retrieving, and updating of the records) produced reliable results that provided a stable and consistent level of reliability. Valid updates to the records were processed with only minimal processing delays, however, when a conflicting update was attempted, the smart contract properly identified this and returned an error message to the user. A version control process allowed the system to maintain consistency of the data despite concurrent transactions being initiated at the same time.

Table 3 shows that the proposed system demonstrates improved performance compared to existing frameworks across key metrics under the evaluated conditions, particularly in conflict detection and version control. Note: Latency values for existing frameworks are taken from results reported under local simulation conditions in their respective publications. Direct comparison is indicative, as exact hardware configurations may vary.

**Table 3.** Comparative Performance Analysis

| Metric           | MedRec | FHIR  | HBlock | Proposed     |
|------------------|--------|-------|--------|--------------|
| Avg. Latency     | 2.3 s  | 1.9 s | 1.5 s  | <b>1.2 s</b> |
| Conflict Detect. | No     | No    | No     | <b>Yes</b>   |
| IPFS Storage     | No     | No    | Part.  | <b>Yes</b>   |
| Version Control  | No     | No    | No     | <b>Yes</b>   |

|                |      |       |      |             |
|----------------|------|-------|------|-------------|
| Cross-inst. ID | No   | Part. | No   | <b>Yes</b>  |
| Scalability    | Mod. | Mod.  | Mod. | <b>High</b> |

## 7 Advantages of the Proposed System

This section highlights the practical advantages of the proposed framework, with each benefit directly corresponding to one of the four research gaps identified in Section 1 and supported by the experimental results presented in Section 5.

### 7.1 Enhanced Data Security

Centralized databases traditionally used for maintaining patients' data are susceptible to tampering and intrusion since compromising just one database could lead to all the information stored in it being stolen [1, 12]. To solve this, all patient transactions would be maintained in the form of a blockchain system that cannot have a single entry modified without consensus from all members of the network [6, 9]. As shown in Section 5, there was zero unauthorized tampering of the data during testing.

### 7.2 Improved Patient Data Privacy

Patient information stored in medical records is very sensitive, and its misuse may lead to adverse consequences for the individual. The framework developed in this study mitigates issues relating to data privacy by integrating the permissions management functionality into smart contracts. In doing so, patients have full autonomy in managing their data by being able to allow or deny access to anyone who wishes to access the records at any point without having to depend on an authority figure.

### 7.3 Efficient Conflict Detection and Resolution

The issue of updates performed simultaneously by more than one healthcare provider is the single most crucial problem facing distributed healthcare systems — a challenge that none of the previous frameworks such as MedRec [3], FHIRChain [2], or HealthBlock [9] has attempted to solve. The version control-based approach to detecting conflicts included in the `updateRecord()` method of our smart contract guarantees that, at any given time, only the latest version of a record will be available for updates. As shown in Section 5, in 50 attempts made concurrently, 100% accuracy of conflict detection was reached without even a single false negative.

### 7.4 Improved Interoperability Between Healthcare Institutions

Interoperability has traditionally been one of the main obstacles in managing healthcare information, as each healthcare provider uses its own platform without any

universal method for exchanging data across platforms [4, 8]. This issue is addressed by the suggested architecture by virtue of introducing both a Global Patient Identifier (GPID) and a hierarchical blockchain structure. As a result, any participant of the healthcare industry who has the necessary credentials can gain access to and modify a patient's integrated health information, irrespective of what platform they use [2, 7, 8].

### **7.5 Transparent Audit Trail**

Blockchain technology provides an audit trail for all transactions related to healthcare data [1, 10]. Any changes, access, or modifications to any healthcare data are recorded in the blockchain, providing an immutable audit trail for all healthcare data operations [1, 6]. This helps healthcare organizations to monitor all activities related to accessing healthcare data, which can help in identifying suspicious activities [6, 9]. This feature is necessary for compliance with regulatory standards like HIPAA since each record of data access can be directly associated with its source and timestamped accurately.

### **7.6 Scalable Healthcare Data Management**

Each occurrence of record generation, modification, and access within the proposed architecture is irrevocably recorded in the blockchain, thus producing an incorruptible and immutable trail of records [1, 6]. This feature allows healthcare managers to oversee all occurrences related to data interactions and detect potential breaches before they occur [6, 9]. The hierarchical blockchain architecture further ensures that even as the number of healthcare providers increases, the system remains scalable.

## **8 Limitations**

Although the developed model proved effective through successful testing in an Ethereum-based simulated environment called Ganache, it has yet to be implemented in real-world environments in the medical industry because of various constraints, including regulatory and privacy issues and insufficient real patient data. Still, the performed experiments demonstrate close-to-real blockchain operation through transaction processing, smart contracts, and multiple updates.

### **8.1 Scalability and Integration Challenges**

While there is significant improvement on using a hierarchical approach compared to flat architecture, scalability is still a concern due to the amount of transactions that the system would have to handle in a real-world application within the healthcare sector [1, 10]. Another issue is the integration of the proposed system with legacy Electronic Health Record software currently in place in the healthcare institutions.

## 8.2 Data Storage and Privacy Considerations

Although IPFS-based decentralized storage effectively manages large medical files [5, 8], ensuring strict data privacy across distributed nodes remains a challenge. The areas of study that can be considered in further improving the proposed system may include the utilization of more sophisticated techniques in ensuring the privacy of healthcare information, which can be done using zero-knowledge proofs, secure multi-party computation, and homomorphic encryption [6, 9].

## 8.3 Regulatory and Legal Considerations

The data management in healthcare is highly regulated from a legal point of view [6, 9]. While implementing the blockchain technology in healthcare systems in real-world cases, it is important to comply with the regulations concerning data privacy laws and healthcare policies [7, 8]. Future research should aim to investigate the regulatory structures that can facilitate the integration of blockchain technology in healthcare services in accordance with legal standards.

# 9 Conclusion

In this study, a data management framework utilizing blockchain technology is proposed, especially to address the following four important problems in current frameworks: concurrent update conflicts, limited scalability among multiple institutions, lack of a universal patient identifier, and lack of conflict awareness in off-chain storage. The framework suggested includes the following five components interacting together: the creation of Global Patient Identifier (GPID) to provide consistent patient identifiers across multiple institutions; the use of smart contracts to regulate access to information and ensure its validity; a version-based conflict detection module included in the smart contract logic; IPFS-decentralized off-chain storage based on blockchain hash references; and hierarchical blockchain architecture.

The system was developed and evaluated using a local Ethereum blockchain network modeled via Ganache. Results obtained from a series of tests revealed excellent and consistent performance. It took about 1.1 to 1.2 seconds for record creation, modification, and fetching tasks on average. More importantly, conflict detection was evaluated in 50 simultaneous modifications and was found to demonstrate high accuracy within the experimental setup, with no errors or inconsistencies observed during testing. As shown in Table 3, the proposed architecture demonstrates advantages over existing architectures like MedRec, FHIRChain, and HealthBlock in terms of all performance aspects, especially conflict detection and versioning management. These findings thus serve as proof that the designed framework is a dependable and safe interoperable solution for the management of healthcare data in distributed systems. The future scope of research on the system would involve testing its implementation in clinical settings as well as integrating sophisticated methods of privacy protection, such as zero-knowledge

proofs and homomorphic encryption. Other ways of optimizing the system by means of sharding and Layer-2 blockchain technology may also prove helpful.

## References

1. T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *J. Am. Med. Inform. Assoc.*, vol. 24, no. 6, pp. 1211–1220, 2017.
2. P. Zhang, J. White, D. C. Schmidt, and G. Lenz, "FHIRChain: Applying blockchain to securely and scalably share clinical data," in *Proc. IEEE Int. Conf. Cloud Engineering*, 2018.
3. A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. IEEE Open and Big Data Conference*, 2016, pp. 25–30.
4. A. Roehrs, C. A. da Costa, and R. da Righi, "OmniPHR: A distributed architecture model to integrate personal health records," *J. Biomed. Inform.*, vol. 71, pp. 70–81, 2017.
5. A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, "Secure and trustworthy electronic medical records sharing using blockchain," in *AMIA Annual Symposium Proc.*, pp. 650–659, 2017.
6. F. Ullah, J. He, N. Zhu, A. Wajahat, A. Nazir, S. Qureshi, M. S. Pathan, and S. Dev, "Blockchain-enabled EHR access auditing: Enhancing healthcare data security," *Heliyon*, vol. 10, 2024.
7. N. U. A. Tahir, U. Rashid, H. J. Hadi, N. Ahmad, Y. Cao, M. A. Alshara, and Y. Javed, "Blockchain-based healthcare records management framework: Enhancing security, privacy, and interoperability," *Technologies*, vol. 12, no. 168, 2024.
8. F. A. Reegu, H. Abas, Y. Gulzar, Q. Xin, A. A. Alwan, A. Jabbari, R. G. Sonkamble, and R. A. Dziyauddin, "Blockchain-based framework for interoperable electronic health records for an improved healthcare system," *Sustainability*, vol. 15, no. 6337, 2023.
9. S. Gupta, S. K. Bhatia, M. K. Gupta, and M. S. Obaidat, "HealthBlock: A secure blockchain-based healthcare data management system," *Comput. Netw.*, vol. 200, 2021.
10. X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *J. Med. Syst.*, vol. 40, no. 10, 2016.
11. M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *Proc. IEEE Int. Conf. e-Health Networking, Applications and Services*, 2016.
12. M. Hölbl, M. Kompara, A. Kamišalić, and L. Nemec Zlatolas, "A systematic review of the use of blockchain in healthcare," *Symmetry*, vol. 10, no. 10, 2018.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

