IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511138

MediVault: A Decentralized RAG-Driven Framework for Secure and Interoperable Healthcare Data Management

Aiman Gulzar[1], Boomika Shetty[2], Deepashree N[3], Janaki K.B[4], Lavanya N.L[5]

Undergraduate Student, Department of CSE, EWCE, Bengaluru, India[1]

Undergraduate Student, Department of CSE, EWCE, Bengaluru, India[2]

Undergraduate Student, Department of CSE, EWCE, Bengaluru, India[3]

Assistant Professor, Department of CSE, EWCE, Bengaluru, India[4]

Head of Dept., Department of CSE, EWCE, Bengaluru, India[5]

Abstract: The increasing digitization of healthcare has created significant challenges in securely storing, managing, and sharing sensitive medical records. Centralized healthcare systems may expose patient information to unauthorized access and provide limited control over medical data. To address these challenges, this paper presents MediVault, a decentralized framework for secure healthcare data management using blockchain, InterPlanetary File System (IPFS), encryption, smart contracts, and a locally deployed Large Language Model (LLM). The proposed system enables patients to manage their medical records and control access granted to authorized healthcare providers. Medical records are encrypted before being stored in IPFS, while blockchain-based smart contracts support access authorization, revocation, and audit-related operations. A break-glass mechanism is incorporated to facilitate controlled emergency access. In addition, a local AI assistant using Ollama and Phi-3 Mini enables users to interact with available medical-record information without relying on external cloud-based AI services. The system is implemented using Streamlit and SQLite, providing a patient-centric approach to secure and privacy-aware healthcare data management.

Keywords: Healthcare Data Management, Blockchain, IPFS, Smart Contracts, Medical Data Security, Patient Privacy, Local LLM, Phi-3 Mini, Break-Glass Access.

I. INTRODUCTION

The rapid adoption of Electronic Health Records (EHRs), telemedicine, and digital healthcare applications has transformed the way medical information is stored, accessed, and managed. Healthcare records contain highly sensitive information, including medical histories, laboratory reports, prescriptions, diagnostic reports, and treatment information. As the volume of digital healthcare information continues to increase, ensuring the confidentiality, integrity, controlled access, and secure management of such information has become an important requirement. [3], [11]

Traditional healthcare information systems commonly rely on centralized databases maintained by hospitals, healthcare providers, or cloud platforms. Although centralized architectures provide convenient access and management of medical records, they can introduce security and privacy concerns because sensitive information is maintained under centralized control. Healthcare records may also be distributed across different healthcare organizations, making it difficult to provide patients and authorized healthcare professionals with consistent access to complete medical information. These challenges emphasize the need for healthcare systems that provide secure data management while giving patients greater control over their medical information. [3], [11]

Blockchain technology provides a decentralized and tamper-resistant mechanism for managing transactions and access-related information. The concept of decentralized transaction management was introduced through blockchain-based systems such as Bitcoin [1], while Ethereum provides a programmable blockchain environment that supports smart contracts [2]. Previous healthcare research has explored blockchain, smart contracts, and IPFS for healthcare data management, demonstrating their potential for improving data integrity, privacy, and decentralized access [3]. However, storing complete medical files directly on a blockchain is impractical because healthcare records can contain large documents and diagnostic files. The InterPlanetary File System (IPFS) provides a decentralized, content-addressed approach for storing such files while allowing their corresponding identifiers to be used for retrieval [4].

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511138

Artificial Intelligence and Large Language Models (LLMs) provide additional possibilities for interacting with healthcare information. Retrieval-Augmented Generation (RAG) combines information retrieval with language generation to provide responses based on retrieved information [7]. Recent research has investigated RAG for healthcare question answering, medical information retrieval, and electronic health-record applications [9]. However, healthcare applications involving sensitive patient information also require consideration of privacy and the risks associated with sending medical data to external AI services. Existing research has also identified challenges such as retrieval noise, latency, explainability, and domain adaptation in healthcare RAG systems [9], [10].

To address these challenges, MediVault is developed as a decentralized healthcare data management system that combines encryption, IPFS, blockchain-based access control, smart contracts, and local AI assistance. The system enables patients to manage their medical records and control access for authorized healthcare providers. Medical records are encrypted before being stored through IPFS, while blockchain and smart contracts support authorization, access control, and audit-related operations. A controlled break-glass mechanism is also incorporated to support emergency access to medical information under predefined conditions.

The system additionally incorporates a locally deployed Large Language Model using Ollama and Phi-3 Mini, allowing users to interact with available medical-record information without depending on an external cloud-based AI service. MediVault is implemented as a Streamlit-based web application, with SQLite used for application-level data management. The combination of encrypted decentralized storage, blockchain-based authorization, emergency access control, and local AI assistance provides a patient-centric approach to healthcare data management.

The current implementation focuses on secure medical-record management, decentralized storage, controlled access, and local AI-assisted interaction. HL7 FHIR interoperability is not implemented in the current version and is considered a future enhancement for enabling standardized exchange of healthcare information.

II. RELATED WORK

Several studies have investigated the use of blockchain, decentralized storage, artificial intelligence, and retrieval-based techniques for improving healthcare data management.

Panduranga Vital Terlapu et al. (2025) proposed a blockchain-based healthcare data management system that combines blockchain, smart contracts, and the Inter Planetary File System (IPFS) for secure healthcare data management. The approach uses blockchain to maintain decentralized transaction information while IPFS is used for storing healthcare data. This work demonstrates the potential of combining blockchain and decentralized storage for healthcare applications. However, challenges related to scalability, interoperability, and integration with existing healthcare systems remain. [3]

Radha Indoriya (2025) presented an AI-driven blockchain framework for secure healthcare data interchange. The framework combines blockchain-based security mechanisms with artificial intelligence techniques to improve healthcare data management and security. The study demonstrates how AI and blockchain can be integrated within healthcare environments, although such combined architectures can introduce additional computational and implementation complexity. [6]

Recent research has also explored Retrieval-Augmented Generation (RAG) for healthcare applications.

Fnu Neha, Deepshikha Bhati, and Deepak Kumar Shukla (2025) presented a comprehensive review of RAG in healthcare, discussing its applications in medical question answering, information retrieval, electronic health records, and clinical decision-support scenarios. The study also identifies challenges including retrieval noise, latency, explainability, and domain adaptation. [9]

Julian Garcia (2025) proposed DF-RAG, a dual federated retrieval-augmented generation framework for collaborative medical AI. The framework combines federated approaches with retrieval-augmented generation to support privacy-preserving collaborative healthcare AI. Although such approaches can reduce the need to centralize sensitive information, they may require significant computational and communication infrastructure. [10]

Adline Freeda R., Anu Abarna B., Jahnavi K., and Lavanya P. (2022) proposed an electronic medical record management system using a recommendation algorithm. The system focuses on managing electronic medical records and providing recommendation-based functionality. However, the centralized nature of conventional EMR architectures can create challenges related to patient-controlled access and decentralized data management. [11]

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511138

Based on these studies, existing research has addressed individual aspects of healthcare data security, decentralized storage, blockchain-based authorization, and AI-assisted information retrieval. However, there remains a need for an integrated implementation that combines encrypted medical-record storage, decentralized IPFS storage, blockchain-based access control, controlled emergency access, and local AI assistance within a patient-centric healthcare application.

MediVault addresses these implementation requirements by integrating these components into a single healthcare data management framework. The current implementation focuses on secure medical-record management, access authorization, decentralized storage, and local AI interaction. HL7 FHIR interoperability is not implemented in the current version and is considered future scope.

Table I. Comparison of Existing Healthcare Data Management Approaches

Study	Main Technologies	Primary Focus	Identified Limitation
Panduranga Vital Terlapu et al [3]	Blockchain, Smart Contracts, IPFS	Secure healthcare data management	Scalability and interoperability challenges
Radha Indoriya (2025) [6]	AI, Blockchain, Smart Contracts	Secure healthcare data interchange	Implementation and computational complexity
Fnu Neha, Deepshikha Bhati, and Deepak Kumar Shukla (2025) [9]	RAG, AI, Healthcare Data	Healthcare information retrieval and generation	Retrieval noise, latency, explainability
Julian Garcia (2025) [10]	Federated Learning, RAG, Knowledge Graphs	Collaborative medical AI	Infrastructure and computational requirements
Adline Freeda R., Anu Abarna B., Jahnavi K., and Lavanya P. (2022) [11]	EMR, Recommendation Algorithm	Electronic medical record management	Limited decentralized and patient-controlled access
Medivault	Blockchain, IPFS, Encryption, Smart Contracts, Local LLM	Patient-centric healthcare data management	FHIR interoperability is future scope

III. PROPOSED SYSTEM / METHODOLOGY

This section explains the design and working methodology of MediVault. The system is designed to provide secure medical-record management by combining encryption, decentralized storage, blockchain-based access control, smart contracts, and local AI assistance.

A. System Overview

MediVault is a web-based healthcare data management system that allows patients to store and manage their medical records while controlling access to authorized healthcare providers. The system consists of three primary users: Patient, Doctor, and Administrator.

When a patient uploads a medical record, the record is encrypted before being stored through IPFS. The resulting IPFS Content Identifier (CID) is associated with the corresponding medical record. Blockchain-based smart contracts are used to manage access permissions between patients and doctors. Patients can grant or revoke access to their records, while authorized doctors can retrieve records according to the permissions provided.

The system also incorporates a break-glass mechanism for controlled emergency access. In addition, MediVault provides a locally deployed AI assistant using Ollama and Phi-3 Mini, allowing interaction with available medical-record information without sending the records to an external cloud-based AI service.

B. System Architecture

The overall architecture of MediVault consists of the user interface layer, application layer, security layer, decentralized storage layer, blockchain layer, and local AI layer.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511138

METHODOLOGY

SYSTEM ARCHITECTURE

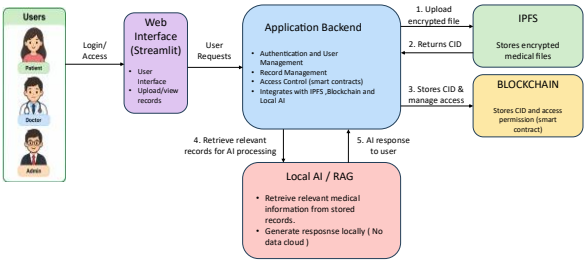


Fig. 1. System Architecture of MediVault

The patient, doctor, and administrator interact with the system through the Streamlit-based web interface. The application layer handles authentication, medical-record operations, access requests, and user interactions. Medical records are encrypted before being transferred to decentralized storage.

IPFS provides storage for the encrypted medical files, while the blockchain and smart contract layer manages access-control information and related transactions. The local AI layer uses Ollama with Phi-3 Mini to provide AI-assisted interaction with available medical-record information.

C. User Roles and Access Control

MediVault provides role-based functionality for three types of users:

1. Patient

The patient is the primary owner of the medical information. The patient can:

- Register and authenticate with the system.
- Upload medical records.
- View stored medical records.
- Manage access permissions.
- Grant access to authorized doctors.
- Revoke previously granted access.
- Interact with the local AI assistant using available medical information.

2. Doctor

The doctor can access patient records only when appropriate authorization has been provided. The doctor can:

- Register and authenticate.
- Request access to patient records.
- View records for which access has been granted.
- Use available medical information through the system's AI functionality.
- Participate in the controlled emergency-access process where applicable.

3. Administrator

The administrator provides system-level management functions. These include managing registered users and supporting administrative operations required by the healthcare application.

This role-based structure separates the responsibilities of patients, doctors, and administrators and supports controlled access to medical information.

D. Medical Record Encryption and IPFS Storage

Security of medical records is an important component of MediVault. Before a medical record is stored through IPFS, the file is encrypted using Fernet symmetric encryption.

The basic process is:

Medical Record → Encryption → IPFS Storage → CID Generation

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511138

When a patient uploads a medical document, the application encrypts the file before sending it to the IPFS storage layer. IPFS then generates a Content Identifier (CID) for the stored content.
The CID is used as an identifier for locating the encrypted record rather than storing the complete medical document directly on the blockchain. This approach separates large medical files from blockchain transaction information.
The encrypted file can subsequently be retrieved using its associated CID and decrypted when an authorized user accesses the record.

E. Blockchain and Smart Contract-Based Access Control

MediVault uses blockchain technology to support decentralized access-control operations. A Solidity-based smart contract, MediVaultAccess, is used to manage important access-related operations.
The smart contract supports functionality such as:

- Patient registration
- Doctor registration
- Doctor approval
- Granting patient-to-doctor access
- Revoking access
- Maintaining IPFS-related record information
- Recording blockchain events
- Supporting emergency-access operations
-

Instead of relying entirely on a centralized access-control mechanism, smart contracts provide programmable rules for managing permissions.

The patient remains responsible for deciding which authorized doctor can access the patient's medical information.

F. Break-Glass Emergency Access

Healthcare systems may require access to medical information during emergency situations where the patient is unable to provide normal authorization.

To address this requirement, MediVault incorporates a break-glass emergency access mechanism. Under the controlled emergency process, emergency access requires participation from the designated authorized parties rather than providing unrestricted access to medical records.

The mechanism is designed to provide access under predefined emergency conditions while maintaining an auditable access process.

This functionality provides an additional access path for emergency situations without making normal patient-controlled authorization unrestricted.

G. Local LLM-Based Medical Record Assistant

MediVault incorporates a locally deployed Large Language Model to allow users to interact with available medical-record information.
The implementation uses:

- Ollama for local model execution
- Phi-3 Mini as the local language model

The local deployment is intended to reduce dependence on external cloud-based AI services when processing sensitive healthcare information.
The AI assistant is designed to respond using the medical-record information available to it rather than generating unsupported medical information. This provides a mechanism for users to interact with their stored information through natural-language queries while keeping the AI processing within the local environment.

It is important to distinguish this implementation from a conventional RAG pipeline: the current MediVault implementation should be described as a local LLM-based medical-record assistant, not as a fully implemented RAG system.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511138

H. Overall System Workflow

The overall workflow of MediVault can be summarized as follows:

- Step 1: The user registers and authenticates with the system.
- Step 2: A patient uploads a medical record through the application.
- Step 3: The medical record is encrypted using Fernet encryption.
- Step 4: The encrypted record is stored through IPFS.
- Step 5: IPFS generates a CID associated with the stored record.
- Step 6: Relevant access and record information is managed through the blockchain and smart contract layer.
- Step 7: Patients give access to their medical record.
- Step 8: The patient can grant or revoke access according to the required authorization.
- Step 9: Once authorized, the doctor can retrieve the permitted medical record.
- Step 10: In emergency situations, the break-glass mechanism provides controlled emergency access according to the defined authorization process.
- Step 11: The local Phi-3 Mini model can be used to interact with available medical-record information without relying on an external cloud AI service.

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

IV. SYSTEM IMPLEMENTATION

The MediVault system is implemented as a web-based healthcare data management application that combines secure medical-record handling, decentralized storage, blockchain-based access control, and local AI assistance. The implementation consists of the application interface, database, encryption module, IPFS storage, smart-contract layer, and local LLM component.

A. Application Interface

MediVault provides a web-based interface developed using Streamlit. The application provides separate functionality for three user roles: Patient, Doctor, and Administrator.

Users authenticate through the application before accessing role-specific functions. Patients can manage their medical records and access permissions, while doctors can request and access records according to the authorization provided. Administrative functions support user and emergency-access management.

B. Database Management

SQLite is used for application-level data management. The database maintains information required by the application, such as user information, medical-record details, access-related information, and system logs.

The database is used for application operations, while the actual encrypted medical files are handled separately through IPFS.

C. Medical Record Encryption

Before a medical record is stored, the system applies Fernet symmetric encryption to the file.
The encryption process protects the medical document before it is transferred to decentralized storage. During an authorized retrieval operation, the encrypted file can be retrieved and decrypted for viewing.
The basic process is:
Medical File → Fernet Encryption → Encrypted File → IPFS

This ensures that the file stored in decentralized storage is not maintained as an unencrypted medical document.

D. IPFS-Based Storage

MediVault uses the InterPlanetary File System (IPFS) for storing encrypted medical records.
After an encrypted file is uploaded to IPFS, the system obtains a Content Identifier (CID). The CID acts as the reference used to locate the stored file.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511138

Instead of storing the complete medical document on the blockchain, the system uses the CID as the reference for the corresponding file. This separates medical-file storage from blockchain-based access-control information.

A local fallback mechanism is also available when the IPFS service is not reachable during development or testing.

E. Blockchain and Smart Contract Implementation

The blockchain component of MediVault uses a Solidity smart contract named MediVaultAccess.

The smart contract provides functions related to:

- Patient registration
- Doctor registration
- Doctor approval
- Granting access
- Revoking access
- Authorization checking
- Emergency access
- Recording relevant blockchain events

The smart-contract layer provides programmable access-control rules for managing permissions between patients and healthcare providers.

F. Local AI Assistant

MediVault incorporates a locally executed AI assistant using Ollama and Phi-3 Mini.

The local model allows users to interact with available medical-record information through natural-language queries. Since the model is executed locally, the system does not require sending the medical-record content to an external cloud-based AI API for this functionality.

The assistant is intended to work with the available record information and avoid generating information that is not present in the provided medical context.

G. Emergency Access Implementation

The system includes a break-glass emergency access mechanism for situations where normal patient authorization may not be available.

The emergency process provides a controlled method for requesting and authorizing access to medical information. The corresponding access operation is handled through the smart-contract access-control mechanism, allowing the emergency operation to be recorded as part of the system's audit process.

V. RESULTS AND DISCUSSION

The implemented MediVault system demonstrates the integration of secure medical-record management, decentralized storage, blockchain-based access control, and local AI assistance within a single application. The system was developed and tested through its major functional components, including user authentication, medical-record handling, encryption, IPFS storage, access authorization, and local AI interaction.

A. Functional Testing

The major functional operations considered during system testing are shown in Table II

Table II. Functional Testing of MediVault

TC ID	Feature	Test Case Description	Input	Expected Output	Actual Output	Result
01	Patient Registration	Verify the patient registration	Username, email, password, blood group, allergies.	Account created with an ID	Account created with ID MED-PAT-0001	Pass

DOI: 10.17148/IJARCCE.2026.1511138

02	User Login	Check if the user can log into the portal	Username and password	Successfully authenticates and opens portal	Successfully authenticated; opens Patient Portal	Pass
03	Medical Record Upload	Verify if the records can be uploaded	Record title, category, doctor name, findings	Record encrypted and IPFS CID generated & displayed	Record encrypted and IPFS CID generated & displayed	Pass
04	IPFS Retrieval	Verify if the files can be accessed	Doctor queries AI about patient history	The record is retrieved	The record is retrieved	Pass
05	Blockchain Audit Trail	Check if Blockchain works	-	Every action performed (request, approval, read, revoke)	Every action performed (request, approval, read, revoke)	Pass
06	Doctor Access Control	Verify if the doctor can access the files	Patient grants access to the doctor	Smart contract updates has Access[doctor Id] to true; Doctor gains view permissions	Smart contract updates has Access[doctor Id] to true; Doctor gains view permissions	Pass
07	AI Query	Verify if local AI retrieves the results.	What are the patient's allergies?"	AI extracts and answers accurate allergy info from patient record	The AI model is small, so the	Partially Passed

B. Security and Access-Control Discussion

The implementation combines multiple security mechanisms at different stages of medical-record management. Encryption is applied before decentralized storage, while the IPFS layer handles the encrypted file. The blockchain and smart-contract layer manages access-related operations.

The separation between medical-file storage and access-control information reduces the need to store complete medical documents on the blockchain. The use of smart contracts also provides a programmable mechanism for managing authorization and revocation.

The break-glass mechanism provides an additional controlled access path for emergency situations. This allows emergency access to be handled separately from the normal patient-controlled authorization process.

C. Local AI Assistant

The local AI component provides natural-language interaction with available medical-record information using Ollama and Phi-3 Mini.

The local execution approach avoids requiring the medical-record information to be sent to an external cloud AI service for the implemented assistant functionality.The current implementation should not be evaluated as a conventional RAG system because a complete retrieval-augmented generation pipeline is not implemented in the current version. RAG remains relevant to the research background and can be considered for future enhancement.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511138

D. System Limitations

Although the current implementation integrates several components into one healthcare application, some limitations remain.

- HL7 FHIR interoperability is not implemented in the current version.
- The current AI component is a local LLM assistant, rather than a complete RAG pipeline.
- The system has been developed and evaluated as a project prototype rather than through deployment across multiple hospitals or large-scale clinical environments.
- Large-scale performance, scalability, and cross-hospital interoperability require further evaluation.

These limitations provide directions for future development rather than being presented as implemented features.

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This paper presented MediVault, a decentralized framework for secure and patient-centric healthcare data management. The system combines encryption, InterPlanetary File System (IPFS), blockchain-based access control, smart contracts, and a locally deployed Large Language Model to address important challenges associated with the storage, sharing, and controlled access of medical records.

The implemented system enables patients to manage their medical records and control access permissions for authorized healthcare providers. Medical records are encrypted before storage, while blockchain-based smart contracts support access authorization and revocation. The system also incorporates a controlled break-glass mechanism for emergency access. In addition, the integration of Ollama with Phi-3 Mini provides local AI-assisted interaction with available medical-record information without requiring an external cloud-based AI service.

Functional testing of the implemented application demonstrated the operation of major components including patient authentication, medical-record management, access granting and revocation, and authorized doctor access. The current implementation therefore demonstrates the feasibility of combining decentralized storage, programmable access control, and local AI assistance within a healthcare data management application.

B. Future Scope

The current MediVault implementation can be extended in several directions to improve usability, interoperability, scalability, and healthcare intelligence.

- Mobile Application: Dedicated Android and iOS applications can be developed to support features such as camera-based prescription uploads, instant access notifications, and QR-based patient identification.
- Real Blockchain Deployment: The current blockchain environment can be extended from the development setup to a real blockchain network such as Polygon or Sepolia, with wallet integration for practical on-chain consent and access management.
- HL7 FHIR Interoperability: HL7 FHIR standards can be incorporated to enable standardized exchange of medical information between MediVault and other hospital or Electronic Medical Record (EMR) systems.
- Upgraded Local LLM: The current Phi-3 Mini model can be replaced or enhanced with a more capable locally deployed model such as Llama or Mistral, potentially connected to a specialized medical knowledge base to improve healthcare information interaction.
- Multilingual Support: Support for regional languages such as Hindi, Kannada, and Tamil can be introduced to improve accessibility for users from diverse linguistic backgrounds, particularly in rural and underserved communities.
- Enhanced Break-Glass Access: The existing emergency-access mechanism can be further enhanced using multi-signature authorization involving a doctor and administrator, time-limited emergency permissions, immutable audit records, and automated patient notifications.

REFERENCES

[1]. S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” SSRN Electronic Journal, Aug. 2019.
[2]. G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Project Yellow Paper,” Apr. 2014.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511138

[3]. P. V. Terlapu, R. Salakapuri, V. Karthikeya, P. Srilakshmi, M. Likhitha, and K. V. Vardhan, "Health Data Management System Using Blockchain Technology," 2025 International Conference on Next Generation of Green Information and Emerging Technologies (GIET), pp. 1–6, Aug. 2025.
[4]. J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," arXiv, Jul. 2014.
[5]. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," EMNLP-IJCNLP, pp. 3982–3992, Nov. 2019.
[6]. E. Radha Indoriya, M. Chawla, and N. Kumar, "AI-Driven Blockchain Framework for Secure Healthcare Data Interchange," EasyChair Preprint no. 15884, Mar. 2025.
[7]. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv, May 2020.
[8]. A. Vaswani et al., "Attention Is All You Need," 2017.
[9]. F. Neha, D. Bhati, and D. K. Shukla, "Retrieval-Augmented Generation (RAG) in Healthcare: A Comprehensive Review," AI, vol. 6, no. 9, p. 226, Sep. 2025.
[10]. J. Garcia, J. Gong, M. Zajac, and A. Hahn, "DF-RAG: A Dual Federated Retrieval-Augmented Generation Framework for Collaborative Medical AI," ACM/IEEE International Conference on Connected Health, pp. 418–423, Nov. 2025.
[11]. A. Abarna, J. Jahnavi, and L. Lavanya, "Electronic Medical Record Management System Using Recommendation Algorithm," ICICICT, pp. 193–199, Aug. 2022.