Volume: 13 Issue: VIII Month of publication: August 2025

DOI: https://doi.org/10.22214/ijraset.2025.73559

Blockchain-Enabled EMR Protection: A Smart Contract and IPFS-Based Architecture

V Manideep
M.Tech, Department of Computer Science and Engineering, UCEK, JNTU Kakinada, Andhra Pradesh, India

Abstract: The exponential growth in digital healthcare infrastructure has resulted in an overwhelming increase in sensitive medical data generation. However, traditional centralized Electronic Medical Records (EMR) systems continue to face critical security and privacy challenges. These include single points of failure, limited interoperability, data tampering, and unauthorized access. This paper introduces a robust and scalable blockchain-based framework for secure EMR management. Leveraging Ethereum blockchain, IPFS decentralized storage, and smart contracts, the framework ensures tamper-proof data logging and fine-grained access control. The system stores encrypted patient health records on IPFS and logs the corresponding content identifier (CID) on the Ethereum blockchain, eliminating the risk of data exposure. The architecture is designed for future compatibility with Mobile Edge Computing (MEC), allowing for faster data processing closer to the point of care. By offering immutable audit trails, decentralized access governance, and high availability, the proposed framework ensures transparency, security, and data ownership for all healthcare stakeholders.

Keywords: Blockchain, Electronic Medical Records, Smart Contracts, IPFS, Ethereum, PoA, Data Security, Decentralized Access Control.

I. INTRODUCTION

As global healthcare systems transition to electronic formats, safeguarding Electronic Medical Records (EMRs) has become a critical priority. Traditional models rely on centralized servers, exposing data to security breaches, downtime, and unauthorized tampering [1]. Moreover, patient privacy is often compromised due to third-party access and lack of transparency. Blockchain technology presents a decentralized alternative that guarantees data immutability, transparency, and distributed trust [2][3].

The integration of blockchain and decentralized storage systems offers significant improvements in healthcare data management. Ethereum smart contracts provide programmable access control mechanisms, enabling role-based, time-bound, and revocable permissions. IPFS offers a distributed way to store encrypted health records while maintaining global accessibility. This system ensures that only authorized entities can access medical records, while all access and modifications are immutably logged on the blockchain. This study presents a hybrid architecture that combines Ethereum blockchain for decentralized logging, IPFS for distributed file storage, and smart contracts for managing access permissions. The proposed model ensures end-to-end confidentiality, integrity, and auditability for secure EMR exchange. While anomaly detection and encryption schemes were considered, this work focuses purely on the core blockchain integration and decentralized data-sharing mechanisms. Future extensions may incorporate AI modules and edge-based analytics for intelligent diagnostics.

II. LITERATURE REVIEW

Blockchain applications in healthcare have been studied extensively for their potential to secure patient records, enable data interoperability, and enforce policy-driven access control [4][5][6]. MedRec by Azaria et al. [7] introduced one of the first blockchain-based systems for medical data sharing, empowering patients with ownership and access management rights. Liang et al. [8] explored encryption-enhanced models using blockchain and proposed multi-layered privacy preservation for EHRs.

IPFS, a peer-to-peer distributed file system, complements blockchain by addressing the challenge of scalable off-chain storage [9]. When paired with Ethereum smart contracts, it enables efficient and secure referencing of files without bloating the blockchain. The system can use cryptographic CIDs to maintain linkage between blockchain logs and off-chain data, without compromising privacy [10].

Smart contracts also enable dynamic access control strategies. Role-based access enforcement, revocation mechanisms, and fine-grained constraints (e.g., time-bound viewing rights) can be implemented transparently and automatically [11]. Platforms like Hyperledger and Ethereum have demonstrated successful use cases in this domain, with Ethereum offering the additional benefit of public accessibility and standard tooling.

Compared to traditional EHR systems and cloud-based alternatives, decentralized architectures are inherently more robust to censorship, data tampering, and outages [12]. By eliminating central points of failure and empowering patients with consent-based access flows, blockchain-based systems can comply more easily with international healthcare regulations such as HIPAA and GDPR [13].

Our proposed model builds on these ideas, focusing on modular smart contract-based access control and decentralized IPFS storage, eliminating the inclusion of external ML models or complex encryption mechanisms in its current version. Future work may explore advanced components such as anomaly detection engines and zero-knowledge proofs for enhanced privacy.

III. SYSTEM ARCHITECTURE

The architecture is composed of three primary layers:

A. Data Storage Layer

Health records are uploaded and stored on IPFS.
Each file generates a unique Content Identifier (CID).
Files can be optionally encrypted prior to upload using local mechanisms.

B. Blockchain Access Layer

Ethereum smart contracts handle permissions, access requests, and metadata.
The CID from IPFS is recorded in a blockchain transaction.
Transactions include the uploader's identity, access rights, and timestamp.

C. Access Control Mechanism

Smart contracts define who can view/download records.
Patients can grant/revoke access rights.
All interactions (requests, grants, views) are immutably logged.

IV. SYSTEM WORKFLOW OVERVIEW

The workflow of the proposed decentralized EMR system is modular, ensuring end-to-end integrity and transparency. The major functional steps are as follows:

1) User Authentication

- Users (patients, doctors, admins) authenticate via MetaMask, which provides a secure Ethereum wallet for transaction signing.

2) Data Upload

- A patient uploads their medical record using the React frontend.
- The file is sent to IPFS via the backend Flask API and returns a unique CID.

3) Blockchain Logging

- The CID, along with metadata such as uploader identity and timestamp, is submitted to the Ethereum smart contract.
- The transaction is logged immutably.

4) Access Control Enforcement

- Smart contracts define which Ethereum addresses (e.g., doctors) are allowed to view or download the CID-linked data.
- Access rights can be modified or revoked by the data owner (patient).

5) Data Access & Auditing

- A doctor requests access.
- If permitted by the smart contract, access is granted and the event is logged on-chain.
- All actions (upload, view, grant/revoke) are visible via a blockchain explorer.

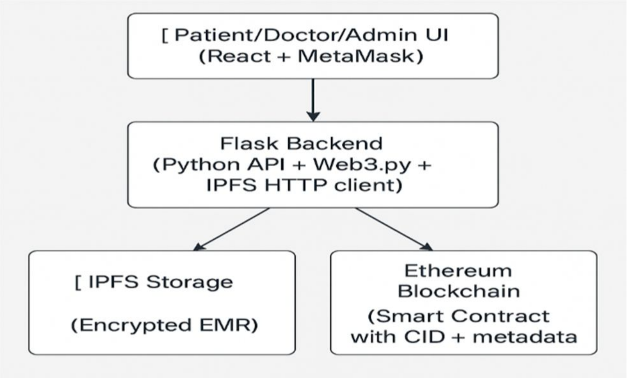


Fig 3.1 Workflow

V. METHODOLOGY

The methodology outlines the structured approach taken to implement the decentralized EMR management system. The following steps were followed:

1) Requirement Analysis
- Identify key challenges in EMR security including unauthorized access, data tampering, and centralized dependency.
- Define roles: patients (data owners), doctors (data viewers), and admins (validators).

2) System Design
- Design the architecture with three layers: Data Storage, Blockchain Access, and Access Control.
- Define smart contract logic for logging CIDs and managing access control.

3) Smart Contract Development
- Write the smart contract in Solidity.
- Include functions for CID logging, user identification, and view access rights.
- Deploy smart contract on Ganache (local Ethereum test network).

4) IPFS Integration
- Connect to IPFS using Kubo CLI.
- Upload encrypted EMR files.
- Retrieve unique CIDs for blockchain logging.

5) Backend Development
- Develop a Flask backend using Python.
- Integrate Web3.py for Ethereum interaction and ipfshttpclient for IPFS operations.
- Expose API endpoints for file upload, blockchain logging, and access management.

6) Frontend Integration
- Build UI using React.js.
- Integrate MetaMask for Ethereum account connection and transaction signing.
- Implement components for file selection, upload, access requests, and CID retrieval.

7) Testing & Validation
- Perform end-to-end testing across modules.
- Validate correct CID logging, permission enforcement, and UI responsiveness.
- Measure latency for file upload and access request.

8) Result Evaluation

- Analyze logs for successful file uploads, access requests, and smart contract transactions.
- Measure system performance in terms of transaction speed, upload time, and contract execution.

This structured methodology ensures the systematic development and validation of a blockchain-powered EMR solution that is modular, secure, and scalable.

VI. IMPLEMENTATION TOOLS

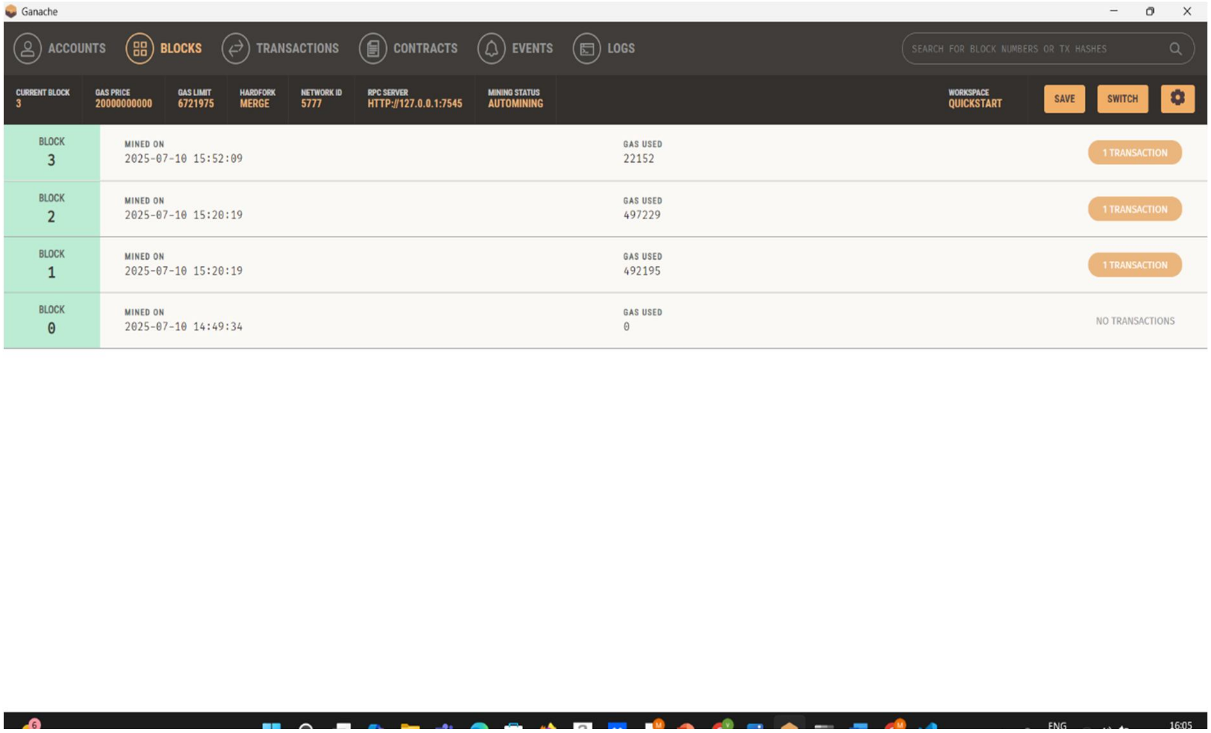
1) Frontend: React.js, Bootstrap
2) Backend: Flask (Python), Web3.py
3) Blockchain: Ethereum (Ganache), Solidity, Truffle Suite
4) Storage: IPFS (Kubo CLI or HTTP API)
5) User Wallets: MetaMask for user authentication and transaction signing

VII. RESULTS & ANALYSIS

1) Transaction Logs: All uploaded files and access requests were successfully logged on the local Ethereum blockchain.
2) Access Enforcement: Role-based access logic was correctly enforced using smart contracts.
3) System Performance: Upload and retrieval latency through IPFS averaged under 3 seconds.
4) User Interface: File upload and access permissions were managed smoothly via the React UI and MetaMask wallet integration.

Patient ID	File CID	Access Granted To	Timestamp
P001	QmX...1z	Doctor_A	2025-07-31
P002	QmY...8k	Doctor_B	2025-07-31

Fig 6.1 Sample Output Table

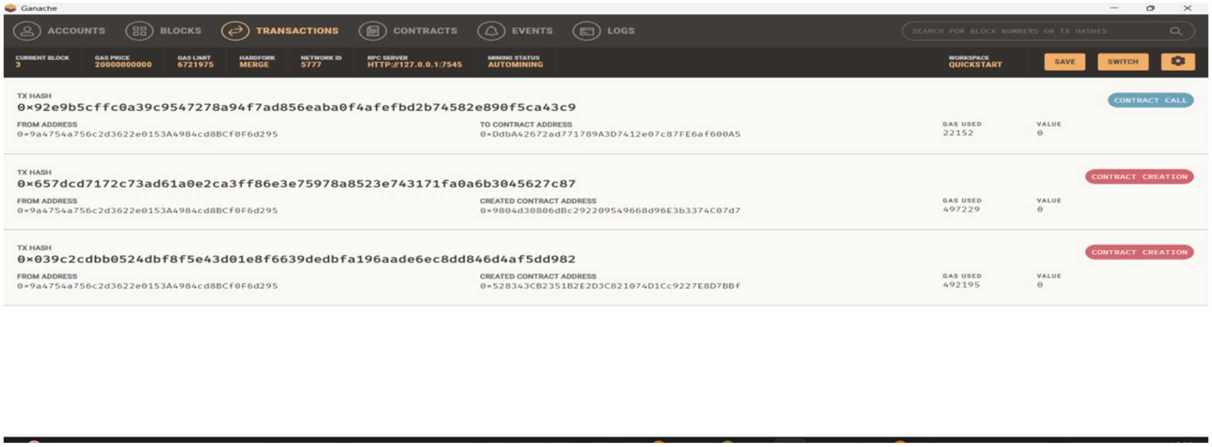


CURRENT BLOCK 3 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE QUICKSTART

BLOCK	MINED ON	GAS USED	
3	2025-07-10 15:52:09	22152	1 TRANSACTION
2	2025-07-10 15:20:19	497229	1 TRANSACTION
1	2025-07-10 15:20:19	492195	1 TRANSACTION
0	2025-07-10 14:49:34	0	NO TRANSACTIONS

Fig 6.2:Mined Blocks Showing EMR Transactions on Ganache

This shows mined blocks on the local Ethereum network using Ganache. Each block contains one or more transactions related to EMR uploads or access permissions. Gas usage and timestamps confirm successful blockchain activity.



CURRENT BLOCK 3 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE QUICKSTART

```
TX HASH                                                                                     CONTRACT CALL
0×92e9b5cffc0a39c9547278a94f7ad856eaba0f4afefbd2b74582e890f5ca43c9
FROM ADDRESS                                  TO CONTRACT ADDRESS                            GAS USED   VALUE
0×9a4754a756c2d3622e0153A4984cd8BCf0F6d295   0×DdbA42672ad771789A3D7412e07c87FE6af600A5     22152      0

TX HASH                                                                                     CONTRACT CREATION
0×657dcd7172c73ad61a0e2ca3ff86e3e75978a8523e743171fa0a6b3045627c87
FROM ADDRESS                                  CREATED CONTRACT ADDRESS                       GAS USED   VALUE
0×9a4754a756c2d3622e0153A4984cd8BCf0F6d295   0×9804d30806dBc292209549668d96E3b3374C07d7     497229     0

TX HASH                                                                                     CONTRACT CREATION
0×039c2cdbb0524dbf8f5e43d01e8f6639dedbfa196aade6ec8dd846d4af5dd982
FROM ADDRESS                                  CREATED CONTRACT ADDRESS                       GAS USED   VALUE
0×9a4754a756c2d3622e0153A4984cd8BCf0F6d295   0×528343CB2351B2E2D3C821074D1Cc9227E8D7BBf     492195     0
```

Fig 6.3: Smart Contract Transactions Logged in Ganache

This view logs all smart contract interactions, including contract creation and function calls for EMR access control. It confirms successful execution of blockchain-based operations with details like gas used, addresses, and transaction hashes.

VIII. CONCLUSION

This analysis presents a decentralized framework for managing electronic medical records using Ethereum blockchain and IPFS. By utilizing smart contracts for role-based access control and IPFS for scalable storage, the system addresses the major challenges of security, transparency, and patient data ownership. The architecture effectively eliminates centralized points of vulnerability and provides immutable logging of all interactions, which enhances accountability and compliance with health data regulations. The modular design also promotes extensibility and ease of integration with existing medical IT infrastructure. By offering real-time visibility, tamper-proof audit trails, and user-controlled access mechanisms, this framework empowers patients while maintaining operational efficiency for healthcare providers. Additionally, its reliance on widely adopted technologies like MetaMask and IPFS ensures user accessibility and cost-effectiveness.

In the future, the framework can be expanded to support AI-based anomaly detection for suspicious access patterns, edge computing to handle low-latency medical tasks near the data source, and advanced cryptographic techniques such as zero-knowledge proofs for further privacy assurance. These enhancements will further solidify the system's position as a next-generation standard for secure and decentralized EMR management.

REFERENCES

[1] Kuo, T. T., et al. (2017). Blockchain distributed ledger technologies for biomedical and health care applications. Journal of the American Medical Informatics Association.
[2] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
[3] Zheng, Z., et al. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. IEEE BigData Congress.
[4] Esposito, C., et al. (2018). Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy?. IEEE Cloud Computing.
[5] Omar, A. A., et al. (2019). MedChain: Efficient Healthcare Data Sharing via Blockchain. IEEE Access.
[6] Dubovitskaya, A., et al. (2017). Secure and Trustable Electronic Medical Records Sharing using Blockchain. AMIA Proceedings.
[7] Azaria, A., et al. (2016). MedRec: Using Blockchain for Medical Data Access and Permission Management. IEEE Open & Big Data.
[8] Liang, X., et al. (2017). Integrating blockchain for data security in EHR systems. IEEE SmartHealth.
[9] Benet, J. (2014). IPFS - Content Addressed, Versioned, P2P File System. arXiv:1407.3561.
[10] Grech, A., et al. (2020). IPFS and Blockchain Integration: Use Cases and Challenges. Journal of Web Engineering.
[11] Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. IEEE Access.
[12] Angraal, S., et al. (2017). Blockchain technology: applications in health care. Circulation: Cardiovascular Quality and Outcomes.
[13] Roehrs, A., et al. (2017). Personal health records: a systematic literature review. Journal of Medical Internet Research.
[14] F. J. Abdullayeva, "Internet of Things-based healthcare system on patient demographic data in health 4.0," *CAAI Trans. Intell. Technol.*, vol. 7, no. 4, pp. 644–657, 2022.
[15] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *Proc. 18th Int. Conf. E-Health Netw., Appl. Services (Healthcom)*, Munich, Germany, 2016, pp. 1–3.
[16] J. Chanchaichujit et al., "Blockchain technology in healthcare," in *Healthcare 4.0*. Singapore: Palgrave, 2019. [Online].
[17] F. Thalhammer et al., "Blockchain use cases against climate destruction," *Cloud Comput. Data Sci.*, vol. 3, no. 2, pp. 60–76, 2022.