

PEER-REVIEWED RESEARCH

OPEN ACCESS

ISSN Online: 2516-3957

ISSN Print: 2516-3949

[https://doi.org/10.31585/jbba-9-2-\(7\)2026](https://doi.org/10.31585/jbba-9-2-(7)2026)

MrBlock+: A Blockchain-Based Framework for Secure, Interoperable, Scalable, and Access-Controlled Electronic Health Records Using Interplanetary File System

Pranalini Joshi¹ and Pratibha Mahajan²¹Department of Computer Engineering, Vishwakarma University, Pune, India²Department of Artificial Intelligence, Vishwakarma University, Pune, India**Correspondence:** pranalini.joshi-317@vupune.ac.in, pratibha.mahajan@vupune.ac.in**Received:** 07 Feb 2026 **Accepted:** 30 May 2026 **Published:** 26 August 2026

Abstract

The management of Electronic Health Records (EHRs) in blockchain-based systems requires addressing key challenges such as scalability, secure and scalable management of large medical files, and fine-grained access control. This paper presents MrBlock+, a blockchain-enabled framework that integrates the Interplanetary File System (IPFS) for decentralised off-chain storage, ensuring efficient handling of large EHR documents while preserving data integrity and auditability. The system employs smart contracts to enforce a hybrid Role-Based and Attribute-Based Access Control (RBAC + ABAC) mechanism, using organisational identity as a critical attribute in access decisions. MrBlock+ supports access-controlled EHR sharing by exposing IPFS Content Identifiers (CIDs) only after successful blockchain-based authorisation. Experimental results demonstrate that MrBlock+ achieves substantial performance gains, including up to 45% faster upload times compared with selected benchmark systems. MrBlock+ also showed lower reported download latency than the selected benchmark system under the evaluated file-retrieval conditions, with an observed reduction of over 99.7% for a 100 MB file. These improvements validate the framework's suitability for real-time, secure, and scalable EHR management across distributed healthcare environments.

Keywords: *Blockchain, Distributed Technology, Hyperledger Fabric Network, Role-based and Attribute-based access control, IPFS, Interoperability, Electronic Health Records*

JEL Classifications: *I10, C88, L86, O33*

1. Introduction

Electronic Health Records (EHRs) are digital repositories of patients' medical histories, clinical reports, diagnostic results, and treatment information. They play a critical role in modern healthcare by enabling efficient data management, improving continuity of care, and supporting clinical decision-making. Today, most hospitals have adopted EHR systems to enhance patient care and operational workflows [1]. The global EHR market continues to grow rapidly and is projected to reach approximately 43 billion USD by 2032 [2].

Despite widespread adoption, effective sharing of EHR data across healthcare institutions remains a significant challenge. Hospitals and healthcare providers are often reluctant to exchange patient records due to concerns related to data privacy, lack of trust, regulatory compliance, and security risks. Such reluctance highlights the critical need for technologies that provide enhanced security, scalability, and privacy protection while enabling interoperability across healthcare systems. Traditional centralised EHR systems often struggle to meet these demands, particularly in multi-hospital

environments and when managing large volumes of sensitive medical data [3].

Blockchain technology has emerged as a promising solution to these challenges due to its decentralised, tamper-resistant, and transparent nature. Blockchain-based EHR systems can enhance data integrity, enable secure and auditable access control, and facilitate trustworthy data sharing without reliance on a single centralised authority. Although blockchain has been widely explored for EHR management over the last decade, its large-scale adoption in healthcare remains limited. Existing blockchain-based EHR systems have proposed solutions for secure data sharing, patient-centric access control, auditability, and decentralised storage. However, many of these approaches still face practical challenges such as scalability limitations, interoperability with existing hospital systems, high on-chain storage overhead, complex access control management, privacy concerns, and limited real-world deployment. These challenges indicate that blockchain alone is not sufficient unless it is combined with efficient off-chain storage, fine-grained access control, and performance-aware system design.

Our prior work introduced MrBlock [4], a blockchain-based framework using Hyperledger Fabric to enhance access control and interoperability of EHR systems. While MrBlock demonstrated improvements in security and interoperability, its reliance on on-chain storage introduced scalability limitations, particularly for storing large clinical files such as medical images and diagnostic reports.

To overcome these limitations, this work extends the MrBlock framework by integrating the Interplanetary File System (IPFS) as a decentralised off-chain storage mechanism. By offloading large data from the blockchain and storing it on IPFS, the system reduces blockchain storage overhead while maintaining data availability and security. IPFS enables content-addressable, peer-to-peer storage and is selected for its lightweight architecture and compatibility with permissioned blockchain environments.

Despite extensive research on blockchain- and IPFS-based EHR systems, several critical limitations remain unresolved. Most existing frameworks rely on patient-only access control, coarse-grained role-based policies, or heavy cryptographic mechanisms that increase computational overhead and reduce real-time usability. Moreover, organisational contexts such as hospital affiliations are rarely enforced as a first-class access attribute, limiting interoperability in multi-hospital healthcare environments. Additionally, many proposed systems lack detailed performance evaluation for large medical files, making their real-world applicability uncertain.

To address these issues, this paper proposes MrBlock+, a blockchain-based EHR framework that integrates Hyperledger Fabric with IPFS-based off-chain storage and smart contract-driven access control. The evaluation in this study is aligned with the key problem areas identified in existing literature, including secure access enforcement, efficient upload and retrieval of medical records, reduced storage burden on blockchain, and improved system performance in terms of latency and throughput. MrBlock+ enforces a hybrid Role-Based and Attribute-Based Access Control (RBAC-ABAC) model, where organisational identity is treated as a key access attribute. Access policies are dynamically enforced through smart contracts deployed on a permissioned Hyperledger Fabric network.

This framework follows a systematic design-and-evaluation methodology, encompassing architectural design, smart contract implementation, functional access control validation, and performance benchmarking using real-world medical file sizes. The key contributions of MrBlock+ are as follows:

- MrBlock+ enhances the original MrBlock framework by integrating IPFS for decentralised off-chain storage, thereby improving scalability and significantly reducing on-chain data load for large EHR files.
- Smart contracts are implemented to enable secure uploading of EHRs to IPFS and controlled retrieval based on blockchain-verified credentials.
- The MrBlock+ framework presents a novel hybrid RBAC-ABAC access control mechanism that leverages hospital identity as a key attribute.

- It supports access-controlled EHR sharing by restricting exposure of IPFS Content Identifiers (CIDs) to blockchain-authorised users, while recognising file-level encryption as a necessary enhancement for stronger confidentiality in real-world deployments.
- The framework exhibits improved scalability and achieves significantly lower upload and download times compared to existing blockchain-based healthcare systems.
- A comprehensive security analysis is performed using the CIA+AA model and STRIDE-based threat modelling to assess confidentiality, integrity, availability, authentication, and authorisation guarantees.

The paper is organised as follows:

Section 2 reviews related work and off-chain storage architecture. Section 3 presents the design and implementation of MrBlock+. Section 4 evaluates system functionality and performance. Section 5 provides a security analysis of the proposed framework, and Section 6 concludes the paper and outlines future research directions.

2. Related Work

Efficient and secure storage of Electronic Health Records (EHRs) in blockchain-based systems remains a critical research challenge, particularly when large medical files and fine-grained access control requirements are involved. While blockchain offers immutability and auditability, its limited storage capacity and transaction throughput constrain direct on-chain data storage in healthcare applications. This section first presents an overview of off-chain storage architectures suitable for blockchain integration and then reviews recent blockchain-IPFS-based EHR frameworks, highlighting their strengths and unresolved limitations in comparison to the proposed MrBlock+ framework.

2.1 Overview of Off-Chain Storage in Healthcare Blockchain Systems

Storing Electronic Health Records (EHRs), such as laboratory reports and medical images, directly on-chain in permissioned blockchain platforms like Hyperledger Fabric introduces significant scalability and performance challenges. Although on-chain storage of binary data is technically feasible, it leads to rapid ledger growth, increased storage redundancy across peers, and long-term degradation of transaction throughput and system responsiveness [5–7]. Since each peer in a Fabric network maintains a full copy of the ledger and world state, embedding large files results in inefficient utilisation of storage resources and limits practical deployment in real-world healthcare environments [6].

Furthermore, Hyperledger Fabric is optimised for structured key-value data rather than unstructured binary objects, making native handling of large medical files inefficient [5]. The absence of partial file access and format-aware operations further restricts its suitability for managing large EHR documents [6]. These limitations necessitate the adoption of an off-chain storage mechanism for scalable blockchain-based EHR systems.

Table 1. Comparison of Off-Chain Storage

Aspect	Centralised Storage	Decentralised Storage	Distributed Storage
Architecture	Managed by a single trusted authority or server	Control is distributed across several authoritative nodes	Data is stored across multiple peer-to-peer nodes with no central control
Control	Single point of control	Shared control among trusted nodes	No central authority; fully peer-to-peer
Fault Tolerance	Low: failure of the central node affects the entire system [12]	Moderate: fault tolerance increases with redundancy [13]	High: redundant nodes provide resilience [10]
Scalability	Limited: bottlenecks under high load	Moderate: coordination overhead at scale [13,14]	High: scales with network growth [10,11]
Data Ownership	Controlled by the central authority	Shared control; partial user autonomy	Greater user control through decentralised access [10,11]
Security Risks	High: vulnerable to internal/external breaches [12]	Reduced: compromise requires attacking multiple nodes [13,14]	Lower: data encrypted and split across nodes
Complexity	Low: simple to deploy/manage [12]	Moderate: requires trust and coordination [13]	High: complexity in maintaining consistency and sync

Off-chain storage refers to storing medical data outside the blockchain while maintaining cryptographic references, such as hashes or Content Identifiers, on-chain. This approach preserves data integrity, auditability, and traceability while significantly reducing blockchain storage overhead and improving system scalability.

Off-chain storage solutions can be broadly classified into centralised, decentralised, and distributed models based on their architectural design and control mechanisms. Centralised storage solutions are simple to deploy but introduce single points of failure and trust concerns, making them less suitable for sensitive healthcare data [8, 9]. Decentralised storage improves fault tolerance by distributing control across multiple nodes, but incurs coordination and consistency overhead as the system scales. Distributed storage adopts a peer-to-peer architecture, offering high scalability, fault tolerance, and data resilience, making it well-suited for managing large EHR files in multi-institutional healthcare systems. Table 1 summarises the key differences among centralised, decentralised, and distributed off-chain storage models.

Numerous distributed storage platforms have been proposed, including Arweave [15], Filecoin [16], and IPFS. However, many platforms impose economic incentives or architectural constraints that limit their suitability for permissioned healthcare blockchains. Among available options, IPFS is particularly suitable for EHR applications due to its lightweight design, content-addressable storage, flexible deployment, and seamless integration with smart contracts [17]. Unlike incentive-driven systems, IPFS enables controlled access without requiring mandatory tokens, making it a practical choice for privacy-sensitive healthcare environments and a natural fit for the MrBlock+ framework.

2.2 Blockchain–IPFS-Based EHR Frameworks:

Blockchain–IPFS integration has gained significant attention for secure and scalable Electronic Health Record (EHR) management. Existing studies primarily aim to improve data privacy, decentralisation, and patient-centric control; however, they differ considerably in access control granularity, organisational awareness, and performance evaluation. This subsection reviews representative blockchain–IPFS-based EHR frameworks and highlights their limitations relative to MrBlock+.

The AB-DAM model [18] integrates blockchain and IPFS with Attribute-Based Encryption (ABE) for EHR protection. Although secure, it lacks organisation-aware authorisation and introduces computational overhead. Guo et al. [19] combine IPFS with edge computing and multi-authority ABE with homomorphic encryption, but do not support detailed metadata logging or organisation-level access policies. MedRSS [20] employs AES and ECC encryption for remote health monitoring but supports only coarse-grained access control. Similarly, Tran et al. [21] use Proxy Re-Encryption (PRE) for patient-managed data sharing; however, access control remains strictly patient-centric, limiting scalability in multi-institution environments.

Hyperledger-based solutions such as Healthchain [22] and the framework proposed by Sonkamble et al. [23] integrate IPFS with permissioned blockchains and support auditability and performance evaluation. Nevertheless, these systems do not enforce fine-grained role- or attribute-based authorisation or organisation-aware access control. Ethereum–IPFS-based approaches, including Boumezbeur et al. [24], focus on encryption and smart contracts but do not report runtime performance metrics required for real-time healthcare applications. HealthChain [26] further relies on deprecated Hyperledger Composer tooling, raising concerns regarding long-term maintainability.

To enable systematic comparison, blockchain–IPFS-based EHR systems are evaluated using four criteria: (i) IPFS integration for scalable off-chain storage, (ii) immutable on-chain storage of Content Identifiers (CIDs), (iii) fine-grained role- and attribute-based access control, and (iv) support for audit logging and performance evaluation. These criteria reflect both security requirements and real-world deployability. Table 2 presents a comparative analysis of existing frameworks against MrBlock+.

The comparison shows that while most frameworks integrate IPFS and store CIDs on-chain, they frequently lack organisation-aware access control, comprehensive audit logging, or rigorous performance benchmarking. In contrast, MrBlock+ uniquely combines hybrid RBAC–ABAC enforcement with organisational identity as a key access attribute, detailed metadata auditing, and empirical performance evaluation within a permissioned blockchain environment. Although file-level encryption is identified as a necessary security extension for real-world deployment, the

current design supports access-controlled EHR sharing by exposing IPFS Content Identifiers only through blockchain-verified authorisation within a controlled permissioned environment.

Table 2. Comparative analysis of the MrBlock+ framework with existing IPFS-integrated EHR systems

Reference	IPFS	CID on Chain	Access Control (Role + Attribute Based)	Audit Logs / Metadata	Performance Evaluation (File Upload/Download Matrix)
[18]	✓	✓	✗ ABE-only, not org-aware	✓ Basic	✓
[19]	✓	✓	✗ General (no org logic)	✓ Partial	✗
[20]	✓	✓	✗ Limited role control	✓ Partial	✗
[21]	✓	✓	✗ Patient-only access control	✗ Minimal	✗
[22]	✓	✓	✗ Patient-only (no ABAC)	✓ General audit support	✓
[23]	✓	✓	✗ Password/SPAKE-based (no ABAC)	✓ Performance logs	✓
[24]	✓	✓	✗ High-level policies only	✗ Not addressed	✗
[25]	✓	✓	✓ CP-ABAC with PRE	✓ Audit + performance	✓
[26]	✓	✓	✗ Basic ABAC with high-level policies	✓ Performance + Caliper	✓
MrBlock+	✓	✓	✓ Role + ABAC (Org ID as attribute)	✓ Detailed	✓

3. System Architecture Enhancement with IPFS Integration

The original MrBlock framework [4] was designed to leverage a permissioned Hyperledger Fabric blockchain network to provide secure access control and interoperability for Electronic Health Records (EHRs). In this architecture, the blockchain maintained a ledger of transactions governing access policies and stored small metadata about EHRs, while all clinical data were stored directly on-chain. While this approach ensured strong data integrity and fine-grained access control, it introduced inherent scalability limitations due to blockchain storage constraints. Storing large clinical files such as medical images (e.g. X-rays and MRIs) and detailed diagnostic reports on the blockchain led to excessive ledger growth and degraded system performance over time.

From a system design perspective, this limitation motivated the need for a hybrid on-chain/off-chain architecture that preserves blockchain guarantees while enabling scalable storage of large EHR files. To address these limitations, we extended the MrBlock architecture as MrBlock+ by integrating the Interplanetary File System (IPFS) as a decentralised off-chain storage solution. IPFS enables storage of large files in a distributed, peer-to-peer manner

while providing content-addressable, immutable data references. By offloading bulky EHR files to IPFS and storing only their cryptographic hashes – referred to as Content Identifiers (CIDs) – on the blockchain, the proposed architecture achieves scalability without compromising integrity, auditability, or access control enforcement. The system architecture is shown in Figure 1, and Figure 2 depicts the procedure for uploading health records to IPFS.

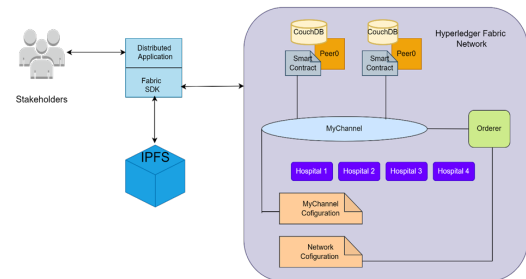


Figure 1. MrBlock+ System Architecture.

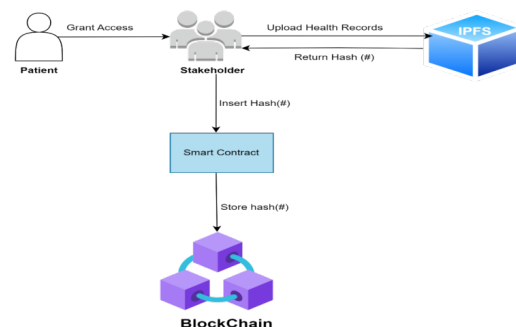


Figure 2. Procedure for Uploading Health Records to IPFS.

In this architecture, stakeholders such as patients, doctors, hospital administrators, lab technicians, and pharmacists interact with the system through a Distributed Application (DApp). The DApp serves as the secure front-end interface, allowing authenticated users to upload, retrieve, and manage health records. All user interactions are mediated by smart contracts deployed on the Hyperledger Fabric network, ensuring policy-driven and auditable operations.

When an EHR file is uploaded via the DApp, it is first stored in the IPFS network. IPFS generates a Content Identifier (CID), which is a unique cryptographic hash derived from the file's contents. This content-addressable mechanism guarantees data integrity, as any modification to the file would result in a different CID. Instead of storing the entire file on-chain, only the CID and associated metadata are recorded on the Hyperledger Fabric ledger through a smart contract. This design significantly reduces blockchain storage overhead while maintaining tamper resistance and traceability.

The architecture supports multiple hospitals connected through a shared channel in Hyperledger Fabric, facilitating controlled and privacy-preserving inter-organisational communication. Each hospital operates its own peer node, which maintains a copy of the ledger and executes smart contracts (chaincode). CouchDB is employed as the state database to enable rich queries and efficient retrieval of EHR metadata. An ordering service based on the Raft consensus

mechanism ensures consistent transaction ordering and fault tolerance across the network. Certificate Authorities (CAs) manage digital identities and issue cryptographic credentials, enabling secure authentication and fine-grained access control. By integrating IPFS with this permissioned blockchain infrastructure, MrBlock+ achieves a balance between scalability, security, and institutional interoperability. The roles of system stakeholders are summarised in Table 3.

Table 3. Role and Permission of Each Stakeholder

Stakeholder	Permissions
Admin	• Register all stakeholders from their respective hospitals. • View records of stakeholders. • In emergencies, Grant or revoke access to Electronic Health Records (EHRs) based on the stakeholder's role.
Patient	• View self-details. • Edit Personal details. • Grant or revoke of access to Electronic Health Records (EHRs) based on the role of the stakeholder involved. • View the history of your own data.
Doctor	• View Self details. • View patient details who have granted access. • Add prescription if needed.
Lab Technician (Pathologist)	• View Self details. • View patient details who have granted access. • Generate and upload lab reports. • Generate Bill.
Druggist (Pharmacist)	• View Self details. • View the patient details who have granted access. • Generate a bill for the medicine provided.

3.1 Access Control in File Upload and Retrieval Using IPFS

The access control mechanism implemented in the original MrBlock framework is retained and extended in MrBlock+ to ensure secure file operations over IPFS. The access model follows a hybrid Role-Based and Attribute-Based Access Control (RBAC–ABAC) approach, where both stakeholder role and organisational affiliation are evaluated at runtime.

As in MrBlock, hospital administrators and doctors belonging to the patient's registered hospital can access the patient's medical details without explicit patient approval. In contrast, external stakeholders – such as pharmacists, lab technicians, or doctors from other hospitals – must be explicitly granted access to the patient's data by the patient through the AccessRight smart contract. This contract governs both grant and revoke operations, ensuring that access decisions are enforced consistently and transparently at the blockchain level.

Only stakeholders with active access permissions are allowed to upload new medical files (e.g. lab reports or prescriptions) or retrieve existing records stored in IPFS. Once access is revoked, all corresponding permissions are immediately invalidated, and the stakeholder can no longer retrieve or modify the files. Because IPFS CIDs are exposed only after

successful smart contract verification, unauthorised entities cannot discover or access stored medical files.

To address emergency medical scenarios, the same structured emergency access control mechanism as MrBlock is followed. If a patient is unable to respond, the hospital administrator may temporarily grant access to external doctors or relevant medical personnel. All emergency access actions are immutably recorded on the blockchain, ensuring accountability, non-repudiation, and post-event auditability.

By integrating this robust access control policy with IPFS, MrBlock+ ensures that decentralised storage does not weaken confidentiality guarantees. Instead, access enforcement remains blockchain-governed, policy-driven, and verifiable.

Two smart contracts are implemented to support the secure upload and retrieval of medical records:

- **UploadMedicalReport:** Enables authorised stakeholders to upload medical documents (PDF/JPEG) to IPFS and securely store the resulting CID on the blockchain. Upload permissions are granted to the patient, as well as doctors and administrators from the patient's enrolled hospital.
- **ViewMedicalRecords:** Controls retrieval of medical records by enforcing access policies based on user role, organisational affiliation, and explicit patient consent. External stakeholders are required to obtain authorisation through the AccessRight smart contract before accessing IPFS CIDs.

These contracts are tightly integrated with the AccessRight contract, which performs all permission checks prior to execution. No upload or retrieval operation can be executed without successful authorisation, ensuring strict policy compliance.

Algorithm 1: UploadMedicalReport

1. **Input:** Receive the patient ID, IPFS hash/link, and medical report details.
If the patient ID or IPFS hash is missing, then terminate with an error.
Else, proceed to the next step.
2. **Check:** Retrieve the uploader's identity, role, and organisation.
If any of this information is missing, then terminate with an error.
3. **Verify:** Look up patient details using the patient ID.
If the patient record does not exist, then terminate with an error.
Else verify access:
 - *If access has already been granted, then* allow upload.
 - *If the uploader is a doctor, admin, or the patient from the same hospital, then* allow upload.*Else, terminate with an* unauthorised access error.

4. **Store:** Get the current timestamp.
Create a report object with all relevant information, including the uploader's identity and the IPFS hash.
Convert this object to JSON format and store it on the blockchain using the IPFS hash as the key.
5. **End:** Display a successful message and exit the function.

Algorithm 1: Algorithms to Upload Medical Documents on IPFS

Algorithm 2: ViewMedicalRecords

Input: patientID (string): The unique identifier of the patient whose medical records are to be retrieved.

Steps:

1. *If patientID is empty, then*
return error "patientID is required"
Else
proceed
2. Retrieve the identity of the user requesting the records
If error occurs then return error "failed to get user identity"
Else
proceed
3. Fetch user attributes including userRole and organisation
If error occurs then return error "failed to fetch attributes"
Else
proceed
4. Retrieve patient details from the ledger using patientID
If no patient record found then
return error "patient does not exist"
Else
proceed
5. Set accessFlag = false
6. Check access permissions for the user
If access is granted explicitly via AccessRight smart contract then
set accessFlag = true
Else if user is doctor, admin, or patient from same hospital then
set accessFlag = true
Else
return error "not authorised"
7. Query ledger for medical records associated with patientID
If no records found then
return error "no medical records found"
Else
proceed
8. For each record retrieved:
– Unmarshal the data
– Assign default values for any missing fields
– Append the processed record to a list
9. Return the complete list of medical reports

Algorithm 2: Algorithms to View Medical Documents Uploaded on IPFS

The algorithmic workflows explicitly demonstrate how access decisions are evaluated before interacting with IPFS, reinforcing the methodological rigor of the proposed system.

3.2 Practical Adoption Perspective of the Proposed Framework:

The adoption of blockchain-based EHR systems depends not only on technical performance but also on users' perception of usefulness, ease of use, trust, privacy, and risk. From a Technology Acceptance Model perspective, the proposed MrBlock+ framework may improve perceived usefulness by enabling secure and interoperable EHR sharing among patients, doctors, and healthcare organisations. Its access control mechanism, auditability, and off-chain storage support may reduce perceived privacy and security risks. Therefore, the proposed design has the potential to improve stakeholder trust and support practical adoption in healthcare environments.

4. Performance Evaluation of MrBlock+

Although the system is designed to support interaction through a decentralised application (DApp), the current implementation does not include a front-end interface. Instead, all functionalities, such as patient registration, access granting, report upload, and medical record viewing, were tested using RESTful API calls in Postman. This approach enables controlled, repeatable, and fine-grained validation of smart contract behaviour without front-end dependencies. This setup allowed for precise testing of each access control scenario, including denied access, granted access, emergency overrides, and revocations. Screenshots of these interactions are included to demonstrate the correct enforcement of smart contract logic.

4.1 Performance Evaluation of Fine-Grained Access Control Through Test Cases

The access control mechanisms implemented within the UploadMedicalReport and ViewMedicalRecord smart contracts play a crucial role in preserving data privacy and enabling secure interoperability across multiple hospitals. To assess the effectiveness of these mechanisms, MrBlock+ was tested against a range of predefined scenarios involving different user roles and access conditions. These test cases were designed to validate both role-based permissions and organisational attribute enforcement. In all cases, MrBlock+ behaved as expected, accurately enforcing access permissions based on the defined policies. Table 4 outlines the test cases used to evaluate MrBlock+'s performance.

Table 4. Summary of Test Cases for System Evaluation

Test Case ID	User Role	Operation	Access Status	Expected Outcome	Actual Result	Status
TC1	Doctor(same hospital)	ViewMedicalRecords	Access not required	The record should be viewable	Record retrieved	Pass
TC2	Doctor(other hospital)	ViewMedicalRecords	Access not granted	Access should be denied	Access denied	Pass
TC3	Doctor(other hospital)	ViewMedicalRecords	Access granted	The record should be viewable	Record retrieved	Pass
TC4	Druggist, Pathologist (any hospital)	ViewMedicalRecords	Access not granted	Access should be denied	Access denied	Pass
TC5	Druggist, Pathologist (any hospital)	ViewMedicalRecords	Access granted	The record should be viewable	Record retrieved	Pass
TC6	Patient	UploadMedicalReport	Direct access	Upload should be successful	Report uploaded	Pass
TC7	Doctor(same hospital)	UploadMedicalReport	Direct access	Upload should be successful	Report uploaded	Pass
TC8	Doctor(other hospital)	UploadMedicalReport	Access not granted	Upload should be denied	Upload denied	Pass
TC9	Doctor(other hospital)	UploadMedicalReport	Access granted	Upload should be successful	Report uploaded	Pass
TC10	Druggist, Pathologist (any hospital)	UploadMedicalReport	Access not granted	Upload should be denied	Upload denied	Pass
TC 11	Druggist, Pathologist (any hospital)	UploadMedicalReport	Access granted	Upload should be successful	Report uploaded	Pass

The results of each test case were verified through RESTful API calls using Postman. To demonstrate MrBlock+'s behaviour, sample screenshots are included for key scenarios such as access denial, access granting, and post-grant access. These results confirm the correctness and robustness of the hybrid RBAC–ABAC policy enforcement implemented in the smart contracts.

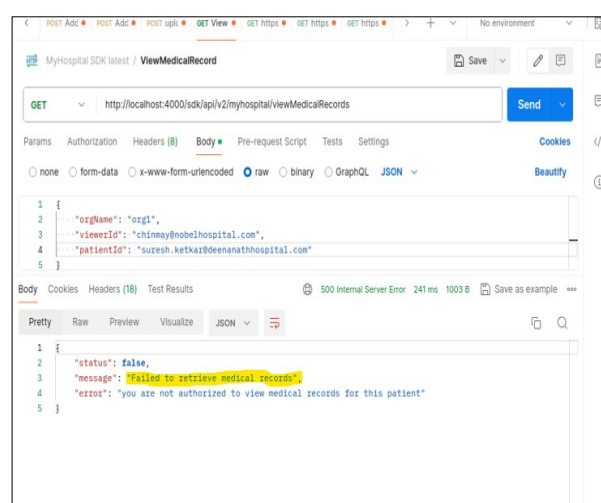
Scenario 1: Unauthorised Access Attempt:

Patient Suresh Ketkar has his medical record uploaded by Dr. Prashant, a physician at Deenanath Hospital, where the patient is registered. Subsequently, Dr. Chinmay from Nobel Hospital attempts to access the patient's medical record. However, since the patient has not granted access, the system denies the request in accordance with the defined access control policies, thereby ensuring the confidentiality of the patient's medical data. This scenario validates the enforcement of organisational constraints across hospital boundaries. Figure 3 displays the system's response to this unauthorised access attempt.

Scenario 2: Authorised Access After Grant:

In continuation of the previous case, the patient grants access to Dr. Chinmay through the smart contract-based access control mechanism. Once permission is granted, Dr. Chinmay is authorised to retrieve the patient's medical record stored on IPFS. Figure 4 confirms the successful grant of access, while Figure 5 demonstrates successful record retrieval. This

scenario highlights the dynamic and patient-controlled nature of access authorisation in MrBlock+.

**Figure 3.** Unauthorised Access Denial for External Doctor.

4.2 Performance Evaluation of File Upload and Download Time

To further evaluate MrBlock+'s efficiency beyond access control, we measured the upload and download times for Electronic Health Records (EHRs). These operations were executed through RESTful API calls without a front-end

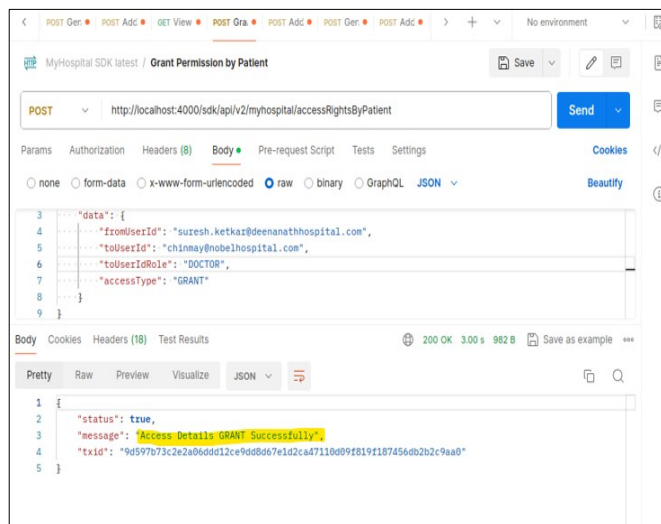


Figure 4. Access Grant Confirmation by Patient.

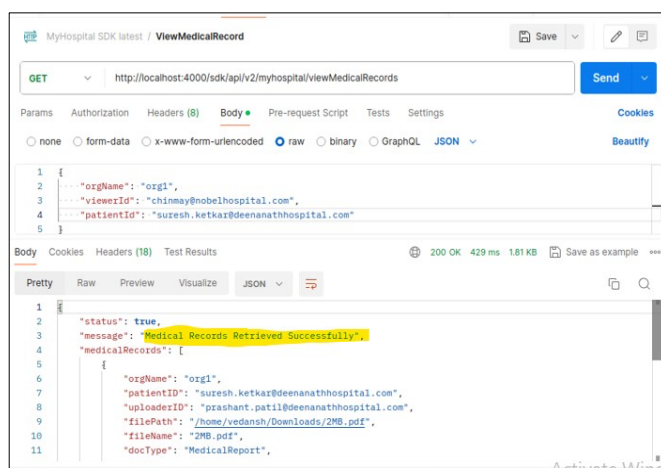


Figure 5. Retrieval of Patient's Record After Access is Granted.

interface, ensuring a precise and controlled testing environment. The tests were conducted on files of increasing sizes (20 MB to 100 MB) to analyse system responsiveness under varying data loads. Table 5 presents the system configuration used for implementation and evaluation.

The upload time is the time required to store the file on IPFS via the blockchain, while the download time is the time required to retrieve it. These metrics are critical indicators of scalability and real-time usability in healthcare environments. Figure 6 illustrates the upload and download performance.

Figure 6(a) and Figure 6(b) show a clear upward trend in both upload and download times as file size increases, which is expected due to larger data volumes. However, the growth remains moderate, indicating efficient handling of large medical files. For example, upload time increases from 17.7 seconds for a 20 MB file to 36.62 seconds for a 100 MB file, showing a controlled linear trend. This behaviour demonstrates the effectiveness of off-chain IPFS storage in reducing blockchain overhead.

Download latency remains consistently low, increasing only from 153 ms to 251 ms across the same file range. Such low retrieval

latency confirms MrBlock+'s suitability for real-time EHR access, even for large clinical files. These results validate the scalability of MrBlock+ without introducing prohibitive delays.

4.3 Comparative Analysis with Existing Systems

To comprehensively assess the performance of MrBlock+, a comparative analysis was conducted against existing blockchain-based EHR frameworks proposed by Mani et al. [22] and Chentharra et al. [26]. All three systems use Hyperledger Fabric and IPFS-based off-chain storage. However, they differ substantially in terms of access control design, transaction flow optimisation, and data retrieval mechanisms.

4.3.1 Selection Criteria for Comparative Analysis

For the comparative analysis, the frameworks proposed by Mani et al. [22] and Chentharra et al. [26] were selected because they are closely related to blockchain-based healthcare/EHR data sharing and provide measurable performance results. Both studies report upload and download times for medical record files, making them suitable for quantitative comparison with the proposed MrBlock+ framework. Since the present work evaluates the upload and retrieval of PDF-based medical records using IPFS-based off-chain storage, studies that do not report comparable upload/download time metrics were excluded from the quantitative comparison. Therefore, the comparison focuses on systems that address secure healthcare data sharing and provide performance values relevant to medical file upload and retrieval operations.

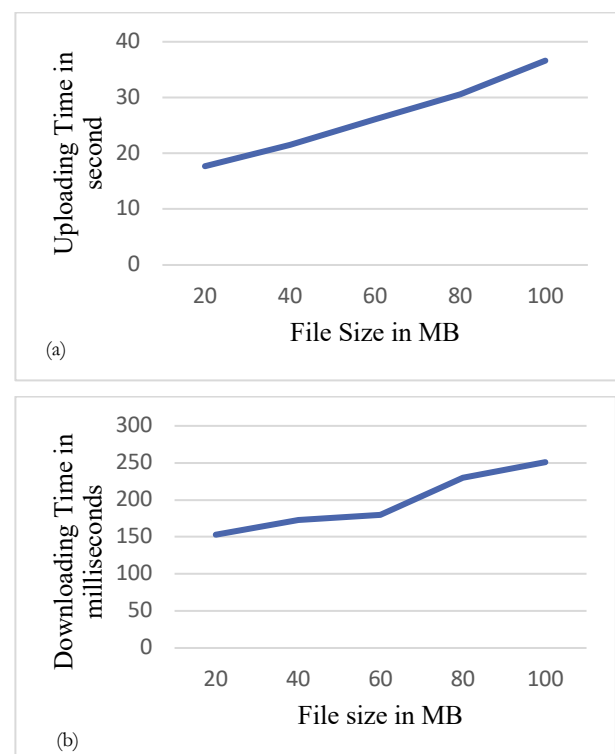


Figure 6. Performance Evaluation of MrBlock+: (a) Upload Time of Medical Records, (b) Download Time of Medical Records.

Table 5. System Configuration

Component	Configuration
Processor	Intel® Core™ i5-1035G1 CPU @ 1.00GHz × 8
Memory	8 GB
Operating system	Ubuntu 20.04.6 LTS
Docker engine	Version 24.0.7
Docker compose	Version v2.21.0
Language Used	Go for a smart contract
Hyperledger Fabric	Fabric 2.4

Table 6. Performance Comparison of Mani [22], Chenthara [26] and MrBlock+

Upload Time in Seconds				Download Time in Milliseconds		
File Size in MB	Mani [22]	MrBlock+	Chenthara [26]	Mani [22]	MrBlock+	Chenthara [26]
20	18	17.7	13	20,000	153	15,000
40	25	21.5	27	32,000	173	15,000
60	38	26.14	35	40,000	180	45,000
80	50	30.61	50	75,000	230	85,000
100	58	36.62	65	105,000	251	110,000

Table 6 compares upload and download times across varying file sizes, while Figures 6(a) and 6(b) illustrate these results graphically.

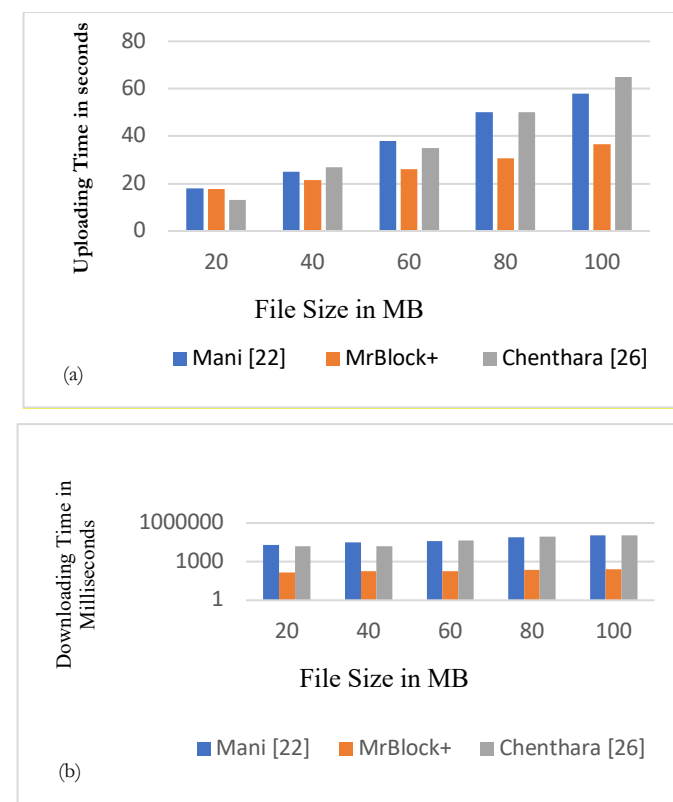


Figure 7. Comparative Performance Evaluation: (a) Upload Time, (b) Download Time for Mani et al. [22], Chenthara et al. [26], and MrBlock+.

Figures 7(a) and 7(b) show that while all systems experience increased processing time with larger files, MrBlock+ demonstrates improved performance under the evaluated experimental conditions. In upload operations, MrBlock+ achieves a 30–45% reduction in upload time compared to

Mani et al. and Chenthara et al., reflecting optimised chaincode execution and reduced transaction overhead.

The download performance shows even more significant gains. MrBlock+ retrieves a 100 MB file in just 251 ms, whereas Mani's and Chenthara's systems require over 100,000 ms. This observed reduction of more than 99.7% indicates lower reported download latency under the evaluated file-retrieval conditions.

Overall, these results indicate that MrBlock+ provides promising baseline scalability, low-latency retrieval, and efficient file upload and retrieval under the evaluated experimental conditions. The combination of hybrid access control and off-chain storage enables secure interoperability without sacrificing real-time responsiveness, making MrBlock+ well-suited for practical healthcare deployment.

The performance results reported in this study are based on a local experimental testbed configured on a single machine. Therefore, the reported latency, throughput, upload time, and download time values should be interpreted within the scope of this controlled environment rather than as definitive results for large-scale deployment. In a realistic multi-hospital Hyperledger Fabric network, performance may vary due to factors such as inter-organisational network latency, ordering service overhead, endorsement policy complexity, number of peer nodes, concurrent transaction load, bandwidth availability, and IPFS node distribution. Future work will evaluate MrBlock+ in a distributed multi-organisation testbed to assess scalability, latency, throughput, and reliability under realistic healthcare network conditions.

5. Security Analysis

Security analysis is essential for blockchain-based Electronic Health Record (EHR) systems due to the sensitivity of medical data and the involvement of multiple stakeholders across institutions. Since MrBlock+ operates in a multi-hospital environment, the framework must ensure confidentiality, integrity, availability, authentication, and authorisation under

Table 7. CIA+AA Evaluation of MrBlock+

Sr.No	Security Property	Description	Mechanisms in MrBlock+
1	Confidentiality	Prevents unauthorised data access.	PKI, X.509 certificates, RBAC–ABAC, restricted IPFS hash sharing.
2	Integrity	Ensures that data cannot be modified illegally.	Immutable ledger, hashing, Raft log consistency.
3	Availability	System remains accessible to authorised users.	Raft crash fault tolerance, distributed peers.
4	Authentication	Validates the identity of users and nodes.	X.509 certificates issued by CA.
5	Authorisation	Controls user permissions.	Hybrid RBAC–ABAC in smart contracts.

realistic threat conditions. Accordingly, the security of MrBlock+ is evaluated using the CIA+AA security model and STRIDE-based threat modelling.

The CIA+AA model extends the traditional CIA triad by incorporating authentication and authorisation, which are critical in permissioned healthcare blockchains. In MrBlock+, confidentiality is enforced through permissioned network access, PKI-based digital identities, and smart contract–driven hybrid RBAC–ABAC policies. Data integrity is ensured by blockchain immutability and IPFS content addressing, while availability is maintained through distributed peers and Raft-based fault tolerance. Authentication and authorisation are enforced using X.509 certificates issued by a trusted Certificate Authority and policy-controlled smart contracts. Table 7 summarises the CIA+AA security properties and the corresponding mechanisms implemented in MrBlock+.

To evaluate resilience against practical attack scenarios, MrBlock+ is further analysed using the STRIDE threat model. Spoofing, tampering, repudiation, information disclosure, denial-of-service, and privilege escalation threats are mitigated through certificate-based identity verification, immutable blockchain logs, controlled IPFS hash sharing, distributed consensus, and strict RBAC–ABAC enforcement in smart contracts. Table 8 presents the STRIDE-based analysis of threats and mitigation mechanisms in MrBlock+.

Although MrBlock+ restricts IPFS CID exposure through smart contract-based access control, restricted hash sharing should not be considered equivalent to cryptographic confidentiality. Since IPFS content is retrievable using its CID, leakage of the CID through node compromise, insider threat, logging exposure, or network-level discovery may allow unauthorised retrieval of the corresponding medical file if the file is stored without encryption. Therefore, unencrypted IPFS storage introduces a residual information disclosure risk.

In the current implementation, this risk is mitigated only under controlled deployment assumptions, where IPFS nodes are operated by trusted healthcare institutions, access to CIDs is restricted through authorisation policies, network access is limited, and all access attempts are logged and auditable. Under these assumptions, deployment without file-level encryption may be acceptable for prototype evaluation or permissioned institutional environments. However, for real-world multi-hospital deployment, file-level encryption is required to provide cryptographic protection even if a CID is leaked. Therefore, future versions of MrBlock+ will integrate encryption-based mechanisms such as Attribute-Based

Encryption (ABE) and Proxy Re-Encryption (PRE) to strengthen the confidentiality of medical images and diagnostic reports stored on IPFS.

Table 8. STRIDE Analysis of MrBlock+

STRIDE Category	Threat Description	Mitigation in MrBlock+
S — Spoofing	Attackers impersonate a legitimate user or node.	X.509 certificates, PKI authentication, permissioned network, CA-based identity issuance.
T — Tampering	Unauthorised modification of data or transactions.	Immutable ledger, hashing, Raft log replication, RBAC–ABAC policies.
R — Repudiation	User denies performing an action or transaction.	Digital signatures, transaction logs, and immutable records.
I—Information Disclosure	Leakage of sensitive medical data.	RBAC–ABAC access control, restricted IPFS hash sharing, secure authentication.
D—Denial of Service (DoS)	The system becomes unavailable due to overload or attacks.	Raft crash fault tolerance, distributed peers, and permissioned access.
E—Elevation of Privilege	Attackers gain higher-level permissions.	RBAC–ABAC enforcement, smart contract checks, certificate-bound identities.

Overall, the combined CIA+AA and STRIDE analysis confirms that MrBlock+ provides a defence-in-depth security architecture suitable for secure, auditable, and resilient EHR management in multi-institutional healthcare environments.

5.1 Healthcare Data Regulation and Compliance Considerations

Healthcare data sharing across multiple institutions must comply with applicable data protection and privacy regulations, such as the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and relevant national data protection frameworks. Since Electronic Health Records contain sensitive personal and medical information, any EHR sharing framework must support consent, privacy, restricted access, auditability, and accountability.

MrBlock+ is designed to support these regulatory-aligned principles through patient consent, controlled access authorisation, and auditable record sharing. Unlike purely patient-centric models, MrBlock+ balances patient autonomy with structured access control mechanisms. The framework

includes a provision for obtaining patient consent during the registration process to store health information on the blockchain and to share EHRs in emergencies. In critical situations, the hospital administrator can grant access to a patient's EHR to support timely medical intervention, while maintaining controlled and auditable access.

However, the current implementation of MrBlock+ should not be considered a complete compliance-certified deployment. Formal compliance validation with GDPR, HIPAA, or jurisdiction-specific healthcare data protection laws is outside the scope of the present study. Further work is required to address institution-specific policy mapping, data retention rules, breach notification procedures, right-to-erasure requirements, and jurisdiction-specific deployment constraints. These aspects are considered important directions for future development and real-world deployment of the proposed framework.

6. Conclusion and Future Scope

This paper presented MrBlock+, an extension of the MrBlock blockchain-based framework, designed to enhance scalable, interoperable, and access-controlled Electronic Health Record (EHR) management. By integrating IPFS with Hyperledger Fabric, the proposed system enables decentralised storage of large medical files while preserving verifiable access metadata on-chain. MrBlock+ implements fine-grained role- and attribute-based access control (ABAC), where organisational identity is treated as a critical attribute, ensuring that only authorised stakeholders from approved institutions can upload or retrieve EHRs. Smart contracts govern access operations and maintain an immutable audit trail of all data interactions.

The framework was evaluated using real-world scenarios involving the upload and retrieval of PDF-based medical records ranging from 20 MB to 100 MB. Comparative performance analysis against representative blockchain-based EHR frameworks proposed by Mani et al. [22] and Chenthara et al. [26] demonstrates improved performance over the selected benchmark systems. The proposed system achieves up to a 45% reduction in upload latency compared with selected benchmark systems. MrBlock+ also shows lower reported download latency than the selected benchmark system under the evaluated file-retrieval conditions, with an observed reduction of over 99.7% for a 100 MB file.

MrBlock+ adopts a patient-centric access control model that empowers patients to manage permissions for their health records, while also supporting administrator-assisted access in emergency scenarios where patient intervention may not be feasible. The combination of decentralised storage, smart contract-enforced policies, and organisation-aware authorisation makes MrBlock+ a practical solution for secure and timely EHR sharing across multi-institutional healthcare networks.

In addition to performance evaluation, the security of MrBlock+ was analysed using the CIA+AA model and STRIDE-based threat modelling. The analysis confirms resilience against common threats such as unauthorised access, data tampering, identity spoofing, privilege escalation, and

denial-of-service attacks. Permissioned blockchain access, PKI-based authentication, smart contract-driven authorisation, blockchain immutability, and IPFS content addressing together provide a defence-in-depth security architecture for multi-hospital EHR sharing. MrBlock+ supports access-controlled EHR sharing by exposing IPFS Content Identifiers (CIDs) only to blockchain-authorised users; however, this mechanism is treated as an access-control measure rather than a substitute for file-level encryption.

Nevertheless, this study has certain limitations that should be considered while interpreting the results. The evaluation was conducted in a controlled local testbed rather than a real distributed multi-hospital environment. Therefore, the reported latency, throughput, upload time, and download time may vary under real-world conditions due to multiple peer nodes, ordering service latency, network delay, endorsement policy complexity, and concurrent user load. The current implementation was tested using RESTful API calls and does not include pilot testing with real healthcare stakeholders such as doctors, patients, and hospital administrators. In addition, file-level encryption for IPFS-stored medical records and formal compliance validation with GDPR, HIPAA, and national data protection laws remain important future enhancements.

Hence, future work will focus on validating MrBlock+ in more realistic healthcare deployment environments. This includes pilot testing with real healthcare stakeholders to evaluate usability, acceptance, and practical effectiveness. Further evaluation will also be conducted in a distributed multi-organisation testbed involving multiple peer nodes, ordering services, IPFS nodes, and inter-institutional network conditions. Advanced cryptographic mechanisms such as Attribute-Based Encryption (ABE) and Proxy Re-Encryption (PRE) will be integrated to ensure that medical records are encrypted before being uploaded to IPFS, while the blockchain stores only the corresponding IPFS hash and access metadata. Future extensions will also include integration with existing hospital information systems and mapping MrBlock+ to applicable healthcare data protection regulations, including data retention and deletion mechanisms, consent withdrawal workflows, breach notification procedures, institution-specific compliance policies, and legal validation for real-world healthcare deployment.

Competing interests:

The authors declare that they have no competing interests.

Authors' contribution:

Pranalini Joshi conceived, designed, and implemented the study, analysed the data, drafted the manuscript and performed proofreading and revisions. Pratibha Mahajan provided guidance on the study design, supervised the research process, reviewed the manuscript, and approved the final version.

Funding:

The authors declare that no funding was received for the conduct of this study or the preparation of this manuscript.

Acknowledgements:

The authors would like to express their sincere gratitude to all those who provided support and guidance during this study. We are especially thankful to Vishwakarma University, Pune, India, for its encouragement, support, and assistance throughout the research process.

References:

- [1] Centers for Disease Control and Prevention (CDC), "National trends in hospital and physician adoption of electronic health records," 2021. Available: <https://www.healthit.gov/data/quickstats/national-trends-hospital-and-physician-adoption-electronic-health-records>.
- [2] Market.us, "Electronic health records market size & share statistics," 2025. Available: <https://media.market.us/electronic-health-records-statistics/>.
- [3] U.S. Department of Health and Human Services, "Challenges to public health reporting experienced by non-federal acute care hospitals," 2021. Available: <https://www.healthit.gov/sites/default/files/page/2021-09/Challenges-to-PH-Reporting.pdf>.
- [4] P. Joshi and P. Mahajan, "Secure and interoperable EHR management via hyperledger fabric: The Mrblock framework," *Peer-to-Peer Netw. Appl.*, vol. 18, no. 3, pp. 1–23, 2025.
- [5] Hyperledger Fabric Documentation, The Linux Foundation Projects. 2023. Available: <https://hyperledger-fabric.readthedocs.io>.
- [6] B. Putz, F. Menges, and G. Pernul, "A secure and auditable logging infrastructure based on a permissioned blockchain," *Comput. Secur.*, vol. 87, pp. 101602, 2019. <https://doi.org/10.1016/j.cose.2019.101602>.
- [7] 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. <https://doi.org/10.1093/jamia/ocx068>.
- [8] D. B. Park, X. Li, A. M. Shahhosseini, and L. S. Tsay, "Data ownership in cloud: Legal issues," *Int. J. Forensic Eng. Manag.*, vol. 1, pp. 125–148, 2021. [CrossRef]
- [9] S. Alboaie and D. Cosovan, "Private data system enabling self-sovereign storage managed by executable choreographies," in *Lecture Notes in Computer Science* (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Neuchâtel, Switzerland: Springer, Volume 10320, pp. 83–98, 2017.
- [10] J. Benet, IPFS – Content addressed, 2014. Versioned, P2P File System. <https://ipfs.io>.
- [11] J. Xie et al., "A survey of blockchain technology applied to smart cities," *IEEE Comm. Surv. Tutor.*, vol. 21, no. 3, pp. 2794–2830, 2019.
- [12] Y. Zhang et al., "Smart contract-based access control for the IoT," *IEEE IoT J.*, vol. 6, no. 2, pp. 1594–1605, 2018.
- [13] M. S. Ali et al., "Applications of blockchains in the IoT: A comprehensive survey," *IEEE Comm. Surv. Tutor.*, vol. 23, no. 2, pp. 1172–1196, 2021.
- [14] W. Li and H. Song, "ArtChain: Blockchain-based multimedia provenance tracking," *IEEE Trans. Multimed.*, vol. 21, no. 5, pp. 1181–1193, 2018.
- [15] Williams, Sam, Abhav Kedia, Lev Berman, and Sebastian Campos-Groth. "Arweave: The Permanent Information Storage Protocol." Arweave, Dec 26 (2023).
- [16] <https://filecoin.io/filecoin.pdf>.

- [17] M. M. Merlec and H. Peter In, "Blockchain-based decentralized storage systems for sustainable data self-sovereignty: A comparative study," *Sustainability*, vol. 16, no. 17, p. 7671, 2024.
- [18] K. Satheesh and S. V. Sree, "AB-DAM: Attribute-based data access model in blockchain for healthcare applications," *Multimed. Tools Appl.*, vol. 81, pp. 23567–23588, 2022. <https://link.springer.com/article/10.1007/s11042-022-12674-w>.
- [19] H. Guo, W. Li, M. Nejad, and C.-C. Shen, "A hybrid blockchain-edge architecture for electronic health record management with attribute-based cryptographic mechanisms," 2023. arXiv preprint, arXiv:2305.19797.
- [20] Z. Sun, D. Han, D. Li, T.-H. Weng, K.-C. Li, and X. Mei. "MedRSS: A blockchain-based scheme for secure storage and sharing of medical records," *Comput. Ind. Eng.*, vol. 183, p. 109521, 2023.
- [21] P. Tran, T. Nguyen, L. Chu, N. Tran, and H. Ta, "A solution for commercializing, decentralizing and storing electronic medical records by integrating proxy re-encryption, IPFS, and blockchain," 2024. arXiv preprint, arXiv:2402.05498.
- [22] V. Mani, P. Manickam, Y. Alotaibi, S. Alghamdi, and O. Ibrahim Khalaf, "Hyperledger healthchain: Patient-centric IPFS-based storage of health records," *Electronics*, vol. 10, no. 23, p. 3003, 2021.
- [23] R. Ganpatrao Sonkamble, A. M. Bongale, S. Phansalkar, A. Sharma, and S. Rajput, "Secure data transmission of electronic health records using blockchain technology," *Electronics*, vol. 12, no. 4, p. 1015, 2023.
- [24] I. Boumezbeur, K. Zarour, D. Keddari, F. Boutouatou, Y. Nassim Benzagouta, I. Harkat, and M. Seghiri, "Secure EHR sharing using blockchain and IPFS," *Stud. Sci. Sci.*, vol. 42, no. 7, pp. 1–14, 2024. ISSN: 1003-2053.
- [25] V. Sharma, N. Lal, and A. Sharma, "Performance evaluation using throughput and latency of a blockchain-enabled patient centric secure and privacy preserve EHR based on IPFS," *Int. J. Grid Util. Comput.*, vol. 15, no. 1, pp. 16–30, 2024.
- [26] S. Chenthara, K. Ahmed, H. Wang, F. Whittaker, and Z. Chen, "Healthchain: A novel framework on privacy preservation of electronic health records using blockchain technology," *Plos One*, vol. 15, no. 12, p. e0243043, 2020.