Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 3, June 2025, pp. 2131~2142
ISSN: 2302-9285, DOI: 10.11591/eei.v14i3.9355

Event-driven integration of electronic medical records with blockchain and InterPlanetary file system

Cahyo Arissabarno[1], Sritrusta Sukaridhoto[1], Idris Winarno[1], Rizqi Putri Nourma Budiarti[2]
[1]Human Centric Multimedia Research Laboratory, Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia
[2]Department of Information System, Universitas Nahdlatul Ulama Surabaya, Surabaya, Indonesia

Article Info

Article history:

Received Sep 24, 2024
Revised Nov 22, 2024
Accepted Dec 25, 2024

Keywords:

Blockchain
Change data capture
Electronic medical record
Hyperledger
InterPlanetary file system

ABSTRACT

The integrity, security, and accessibility of electronic medical record (EMR) are often compromised by traditional systems, which struggle to ensure data integrity, transparent audit trails, and secure long-term storage. This research addresses these challenges by integrating EMR with a private blockchain and InterPlanetary file system (IPFS) cluster, using change data capture (CDC) for real-time updates and integrate with existing EMR systems, avoiding the need for building new EMR software. Implemented in the OpenEMR framework, the system's performance is evaluated across various processes, including document uploading, sharing, access, deletion, and integrity verification. Testing with anonymized medical records in PDF formats ranging from 1 MB to 100 MB shows that uploading to IPFS takes 0.7 seconds per MB, blockchain transaction processing averages 4.2 seconds, CDC time is 1.1 seconds per MB, and OpenEMR uploads average 0.98 seconds per MB. These results demonstrate significant improvements in data security, integrity, and availability, following the CIA triad principles. The system provides a traceable and secure solution for EMR management.

Corresponding Author:

Sritrusta Sukaridhoto
Human Centric Multimedia Research Laboratory, Department of Informatics and Computer Engineering
Politeknik Elektronika Negeri Surabaya
PS10.02, Raya ITS Road, Keputih, Sukolilo, Surabaya, East Java, Indonesia
Email: dhoto@pens.ac.id

1. INTRODUCTION

An electronic medical record (EMR) is a digital version of a patient's conventional paper chart, containing a wide range of information including medical history, diagnoses, prescribed medications, treatment plans, immunization records, known allergies, radiological images, and laboratory test results [1]. EMR is a clinical data storage system that allows for computerized data entry, standardizes terminology, organizes medical and pharmaceutical documentation, and supports clinical judgments within healthcare settings [2]. EMR systems enhance medical care by reducing paperwork, enhancing efficiency, and improving healthcare facilities, but their widespread adoption is hindered by integrity and traceability concerns [3], [4]. Medical records are considered as confidential documents within the realm of sensitive information. In the year 2023, there were 395 cases of medical record data breaches, impacting a total of 59,569,604 individuals. The current hospital information systems are still centralized and lack enhanced security measures to safeguard the confidentiality of medical records for various patients. Administrators of the hospital information systems also have the opportunity to manipulate or leak patient records without detection or documentation. Derek Lewis, an EMR specialist at the largest hospital in the United States, faces serious risks due to weaknesses in EMR software. These issues include errors in tracking patient medical records that list prescriptions and medication dosages. This has the potential to endanger patient safety.

The combination of InterPlanetary file system (IPFS) and blockchain technology can address the challenges of EMR system integrity and traceability. IPFS is a decentralized file storage system that allows users to store files more securely, using unique cryptographic-based addresses and identifiers (hashes) for access [5]. Blockchain is a distributed data storage system that uses a peer-to-peer network secured with cryptography, making it highly resistant to manipulation and employing consensus mechanisms to store data [6]. Blockchain is highly resistant to tampering and employing consensus mechanisms for data addition in traceable way [7]. Each piece of data stored on the blockchain is interconnected through cryptography, making it well-suited for systems requiring traceability and security [8], [9], such as internet of thing (IoT) [10] and supply chain [11]. Blockchain is not designed to store files as it is intended only for storing text-based data. To overcome this limitation, IPFS can be combined with blockchain, as it shares the same distributed concept. IPFS stores files in a decentralized manner and provides a unique hash for each file, which can then be stored on the blockchain as a reference. This combination enables efficient file storage while ensuring data integrity and traceability through the blockchain.

Several previous research have implemented the integration of blockchain and IPFS with EMR software. Research by Satrio *et al.* [12] integrates blockchain with an open-source hospital system using Hyperledger Fabric and Kafka, but focuses on data security, excluding patient medical files. Research by Hovorushchenko *et al.* [13] presents a medical data transaction methodology on the blockchain, but it hasn't been tested on real servers or secured file storage. Research by Shao *et al.* [14] proposes a blockchain and IPFS-based EMR system using pairing-based cryptography, but it's limited to trials and lacks direct integration with existing EMR systems.Pakkala *et al.* [15] develops a dApp for medical data on IPFS, but it's not integrated with hospital systems or ready for production. Mohsan *et al.* [16] introduces a conceptual framework using Ethereum and IPFS for decentralized medical metadata, but relies on a test blockchain network (TESTNET).

Table 1 shows that previous studies haven't fully addressed issues with the integrity, traceability, and real-time integration of EMR files. This research proposes a solution using an event-driven approach with a private blockchain and IPFS cluster. The system uses change data capture (CDC) to provide real-time updates and integrate with existing EMR systems, avoiding the need for building new EMR software. This method allows for immediate, secure updates of altered data [17]. The use of a private blockchain and IPFS Cluster improves the integrity and traceability of medical records. IPFS cluster offers secure, distributed file storage, while the blockchain manages file ownership and access record. This approach maintains document integrity and traceability, avoids the costs and disruptions of new EMR system development, and enhances overall data security and efficiency.

Table 1. Comparison with previous works

Feature	Satrio *et al.* [12]	Shao *et al.* [14]	Pakkala *et al.* [15]	Mohsan *et al.* [16]	Proposed
Blockchain type	Private (Hyperledger Fabric)	Public blockchain	Public blockchain	Ethereum (Testnet)	Private (Hyperledger Besu)
IPFS integration	No	Yes	Yes	Yes	Yes
File integrity	No	Yes	Yes	Yes	Yes
Integration with Existing EMR	Partial	No	No	No	Yes
Smart contract use	Yes	Yes	Yes	Yes	Yes
Data security focus	EMR input data	EMR input data	Medical file	Reports and images	Medical record document

This paper is organized into several sections: section 1, we review prior research on blockchain applications in medical records. Section 2, we describe the proposed architecture. Section 3, we present the implementation and analytical results. Section 4, we discuss our conclusions.

2. METHOD

This section details the tools and methods used in this research to integrate OpenEMR with blockchain and IPFS. As shown in Figure 1, key components include the OpenEMR system for managing medical records, CDC for real-time data updates, a private blockchain for secure record-keeping, and IPFS for decentralized storage. The following subsections explain the implementation and purpose of each component in the overall system.

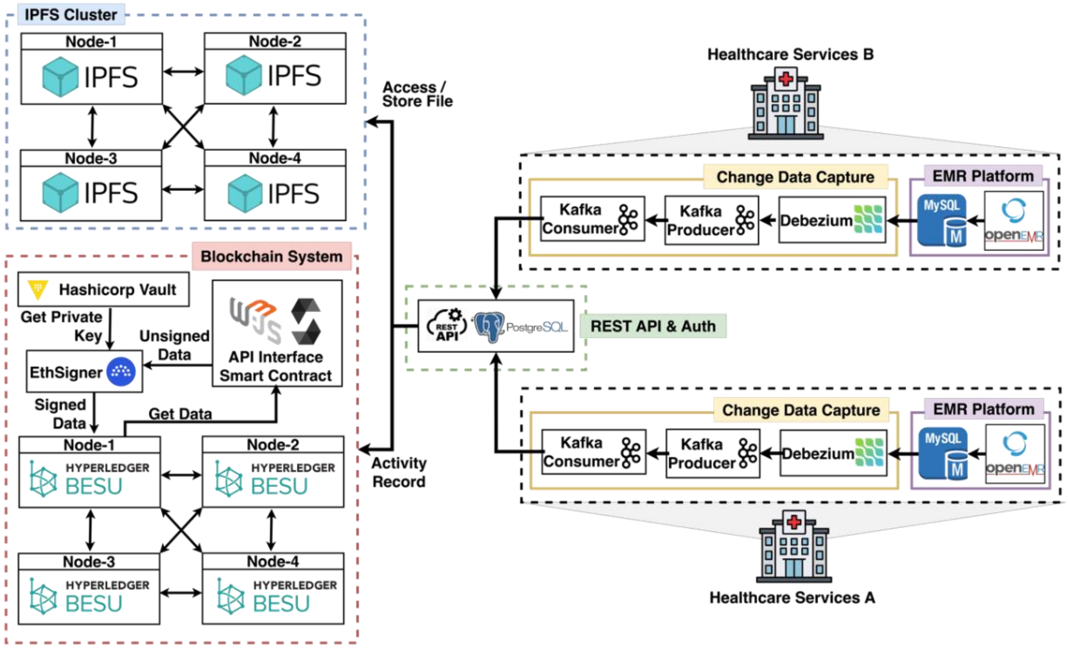

Figure 1. Proposed system diagram

2.1. OpenEMR

In this research, an EMR system was implemented using OpenEMR, a web-based, open-source medical information system platform. OpenEMR is built with PHP, and MariaDB serves as the underlying database, providing a robust and scalable solution for managing medical records. OpenEMR offers a comprehensive API that facilitates seamless integration with other systems. The software is designed with two primary platforms: one for administrators and one for patients, ensuring that both user groups have access to the tools and information they need. Additionally, the platform supports the storage and retrieval of EMR, management of patient prescriptions, handling of medical billing and insurance claims, and the generation of various medical and administrative reports. The patient platform, on the other hand, provides access to personal health records, allowing patients to view their medical history and records. This dual-platform approach ensures that both administrative staff and patients have tailored access to the functionalities they require. In this study, OpenEMR was installed on a virtual machine (VM). In this research, CDC process, aimed at capturing every MariaDB database activity within OpenEMR utilizing the Debezium connector. Each activity captured by the connector is broadcasted through Kafka, distributed across multiple topics. To enable the REST API to detect and relay each activity to IPFS and blockchain, a Kafka consumer is essential. The Kafka consumer triggers the REST API endpoint, starting the necessary processes to IPFS and blockchain.

2.2. Change data capture

CDC is a methodology designed to identify, monitor, and capture changes in real-time within a source database [18]. It tracks data modification operations such as inserts, updates, and deletes, recording these changes in the order they occur [19]. This ensures that downstream systems receive real-time updates, which is crucial for maintaining data integrity and consistency. CDC facilitates near-real-time extract, transform, load (ETL) processes for data warehouses by leveraging transaction logs. These logs provide a detailed account of every change made within the database, allowing CDC to propagate changes efficiently and synchronize multiple databases with minimal latency. Figure 2 visualize the flow of CDC method.

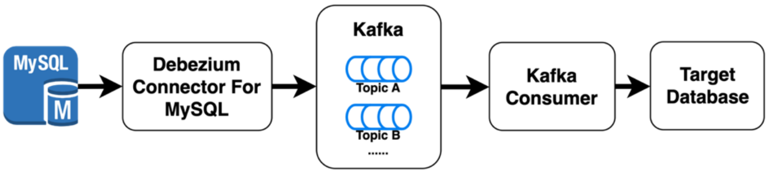

Figure 2. CDC flow diagram

2.2.1. Debezium connector

This service enables applications to effectively monitor row-level changes in databases. It captures and records every committed row-level modification for each table in a transaction log, allowing applications to observe and react to these changes in real time [20]. For SQL-based databases, it relies on a binary log (binlog) to document all operations in the precise order in which they are committed to the database. This log not only includes data modifications within tables but also captures schema alterations. The binlog plays a crucial role in the replication and recovery processes of databases.

2.2.2. Apache Kafka

Apache Kafka serves as a distributed platform for event streaming that serves as both a publish-subscribe message broker and a distributed logging tool [21], [22]. Kafka producers create messages and distribute them to designated topics, whereas consumers enroll in these topics to receive and process the messages [23]. Consumers in Kafka play a crucial role in fetching and processing these messages.

2.3. Private blockchain

Private blockchains are networks that are restricted to specific organizations, allowing them to maintain a decentralized ledger with limited access [24]. These exclusive blockchains facilitate collaboration among designated organizations and ensure that the decentralized ledger is only accessible to authorized participants. This setup guarantees the secure processing, sharing, and management of sensitive data [25].

This research uses Hyperledger Besu, an open-source Ethereum client, to build a private blockchain. It employs the IBFT2.0 consensus mechanism, a type of proof of authority (PoA). Each authorized node necessitates an address along with a pair of keys, comprising both a public key and a private key, for its configuration, which Hyperledger Besu simplifies by generating a consensus configuration file. A configuration file is essential for each node to function effectively. Figure 3 illustrates the workflow for adding new nodes and processing transactions, where new nodes must be approved before joining. Validator nodes vet transactions, ensuring that only valid ones are added to the blockchain, thus maintaining network security and transaction integrity.

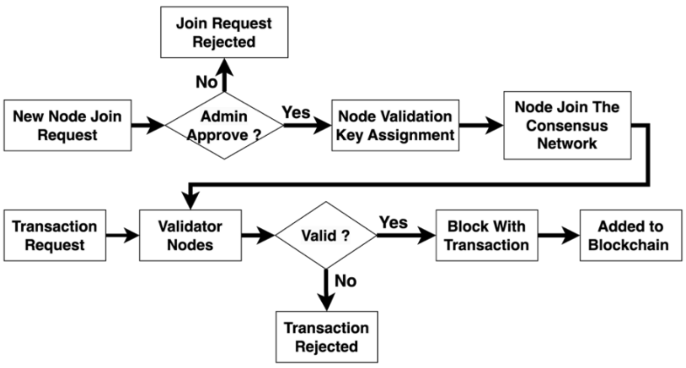

Figure 3. Proof-of-authority consensus diagram

2.4. Smart contract

A smart contract is a computer program developed as a script that is pinned to a blockchain used to execute predefined actions [26]. This study implements a smart contract using the Solidity programming language, deployed on the blockchain with the truffle framework. The algorithm for the smart contract is detailed in Algorithm 1 and is designed to handle document management through several essential variables. For instance, cidAddress signifies the address derived from the content identifier (CID) used for file indexing, and cid refers to the content ID associated with the file. Other important variables include nik, which represents the patient's National Identification Number; from, indicating the address of the transaction sender; category, which categorizes the document (e.g., medical record or prescription); and status, reflecting the current state of the document (e.g., owner or shared). Additionally, the smart contract monitors various actions performed on the document, such as uploading, viewing, sharing, deleting, or downloading. It also captures supplementary details about the document, including its name, size, type, and the identity of the individual who shared it. Event *dataDocumentEMR()* is emitted whenever *setDocument()* is executed. This event ensures that all actions related to the document are meticulously recorded and remain traceable within the system.

Algorithm 1. Smart contract: document EMR

```
Data: Initialize
   cid address: address indexed,
   cid: string,
   nik: uint indexed,
   from: address indexed,
   category: string,
   status: string,
   action: string indexed,
   details: string,
   did: uint,
   date: uint
Result: Updated cid address, cid, nik, from, category, status, action, details,
did, date
Event dataDocumentEMR (cid address: address, cid: string, nik: uint, from:
address, category: string, status: string, action: string, details: string, did:
uint, date: uint)
Function setDocument():
   Set cid address;
   Set cid;
   Set nik;
   Set from;
   Set category;
   Set status;
   Set action;
   Set details;
   Set did;
   Set date;
   Emit dataDocumentEMR event with (cid address, cid, nik, from, category, status,
action, details, did, date);
```

The algorithm operates through a series of steps where the change detection and synchronization mechanisms are formally defined. Let D represent the document being managed and C_{new} denote any detected changes. The change detection function $f(C_{new})$ identifies these modifications and is mathematically expressed as in (1). Following the detection of changes, they are synchronized with IPFS using the function g, which can be described as in (2). The synchronization time, denoted T_{sync}, is modeled as in (3). Subsequent to synchronization, changes are recorded on the blockchain via the transaction function T, as in (4). The time for this transaction, $T_{transaction}$, includes as in (5).

$$f(C_{new}) = detect\_changes(D) \tag{1}$$

$$g(D, C_{new}) = sync\_to\_ipfs(D, C_{new}) \tag{2}$$

$$T_{sync} = T_{upload\_ipfs} + T_{proccessing} \tag{3}$$

$$T(D, C_{new}) = record\_transaction(D, C_{new}) \tag{4}$$

$$T_{transaction} = T_{hashing} + T_{recording} \tag{5}$$

2.5. InterPlanetary file system cluster

The IPFS is a decentralized protocol that facilitates peer-to-peer data storage, with the objective of improving both security and reliability through the distribution of data across a network. This methodology mitigates the vulnerabilities inherent in centralized systems, thereby ensuring that data remains accessible and robust against potential failures or security breaches [27].

As shown in Figure 4, IPFS cluster is made up of a coordinated group of nodes operating within the IPFS, working together to manage and distribute hashed files. This collaborative approach greatly improves the efficiency of data distribution in peer-to-peer file sharing. In this research, focused on the establishment of four dedicated IPFS nodes. The formation of a private network commences with the generation of a unique swarm key on the primary node, which acts as a secure identifier. This key is then distributed to all other nodes intended to join the network. Subsequently, each node is methodically integrated into the network by adding the hash address of the boot node, which serves as a central hub for initiating interactions between the nodes. Once the private IPFS network is fully operational and all nodes are interconnected, an IPFS-Cluster can be constructed on this framework. This cluster enables automatic data replication, leveraging the bootstrapping process to improve the network's efficiency, reliability, and resilience in the management and distribution of data.

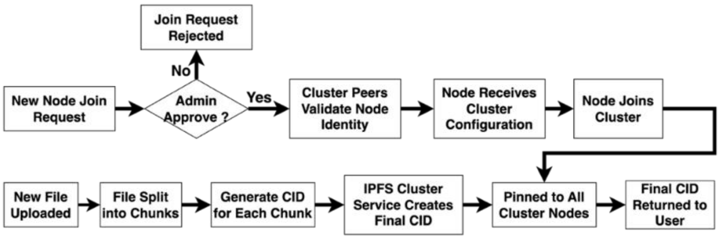

Figure 4. IPFS cluster diagram

2.5. System flow

The flowchart depicts the amalgamation of EMR with Blockchain technology and the IPFS to bolster the security, transparency, and integrity of medical data management practices. This integration aims to create a more reliable framework for handling sensitive medical information, ensuring that data remains secure and accessible only to authorized personnel.

As shown in Figure 5, process of uploading begins when an administrator submits patient medical files through the administrative web portal. The system employs a CDC mechanism to identify the newly added medical record. Following this detection, the file is transferred to IPFS, where it is segmented into various fragments and stored in a decentralized manner. A unique CID is generated to identify the file within the IPFS network, and this information, along with the CID, the hospital's address, the action taken (upload), the ownership status, the patient's identification number (NIK), and the timestamp, is recorded on the blockchain to guarantee the protection and reliability of the data. During the download process, an authorized user or administrator accesses the patient's portal to choose the medical file they wish to download. Prior to the download, a digital signature is affixed to the file to verify its authenticity. Once the file is downloaded, the action, along with relevant details such as CID, the user’s address, the action taken (access), the status (shared), the patient’s NIK, and the timestamp, is documented on the blockchain. The deletion process follows a similar protocol, where the administrator removes a patient’s medical file through the admin portal, and this action is also logged on the blockchain, ensuring a comprehensive record of all interactions with the medical data.

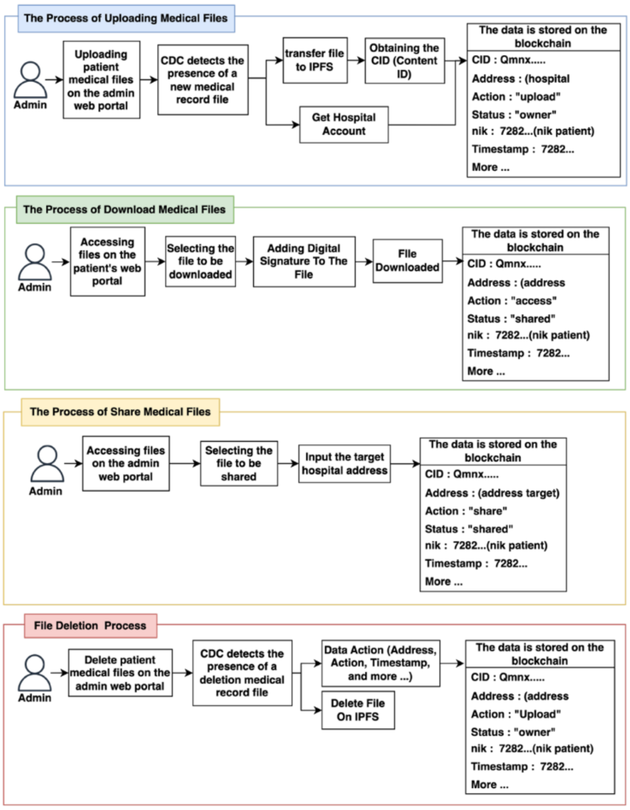

Figure 5. Flow process diagram

3. RESULTS AND DISCUSSION

This section highlights the key findings that emerged from the integration of OpenEMR with blockchain technology and the IPFS. The focus is on evaluating the performance and security of the system across various processes, including document upload, sharing, access, deletion, and integrity analysis. Each process is analyzed in terms of its objectives, outcomes, and impact on system reliability. The following subsections provide detailed results and discussions for each process. The datasets used for this experiment included anonymized patient medical records consisting of PDFs files. These files ranged in size from 1MB to 100MB, representing typical usage in a healthcare platform. Table 2 provides a detailed overview of the specifications for each node used in this experiment, which includes four nodes allocated for both the blockchain and IPFS.

Table 2. Node specification

Items	Specification
CPU	3 Core
RAM	7 GB
Storage	100 GB

3.1. Upload document process

In this research, an EMR system was implemented using OpenEMR. The upload document process is an essential step in ensuring the secure and reliable storage of medical records within the EMR system, particularly OpenEMR, when integrated with blockchain and IPFS technologies through CDC. During this process, documents are first uploaded into OpenEMR, where they are then processed by the CDC system, which detects changes and synchronizes them with both the blockchain and IPFS. The results from the CDC are depicted in Figure 6.

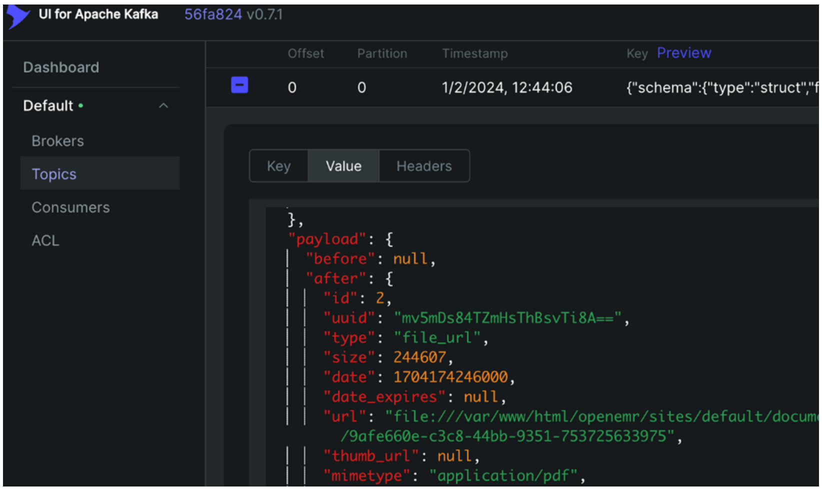

Figure 6. CDC data in Kafka

The results indicate that the time required for the upload process can be segmented into three main components: blockchain transaction time, IPFS upload time, and CDC process time. The blockchain transaction time refers to the duration needed to hash and record document metadata onto the blockchain, ensuring that the document's integrity is preserved. The IPFS upload time measures how long it takes to store the actual document content within the distributed IPFS network, providing redundancy and accessibility. The CDC process time accounts for the period necessary for the system to detect changes and update the blockchain and IPFS accordingly.

$$Average\ Upload\ Time = \frac{\Sigma\left(\frac{Upload\ Time}{File\ Size\ (MB)}\right)}{Number\ of\ Entries} \quad (6)$$

$$Average\ Transaction\ Time = \frac{\Sigma(transaction\ time)}{Number\ of\ Entries} \quad (7)$$

$$Average\ CDC\ Time = \frac{\Sigma\left(\frac{Upload\ Time}{File\ Size\ (MB)}\right)}{Number\ of\ Entries} \quad (8)$$

Comparative analysis through graphical representation highlights the overall efficiency of the process. Our results, as shown in Figure 7, using (6) to (8) indicate that the typical upload time to IPFS is approximately 0.7 seconds per megabyte, while the time required for processing transactions related to storing file activity records is substantial, averaging 4.2 seconds per transaction. The CDC process time is highly sensitive to file size, averaging around 1.1 seconds per MB. Additionally, the upload time to OpenEMR averages 0.98 seconds per MB, indicating a significant overhead when handling larger files.

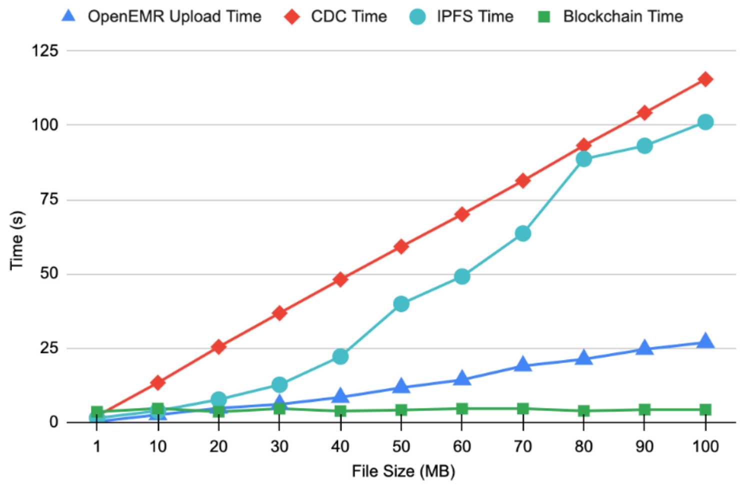

Figure 7. Upload time

Table 3 shows processing durations for CDC across file sizes from 1 MB to 100 MB. The columns detail the time for each phase: "DB to Kafka (s)" reflects the transfer time from the database to Kafka, "Kafka to Consumer (s)" indicates the transmission time to the consumer, and "Consumer Get File (s)" shows how long the consumer takes to process the file. The "Total Time (s)" column summarizes the overall CDC duration. The data reveals that as file sizes increase, processing times for each phase also rise, indicating that larger files require longer processing times throughout the CDC workflow.

Table 3. Change data capture proccess time

File Size (mb)	DB to Kafka (s)	Kafka to consumer (s)	Consumer get file (s)	Total time (s)
1	0.196	0.259	1.641	2.096
10	0.122	0.33	12.844	13.296
20	0.132	0.312	24.957	25.401
30	0.123	0.287	36.341	36.751
40	0.172	0.265	47.658	48.095
50	0.145	0.302	58.741	59.188
60	0.135	0.272	69.654	70.061
70	0.164	0.267	80.987	81.418
80	0.153	0.327	92.762	93.242
90	0.139	0.272	103.871	104.282

3.2. Share and access document process

The share and access document process is critical for enabling secure document sharing and maintaining an immutable record of access events. This process ensures that authorized users can share medical documents with other entities while keeping a transparent audit trail. As shown in Figures 8 and 9, it indicates that the healthcare institution's admin can securely share and view medical record files with the intended hospital, and every access can be monitored. Additionally, the file owner admin can control access rights to the file. The primary objective in this context is to ensure that access to documents is limited exclusively to those who have received authorization, and all access events are permanently recorded on the blockchain, protecting against unauthorized access while ensuring adherence to data protection laws is essential.

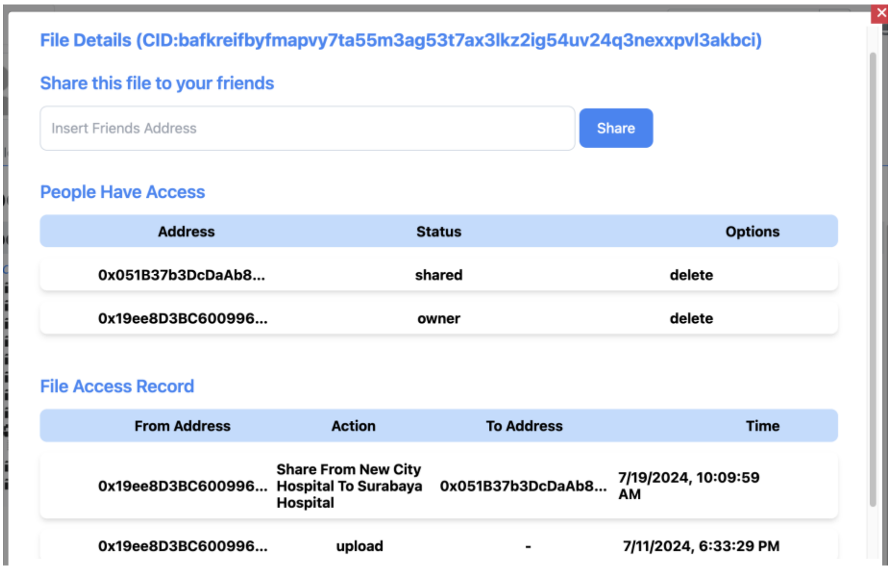

Figure 8. Share and access record

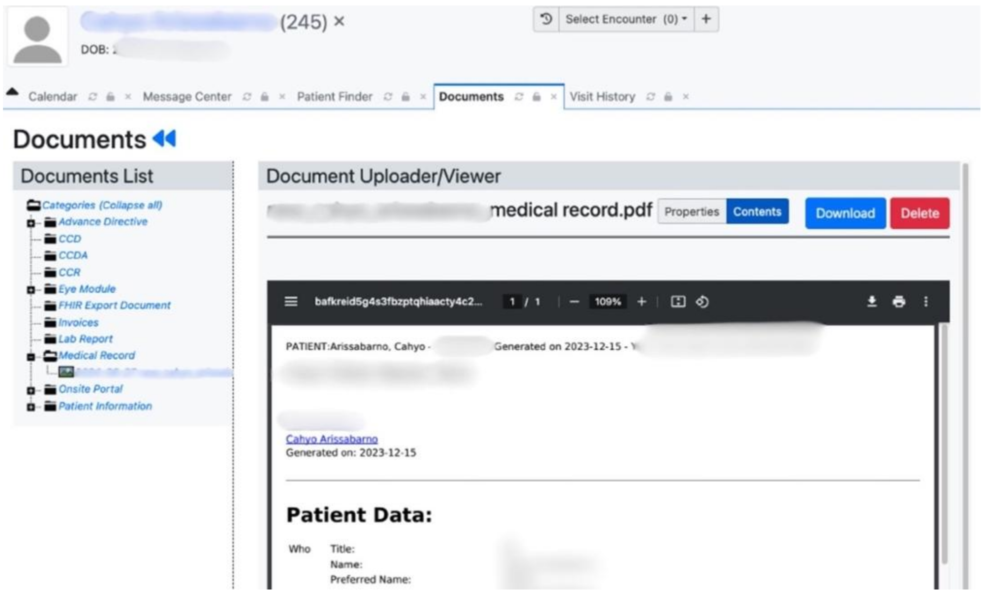

Figure 9. OpenEMR view document page

Results show that documents can be shared securely using cryptographic access controls, and real-time tracking of access events is effectively maintained. This tracking not only provides a comprehensive history of who accessed what document and when but also enforces strict data protection measures. The implementation of cryptographic keys during the sharing process, combined with blockchain's immutable nature, ensures that document sharing is both secure and traceable.

3.3. Delete document process

The delete document process is designed to manage the removal of documents from the EMR system while ensuring that such actions are auditable. When a document is deleted from OpenEMR, this action is recorded on the blockchain, creating an unalterable record of the deletion event. In the context of IPFS, although documents may not be physically deleted due to the distributed nature of the system, they can be "unpinned" to reduce accessibility, effectively rendering them unavailable for future access. The primary objective of this process is to maintain transparency and verifiability in data management, ensuring that document deletions are properly tracked and compliant with data governance policies. The results demonstrate that the system can effectively record deletions on the blockchain, providing an audit trail that upholds the integrity of data management practices. Additionally, the unpinning time on IPFS highlights the process's efficiency in making documents inaccessible while still adhering to the principles of decentralized storage.

3.4. Document integrity analysis

The analysis of document integrity is grounded in the confidentiality, integrity, and availability (CIA triad). In terms of confidentiality, the system ensures that medical documents are only accessible to authorized users, utilizing blockchain and cryptographic techniques to control and monitor access. This approach effectively protects sensitive information from unauthorized users. Regarding integrity, blockchain technology plays a pivotal role in guaranteeing that medical records remain unaltered after their initial upload. Any modification attempts would be detected due to the mismatch in the blockchain, thus preserving the original document's state. Lastly, the availability of documents is ensured through the IPFS system, which, due to its decentralized nature, provides redundancy across multiple nodes. This ensures that even if some nodes become unavailable, the documents can still be retrieved from other parts of the network. The combination of these technologies in the EMR system strongly upholds the CIA triad principles, ensuring that the medical documents are confidential, unaltered, and always available when needed.

4. CONCLUSION

The integration of EMR with private blockchain and IPFS Cluster effectively addresses the challenges of document integrity and traceability in healthcare. By combining IPFS's decentralized storage and blockchain's transparent ledger, the system ensures secure and reliable management of sensitive medical data. The use of CDC enables real-time synchronization with existing EMR systems, providing a seamless and efficient solution without the need for new software development. Experimental results demonstrate strong performance, with document uploads averaging 0.7 seconds per MB, transaction processing at 4.2 seconds, and CDC processing at 1.1 seconds per MB. These results show the system's ability to uphold the CIA triad, confidentiality, integrity, and availability, while significantly improving data security and reliability in healthcare. These results establish a foundation for further analysis and potential performance enhancements in blockchain-IPFS based integration method with EMR. Future research could focus on identifying the most suitable blockchain and decentralize storage technologies to optimize integration with EMR.

FUNDING INFORMATION

This research was supported by the Ministry of Education Culture, Research, and Technology (Kemdikbudristek) with the scheme "Penelitian Dasar - Penelitian Tesis Magister". (Grant Numbers: 67/SPK/D.D4/PPK.01.APTV/III/2024).

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Cahyo Arissabarno	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
Sritrusta Sukaridhoto	✓	✓			✓		✓			✓		✓	✓	✓
Idris Winarno	✓	✓		✓	✓		✓			✓		✓	✓	✓
Rizqi Putri Nourma Budiarti	✓				✓		✓			✓				✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

DATA AVAILABILITY

The data that support the findings of this study are not publicly available and can only be obtained upon reasonable request from the corresponding author.

REFERENCES

[1] F. Wurster *et al.*, "The Implementation of an Electronic Medical Record in a German Hospital and the Change in Completeness of Documentation: Longitudinal Document Analysis," *JMIR Medical Informatics*, vol. 12, no. 1, 2024, doi: 10.2196/47761.

[2] A. Windari, E. Susanto, and I. Q. Fadhilah, "Hospital administrative services with electronic medical records: A meta-analysis," *Journal of Public Health and Development*, vol. 21, no. 3, pp. 333–348, Oct. 2023, doi: 10.55131/jphd/2023/210325.

[3] S. Thakur, B. Gupta, U. Mathur, and D. Bansal, "Electronic Health Record Systems for Enhanced Medical Care: A Survey," in *Proceedings of the 2023 International Conference on Intelligent Systems for Communication, IoT and Security, ICISCoIS 2023*, 2023, doi: 10.1109/ICISCoIS56541.2023.10100356.

[4] F. Wurster *et al.*, "The implementation of an electronic medical record (EMR) and its impact on quality of documentation," *European Journal of Public Health*, vol. 33, no. Supplement_2, 2023, doi: 10.1093/eurpub/ckad160.864.

[5] T. Rahman, S. I. Mouno, A. M. Raatul, A. K. Al Azad, and N. Mansoor, "Verifi-Chain: A Credentials Verifier using Blockchain and IPFS," *arXiv*, Jul. 2023, doi: 10.48550/arXiv.2307.05797.

[6] I. Bashir, *Mastering blockchain : inner workings of blockchain, from cryptography and decentralized identities, to DeFi, NFTs and Web3*, Packt Publishing, 2023.

[7] C. Arissabarno *et al.*, "Blockchain Integration for Mixed Reality Based Smart Lab Systems," in *IES 2023 - International Electronics Symposium: Unlocking the Potential of Immersive Technology to Live a Better Life, Proceeding*, 2023, doi: 10.1109/IES59143.2023.10242494.

[8] A. Singh, A. Gutub, A. Nayyar, and M. K. Khan, "Redefining food safety traceability system through blockchain: findings, challenges and open issues," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 21243–21277, 2023, doi: 10.1007/s11042-022-14006-4.

[9] C. Arissabarno, S. Sukaridhoto, and I. Winarno, "Secure & Traceable Companies File Management System Using Blockchain and IPFS," in *2024 IEEE International Symposium on Consumer Technology (ISCT) (ISCT'24)*, Bali, Indonesia, Aug. 2024, p. 6.

[10] W. N. Hidayat *et al.*, "Digital Twin System for Smart Buildings Integrated with Blockchain and Mixed Reality Technology," in *2024 IEEE International Symposium on Consumer Technology (ISCT) (ISCT'24)*, Kuta, Bali, Indonesia, Aug. 2024, p. 6.

[11] A. Prayudi, O. S. Hakim, M. A. Zainuddin, S. Sukaridhoto, and R. P. N. Budiarti, "Private Blockchain-Based Platform for Cool Box Truck Based on LoRa Communication Protocol," in *2024 IEEE International Symposium on Consumer Technology (ISCT) (ISCT'24)*, Kuta, Bali, Indonesia, Aug. 2024, p. 7.

[12] N. A. Satrio, S. Sukaridhoto, M. U. H. Al Rasyid, R. P. N. Budiarti, I. A. Al-Hafidz, and E. D. Fajrianti, "Blockchain integration for hospital information system management," *Bali Medical Journal*, vol. 11, no. 3 SE-ORIGINAL ARTICLE, pp. 1195–1201, Sep. 2022, doi: 10.15562/bmj.v11i3.3540.

[13] T. Hovorushchenko, A. Moskalenko, and V. Osyadlyi, "Methods of medical data management based on blockchain technologies," *Journal of Reliable Intelligent Environments*, vol. 9, no. 1, pp. 5–16, Mar. 2023, doi: 10.1007/s40860-022-00178-1.

[14] M. Shao, M. Liu, and Z. Wang, "Privacy-preserving Electronic Medical Records Sharing Solution Based on Blockchain," *International Journal of Network Security*, vol. 25, no. 1, pp. 68–75, 2023, doi: 10.6633/IJNS.202301.

[15] R. Pakkala, "Blockchain Enabled Decentralized Application for Securing Electronic Medical Records with Smart Contracts," *Research Square preprint*, 2023, doi: 10.21203/rs.3.rs-2807625/v1.

[16] S. A. H. Mohsan, A. Razzaq, S. A. K. Ghayyur, H. K. Alkahtani, N. Al-Kahtani, and S. M. Mostafa, "Decentralized Patient-Centric Report and Medical Image Management System Based on Blockchain Technology and the Inter-Planetary File System," *International Journal of Environmental Research and Public Health*, vol. 19, no. 22, Nov. 2022, doi: 10.3390/ijerph192214641.

[17] A. Andreakis and I. Papapanagiotou, "DBLog: A Watermark Based Change-Data-Capture Framework," *arXiv*, Oct. 2020, doi: 10.48550/arXiv.2010.12597.

[18] L. Hao, T. Jiang, Y. Lin, and Y. Lu, "Methods for Solving the Change Data Capture Problem," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 153, 2023, doi: 10.1007/978-3-031-20738-9_87.

[19] F. M. Imani, Y. D. L. Widyasari, and S. P. Arifin, "Optimizing Extract, Transform, and Load Process Using Change Data Capture," in *2023 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2023, pp. 266–269, doi: 10.1109/ISRITI60336.2023.10468009.

[20] A. Sayar, S. Arslan, T. Cakar, S. Ertugrul, and A. Akcay, "High-Performance Real-Time Data Processing: Managing Data Using Debezium, Postgres, Kafka, and Redis," in *2023 Innovations in Intelligent Systems and Applications Conference, ASYU 2023*, 2023, doi: 10.1109/ASYU58738.2023.10296737.

[21] L. F. Barbulescu, E. Ganea, and N. Enescu, "Automated script-based engine for Apache Kafka messaging system," in *Proceedings - RoEduNet IEEE International Conference*, 2023, doi: 10.1109/RoEduNet60162.2023.10274942.

[22] N. Sanjana, S. Raj, H. Vishalakshi Prabhu, and S. Sandhya, "Real-time Event Streaming for Financial Enterprise System with Kafka," in *2023 3rd Asian Conference on Innovation in Technology, ASIANCON 2023*, 2023, doi: 10.1109/ASIANCON58793.2023.10270532.

[23] A. Talukdar, "Analysis of Streaming Information Using Subscriber-Publisher Architecture," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021, doi: 10.1007/978-3-030-68449-5_7.

[24] P. Ariappampalayam Krishnamoorthi, S. Shahid, and O. Boydell, "Preserving Privacy in Private Blockchain Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2022, doi: 10.1007/978-3-030-96527-3_8.
[25] P. Purwono, K. Nisa, S. K. Wibisono, and B. P. Dewa, "Private Blockchain in the Field of Health Services," *Journal of Advanced Health Informatics Research*, vol. 1, no. 1, 2023, doi: 10.59247/jahir.v1i1.14.
[26] R. Taş, "Smart Contract Security Vulnerabilities," *Erzincan University Journal of Science and Technology*, vol. 16, no. 1, pp. 196–211, 2023, doi: 10.18185/erzifbed.1105551.
[27] P. Kumar, M. Gupta, and R. Kumar, "Improved Cloud Storage System Using IPFS for Decentralised Data Storage," in *2023 International Conference on Data Science and Network Security, ICDSNS 2023*, 2023, doi: 10.1109/ICDSNS58469.2023.10245317.

BIOGRAPHIES OF AUTHORS

Cahyo Arissabarno is a Master of Engineering student in Informatics and Computer Engineering at Politeknik Elektronika Negeri Surabaya, Indonesia. He completed his Bachelor of Applied Science in Computer Engineering from the same institution in August 2023. He is a member of the Human-Centric Multimedia Lab. His research interests focus on blockchain, internet of things, and distributed systems. He can be contacted at email: cahyo@pasca.student.pens.ac.id.

Sritrusta Sukaridhoto received his B.E. in Electrical Engineering from Sepuluh Nopember Institute of Technology in 2002 and a Ph.D. in Communication Networks Engineering from Okayama University in 2013. He joined Politeknik Elektronika Negeri Surabaya as a Lecturer in 2002 and became Assistant Professor in 2011. He is currently Head of the Human-Centric Multimedia Lab and collaborates with government and industry. His research interests include computer networks, immersive multimedia, and the industrial internet of things. He has received several awards, including the IEEE Young Researcher Award in 2009, and is a member of IEEE. He can be contacted at email: dhoto@pens.ac.id.

Idris Winarno received the B.Eng. degree in Information Technology from Politeknik Elektronika Negeri Surabaya (PENS), Indonesia, in 2005, the M.S. degree in Computer Science from Sepuluh Nopember Institute of Technology, Indonesia, in 2008, and the Dr.Eng. degree in Computer Science from Toyohashi University of Technology, Japan, in 2018. He joined the Department of Computer Science, PENS, as a Junior Lecturer, in 2008. His research interests include computer networks, network security, and resilient computing. He can be contacted at email: idris@pens.ac.id.

Rizqi Putri Nourma Budiarti received the B.Eng. degree in Computer Systems Engineering from Institut Teknologi Sepuluh Nopember (ITS), Indonesia, in 2009, and the M.Eng. degree in Multimedia Intelligent Networks from the same institution in 2017. She is currently a Lecturer in the Department of Information Systems, Faculty of Business Economics and Digital Technology, Universitas Nahdlatul Ulama Surabaya (UNUSA). Her research interests include machine learning, data mining, virtual reality, networking, and big data. She can be contacted at email: rizqi.putri.nb@unusa.ac.id.