Hadi and Ghani *Iraqi Journal of Science, 2025, Vol. 66, No. 12, pp: 5730- 5750*
DOI: 10.24996/ijs.2025.66.12.37

ISSN: 0067-

A Distributed Blockchain Framework for Secure Healthcare Across Multiple Channels

Hend Abdul Ameer Hadi[1]* Rana Fareed Ghani[2]
[1] *Informatics Institute for Postgraduate Studies, Iraq Commission for Computers & Informatics, Baghdad, Iraq*
[2] *Department of Computer Science, University of Technology-Iraq, Baghdad 10066, Iraq*

Received: 8/7/2024 Accepted: 28/11/2024 Published: 30/12/2025

Abstract

Communication and information technologies have facilitated the rapid adoption of electronic medical records, leading to patient privacy and data security concerns. Blockchain technology offers a promising solution to address these issues. However, scalability remains a significant challenge for blockchain-based electronic health records (EHR) systems. In this study, we aimed to develop and evaluate an EHR management system based on blockchain technology. Therefore, we propose a management model based on organizations and user roles and implemented it using Hyperledger Fabric and the InterPlanetary File System (IPFS). The blockchain consists of three channels: one for patient registration and EHR retrieval and two additional channels dedicated to two hospitals for storing patients' EHRs. A scalable multichain e-health system using the Hyperledger Fabric platform provides a practical option to address scalability issues and protect patients' privacy, security, and medical data. The proposed model uses IPFS to store medical images and generate hash values, which are then stored in the blockchain. The system was evaluated using Hyperledger Explorer and Hyperledger Caliper, focusing on several performance metrics: transactions per hour, transactions per minute, blocks per hour, blocks per minute, response time, maximum latency, minimum latency, average latency, throughput, CPU and memory usage, and runtime. A comparative analysis was conducted against single-ledger EHR systems to assess the proposed system's performance. The Hyperledger Caliper report shows that the average latency for each organization ranges from 0.11 to 0.55, and the throughput ranges from 24.2 to 200 for 1000 assets at sending rates of 25, 50, 100, and 200.

Keywords: Blockchain, Multi-chain, Chain-code, Electronic Healthcare Records, Scalability, Hyperledger Fabric.

إطار عمل موزع باستخدام تقنية البلوك تشين للأمان في الرعاية الصحية عبر قنوات متعددة

هند عبد الامير هادي[1] رنا فريد غني[2]
[1] معهد المعلوماتية للدراسات العليا، الهيئة العراقية للحاسبات والمعلوماتية، بغداد، العراق
[2] علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق

الخلاصه

أدى الاعتماد السريع للسجلات الطبية الإلكترونية التي سهلتها تكنولوجيات الاتصالات والمعلومات إلى مخاوف بشأن خصوصية المريض وأمن البيانات. تقدم تقنية (Blockchain) حلاً واعداً لمعالجة هذه المشكلات. ومع ذلك، لا تزال قابلية التوسع تمثل تحديًا كبيرًا لأنظمة السجلات الصحية الإلكترونية (EHR)

---

*Email: phd202210700@iips.edu.iq

التي تستخدم تقنية (Blockchain) . هدفنا في هذه الدراسة إلى تطوير وتقييم نظام إلكتروني لإدارة السجلات الصحية يعتمد على تقنية (blockchain) متعددة القنوات. ولذلك، اقترحنا نموذجًا لإدارة السجلات الصحية الإلكترونية استنادًا إلى أدوار المؤسسات والمستخدمين ونقوم بتنفيذه باستخدامHyperledger Fabric ونظام (IPFS)تم بناء نظام ال blockchain بحيث يتكون من ثلاث قنوات: واحدة لتسجيل المرضى واسترجاع السجلات الصحية الإلكترونية، وقناتين إضافيتين مخصصتين لمستشفيين لتخزين السجلات الصحية الإلكترونية للمرضى. يوفر نظام الصحة الإلكترونية متعدد السلاسل القابل للتوسع باستخدام منصة Hyperledger Fabricخيارًا عمليًا لمعالجة مشكلات قابلية التوسع وحماية خصوصية وأمن المرضى وبياناتهم الطبية. يستخدم النموذج المقترح IPFS لتخزين الصور الطبية. تم تقييم النظام المقترح باستخدام Hyperledger Explorer و Hyperledger Caliper، مع التركيز على عدة معايير أداء. تشمل هذه المعايير (transactions per hour)، و(transactions per minute)، (blocks per hour)، و(blocks per minute)، ووقت الاستجابة (response time)، و(maximum latency)، و(minimum latency)، و(average latency)، والإنتاجية (throughput)، ووقت التشغيل (runtime). يتم إجراء تحليل مقارن مع أنظمة السجلات الصحية الإلكترونية (EHR) التي تعتمد على سجل واحد لتقييم أداء النظام المقترح. يُظهر تقرير Hyperledger Caliper أن متوسط التأخير لكل منظمة يتراوح بين 0.11 و 0.55، وأن الإنتاجية تتراوح بين 24.2 و 200 عند إرسال 1000 أصل بمعدلات إرسال تبلغ 25، 50، 100، و200.

1. Introduction

In recent years, the emergence of blockchain technology has piqued the interest of the healthcare industry [1]. The healthcare system comprises three key components: main services suppliers such as hospital administrations, doctors, nurses, and technicians, emergency-related services, and patients [2]. Blockchain is regarded as a general-purpose technology and has been effectively used across various industries [3], [4], [5], [6]. The most popular use case for blockchain applications in healthcare is electronic health records (EHRs) [7], [8], [9], [10]. The distributed ledger of blockchain technology presents a novel alternative to traditional data management techniques that depend on third-party cloud services or on-site data servers. This could help with EHRs' privacy and security issues [11].

Data is frequently created, retrieved, and broadcasted in the clinical healthcare sector, which is data intensive. Because of the sensitive nature of data and other constraints, such as security and privacy, storing and sharing this considerable quantity of data is both critical and challenging. Diagnostics in healthcare and clinical settings heavily depend on safe, secure, and scalable (SSS) data sharing. Data sharing is crucial to enable clinical practitioners to send their patients' clinical data to the appropriate authorities for prompt follow-up [12].

Caregivers and general practitioners should transfer patients' clinical data promptly and sensitively to guarantee that both participants have up-to-date, complete information and to protect patient privacy. On the other hand, clinical data may be sent remotely to a specialist (who may be located far away) to obtain an expert opinion in two widely utilized fields: telemedicine and e-health [13], [14].

Health records, including EHRs, patient information, and clinical images, are stored centrally in servers, clouds, or databases owned by healthcare organizations. The central database may be vulnerable to cyberattacks threatening EHRs' security and privacy. Sharing health information is difficult because of differing formats and standards. When EHRs are deleted from the central database, they can be permanently lost. Tamper-proofing the system is required to guarantee authorized access. Another issue is that patients do not have complete control over managing their health records since the organizations manage the records. The security and scalability of health records have become key interests as the amount of medical data increases. One of blockchain technology's limitations is that it cannot store large files such as medical images or other health-related data directly on-chain. The InterPlanetary File

System (IPFS) can be quite useful in this regard. IPFS is a peer-to-peer protocol created for distributed file storage that allows for the safe, decentralized sharing of large files [15].

Therefore, technology that efficiently stores data is essential, especially in the healthcare system. This can be accomplished by designing an efficient healthcare distributed system for storing patients' health records that instantaneously addresses security, integrity, privacy, scalability, and interoperability. This research proposed a multi-chain framework for storing and securely accessing records based on the Hyperledger Fabric blockchain. In this work, we constructed a permissioned network in a Hyperledger Fabric with multiple channels that ensure health record scalability, security, integrity, and privacy. Patient data is stored on the blockchain, while medical images and files are stored as hashes generated by storing original files on IPFS to maintain scalability and efficiency.

The rest of the document is organized as follows: Section 2 discusses the related work. Section 3 explains some tools and concepts related to the blockchain. Section 4 presents the proposed e-healthcare system framework. The outcomes and performance evaluation are covered in Section 5. Section 6 concludes the study with suggestions for future work.

2. Related Work

Blockchain technology can be used in many fields, such as financial services, healthcare, and education. Even though it is still in its early development phases, much interest has been expressed in using decentralization technology within the e-health system. This section provides an overview of the latest studies in blockchain-based e-health systems.

Singh et al. [16] presented a patient-centric paradigm for a blockchain-based healthcare system that resolves data privacy, immutability, and authentication. They used the Hyperledger platform and Hyperledger Caliper for implementation and evaluation. The main results indicated that average write latencies are approximately 47% more than read latencies, and maximum throughput is reduced when transitioning from One-org1peer to Two-org2peer. Doubling the block size resulted in an approximate tenfold augmentation in transactions per second (TPS). The framework facilitates client communication over a REST server, potentially revolutionizing next-generation electronic health record systems.

Mani et al. [15] presented a patient-centric healthcare data management system. Hashes of health records are saved in Hyperledger Fabric as chains for on-chain storage. At the same time, the IPFS provides off-chain solutions to encrypt real health data and save it securely. IPFS is the decentralized cloud storage platform that provides confidentiality and scalability to solve the blockchain data storage problem. They used the Byzantine fault tolerance (BFT) consensus to host a security-smart contract to ensure patient privacy. The performance of the proposed system is tested in terms of resource utilization, transaction latency, and transaction per second using Hyperledger Caliper benchmarks.

Panwar et al. [17] provided an innovative architecture for managing personal health records that uses blockchain technology and IBM cloud data lakes to enable efficient healthcare administration. The suggested method focuses on increasing throughput and latency. Several matrices, including the F1 score, recall, and confusion matrices, are used to calculate the outcome of the suggested system. They used the practical Byzantine fault tolerance mechanism. As a result, the proposed approach outperformed earlier methods in terms of accuracy and output. The system completed 2,300 transactions in 140 seconds, achieved 33.12 seconds of latency at 250 transactions per second, and reached 114 transactions per second with a configuration of two organizations and two peers.

Thakkar and Shah [18] proposed using Hyperledger Fabric to improve security, interoperability, privacy, and transparency. They used Arweave's peer-to-peer databases to permanently store and retrieve data. To attain user, unlinkability, and transaction data privacy throughout consensus, they use Hyperledger Fabric's Java Software Development Kit and privacy-preserving features, such as private data collection, identity mixer, and transient field.

Abutaleb et al. [19] designed a framework for controlling patient health records to ensure that patients can provide those who need access to their medical records with the necessary authorizations. All activities are recorded in this framework through the blockchain and usage controls. The study aimed to provide a user-centric and privacy-aware experience using this framework. They used Hyperledger Fabric to start their design due to its performance.

Díaz and Kaschel [20], using a two-channel EHR configuration, proposed a management model based on entities and user roles, implemented with Hyperledger Fabric. They developed a prototype in Fabric with one and two simultaneous channel configurations. They designed and experimented to verify whether the proposed system can be improved in terms of scale.

Table 1 summarizes related works on blockchain technology, databases, and the number of organizations, peers, and channels.

Table 1: Summary of the related work

Study	Blockchain type	Storage	No. of Orgs	No. Peers	No. of Channels	Consensus algorithm	Performance metrics
[16]	Hyperledger Fabric	On-chain	1-2	1-2	1	-	• Latency • Throughput • CPU utilization • Memory usage • Network traffic (incoming and outgoing) • Disc write
[15]	Hyperledger Fabric	• On-chain • Off-chain (IPFS)	1	3	1	BFT	• Latency Resource utilization
[17]	Hyperledger Fabric	• CouchDB	2	1	1	BFT	• Block creation time • Transactional computational overhead • Upload/download time Throughput
[18]	Hyperledger Fabric	• CouchDB • Off-chain (Arweave peer-to-peer)	5	2-3	1	Replicated and Fault-Tolerant (RAFT)	• Latency • Throughput • CPU utilization • Memory usage • Network traffic (incoming and outgoing)
[19]	Hyperledger Fabric	• CouchDB	2	4	1	-	• Latency • Throughput
[20]	Hyperledger Fabric	• On-chain	4	1	1-2	RAFT	• Latency • Throughput • Number of successful transactions • Throughput achieved per specified send rate

However, there are major scaling problems in the single ledger architecture. This model was created to store all transactions in a single ledger, but when the network grew, it began to struggle. A single ledger becomes a bottleneck in the network when the number of transactions increases, obstructing network performance. This obstruction could lead to increased execution times and latency. Consecutively, the network's ability to scale is reduced. This study proposed a multi-channel e-health system that implements a ledger for each hospital to save its transactions and a single ledger for patient data to solve this problem.

3. Blockchain Overview

The concepts and tools of the blockchain that are used in the proposed system are briefly addressed in this section.

3.1 Blockchain

Blockchain technology uses distributed and decentralized ledgers to help secure transactions and ensure transparency. It uses cryptographic hashing, consensus mechanisms, and immutability to guarantee an unchangeable record of information. This further enhances security and efficiency in many businesses while eliminating intermediaries [21]. A blockchain is a distributed ledger containing cryptographically signed transactions that are shared, immutable, and organized into blocks. Each block has a hash value linking it to the previous one. All nodes have the exact copy of the ledger and collaborate to add new transactions to the ledger that meet predefined criteria using a consensus algorithm [22].

Multiple types of blockchain technology exist, each with special attributes and intended applications. These include platforms for smart contracts, such as Ethereum, cryptocurrencies, such as Bitcoin, and private blockchain, such as Hyperledger for specialized applications, such as e-health [23]. Blockchains are classified into public, private, and consortium blockchains based on their accessibility and control. A public blockchain is a decentralized network that allows participants to freely join the network without permission to view and transmit transactions, participate in the consensus process, and maintain the shared ledger. Public blockchains are still secure because they use computationally costly consensus methods like puzzle-solving to verify new blocks [23], [24]. [25], [26].

Private or permissioned blockchains are considered for a single company. Participants must have permission to join the network and perform specific tasks to maintain the decentralized nature of the blockchain [24], [25]. The private blockchains that are designed for multiple organizations are called consortium blockchains. As in private blockchains, only authorized participants are permitted to join the network [24].

3.2 Block

In blockchain, a block is the data structure that holds transaction records, and it is made of two pieces: the block header and the block body. The block header consists of the header hash, the parent block hash (previous block), the timestamp in second, nonce (an increased mathematical value), and the Merkle root that stores transaction hashes. The transactions and transaction counter are stored in the block body. The block size determines the block's capacity to store the transactions. Figure 1 shows the architecture of the blockchain block [23].

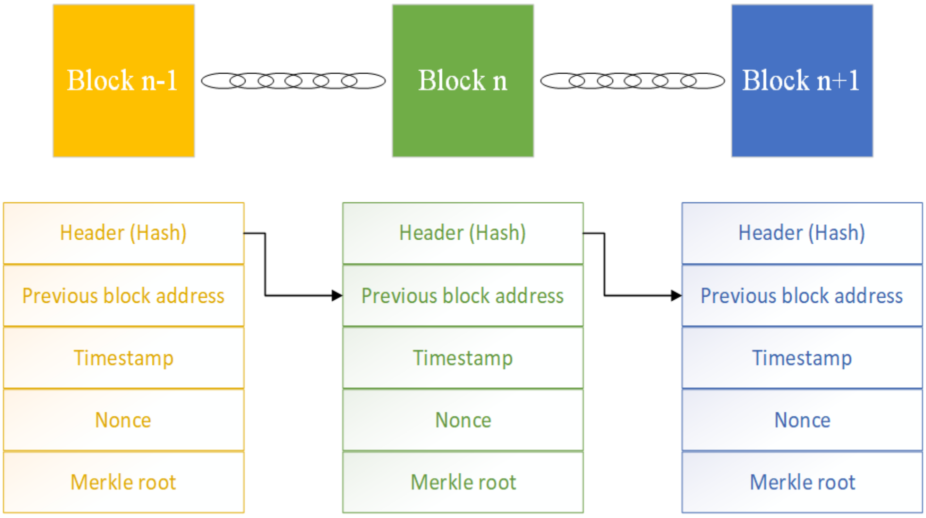


Figure 1: Block architecture

3.3 Hyperledger Fabric

Due to the success of cryptocurrencies like Bitcoin, blockchain technology is popular. However, permissionless blockchains, such as Bitcoin or Ethereum, encounter challenges such as time-consuming execution, limited transaction throughput, wasted resources, and inconsistent data. The Hyperledger project was launched in 2015 by the Linux Foundation, trying to create a solution to address the above issues [26]. Hyperledger Fabric has developed as the principal framework for private blockchains among all available blockchain platforms [22].

Enterprise blockchain applications have widely used Hyperledger Fabric, which is not dependent on cryptocurrency like Bitcoin. It is a network that only verified users can access, guaranteeing transaction security and integrity [28].

Hyperledger Fabric's most critical attributes are scalability, confidentiality, adaptability, and resilience. Due to these attributes, the Hyperledger Fabric-based solution is appropriate for a broad range of industrial applications. Fabric's revolutionary architectural methodology enhanced the blockchain's ability to adjust to issues with non-determinism, resource depletion, and performance to offer this flexibility. A collection of organizations can build their transaction ledgers for their company by leveraging Hyperledger Fabric to establish channels [29].

The Fabric Certificate Authority (CA), peers, orderers, and smart contracts (chaincode) are some of the elements that make up Hyperledger Fabric. Together, these elements enable the Hyperledger Fabric network's transaction processing to be safe and effective [22].

3.3.1 Peer

Peer nodes are divided into five types in the context of Hyperledger Fabric. The certificate node oversees providing registration certificates to nodes and users. A submission node or submitter is responsible for producing proposals and allocating them to the suitable endorsement node. The endorsement node or the endorser is in charge of executing the contract and providing an endorsement response. The sort node (or orderer) handles block stuffing and proposal organizing. The confirm node or committer is responsible for confirming the contract outcomes' validity and preserving the ledger and blockchain structure. These roles are not mutually exclusive; instead, they are logically divided. While some nodes have endorsement or sorting functions, most nodes in the network have validation functions [30].

3.3.2 Chaincode

In the Hyperledger Fabric, the chaincode is the smart contract part of the network. It is a program typically written in Go, Java, JavaScript, or others. All chaincodes have init and invoke functions. Init functions, used to initialize contracts, only run once over the chaincode life cycle. The invoke function is mainly used to update or query the ledger. The update transaction modifies the ledger, while the query transaction just retrieves data from the ledger [30].

3.3.3 Channel

Organizations in the Hyperledger network are connected through channels. Channels provide restricted and isolated access to the transactions and data within a specific set of participants. They provide scalability and flexibility. The network may have multiple channels, allowing different consortium entities to have private independent communication. One of the key benefits of implementing multiple channels is improved scalability and privacy. Because only peers participating in the channel can access transactions and data, there is less of a surface for possible attacks and weaknesses. Every channel keeps its ledger in Hyperledger Fabric, tracking transactions specific to that channel [27].

3.3.4 Orderer

In a blockchain network, transactions are arranged in blocks by the ordering service once the endorser peer nodes have granted them. Before transactions are committed to the ledger, they must first receive cryptographic signatures from each endorsing peer and are then sent to the orderer [31].

3.3.5 Membership Service Provider (MSP)

Participants who are peers, orderers, and clients must register through MSP to join the Hyperledger Fabric network. As a result, not everyone can see the transactions; only authorized members have access permission [32].

3.4 InterPlanetary File System

As a distributed file system, the IPFS connects many peers that use the identical file system. It is a peer-to-peer distributed file system that tracks where files are located and connects nodes through distributed hash tables (DHT). The DHT takes the structure of a Merkle Tree, which uses directed acyclic graphs to organize itself. It removes the need for a central server by using data deduplication technology. IPFS will make frequently requested data readable immediately on subsequent requests. Therefore, IPFS is a high-performance, secure file system with a large storage volume and a content-addressing storage paradigm and supports synchronous access [22], [33].

4. The Proposed Methodology

The latest version of Hyperledger Fabric (v2.5) is used as the fundamental blockchain network to provide trust and immutable ledgers for storing and managing e-health records. The suggested design for the Hyperledger network involves three organizations, Hospital1, Hospital2, and Patient, to securely record patients' personal information. It integrates three separate channels, each supporting an independent ledger for every organization involved, which solves the scalability problem when a single ledger is used across the network. To ensure immutability and security, medical images are stored using IPFS (decentralized storage), while the hash values of these images generated by the IPFS are stored on the Hyperledger Fabric blockchain. Figure 2 shows the architecture of the proposed system.

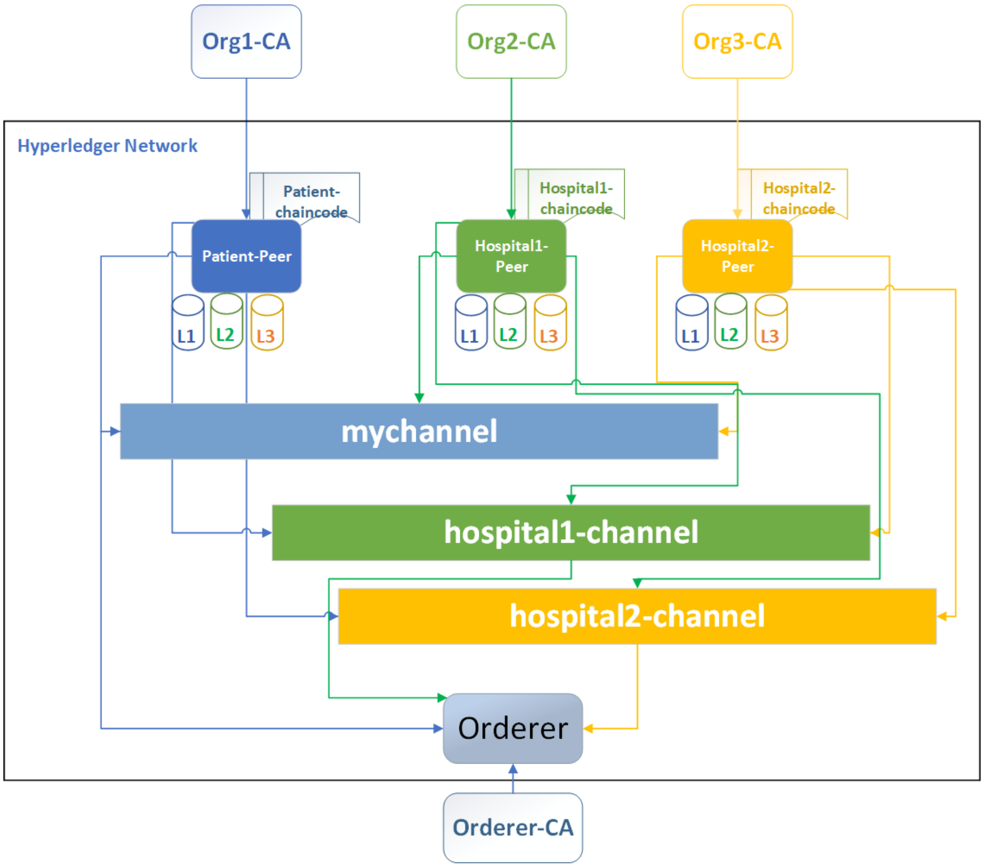


Figure 2: Architecture of the proposed multi-chain Hyperledger Fabric e-health system

4.1 Network Architecture

Organizations and CAs: The e-health network includes three distinct organizations, each with its own CA. The CA uses the Elliptic Curve Digital Signature Algorithm with the P256 curve for digital signatures to ensure robust identity management and authentication.
1. Org1-CA: Patient registration authority
2. Org2-CA: Hospital 1
3. Org3-CA: Hospital 2
4. Orderer-CA: Ordering service

Peer Nodes: Each organization operates a peer node.
1. Patient-Peer (Org1)
2. Hospital1-Peer (Org2)
3. Hospital2-Peer (Org3)

Peers utilize LevelDB as the state database for effective key-value storage and retrieval. The system employs three channels for data separation and confidentiality:
1. mychannel: For patient-peer (Org1) to register patient data and retrieve all patient medical records.
2. hospital1-channel: Private channel for hospital1-peer (Org2) to store and restore medical records.
3. hospital2-channel: Private channel for hospital2-peer (Org3) to store and restore medical records.

Ordering Service: A Raft ordering service based on etcd with three nodes guarantees crash fault tolerance.

4.2 Chaincode Implementation

The proposed system is meant to work in three different parts. The first permits patients to sign up and open an account by submitting their details. An automated identification number (ID) is given to every patient. This ID is necessary because it helps save and recover the patient's EHRs. These EHRs are safely kept within a Hyperledger Fabric blockchain network, guaranteeing they are tamper-proof. Go programming language and the Hyperledger Fabric contract API are utilized to write this chaincode.

The diagram in Figure 3 depicts the interaction between four entities: user interface, Hyperledger software development kit (SDK), peer organization, and orderer organizations. It outlines the procedures for processing a patient's information request.

1. The user makes a transaction request by entering information about the patient using an application or website.
2. The Hyperledger SDK prepares and sends a transaction proposal to the endorsing peers (the three organizations in this study).
3. The endorsing peers (peer organizations) simulate the transaction proposal based on the ledger's current state and return an endorsement response to the SDK.
4. Once the SDK collects the required number of endorsements (as specified by the endorsement policy), it assembles the endorsed transaction and forwards it to the orderer.
5. The orderer organizations then batch the transaction into blocks and broadcast them to peers.
6. The peer organization performs transaction validation and block commitment to append the block to their copy of the ledger.
7. The SDK sends a request-response (confirmation) back to the user interface.
8. Finally, the user receives a network response, completing the transaction.

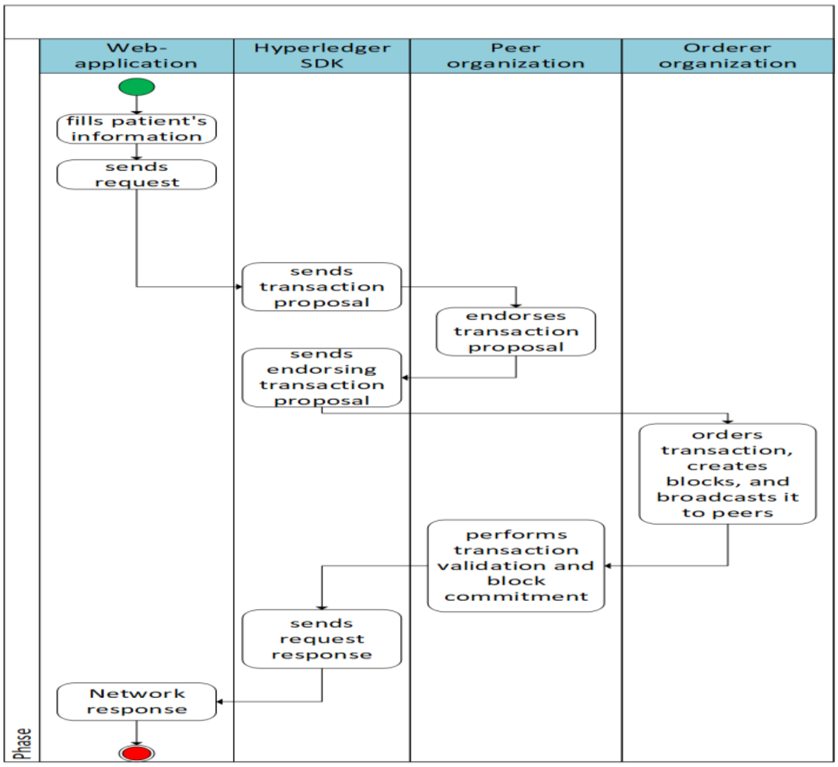


Figure 3: Activity diagram of Hyperledger Fabric transaction flow

4.2.1 Patient Application Chaincode (Organization 1)

The implementation of the patient application in Organization 1 using chaincode is provided in this section. This chaincode has a core that consists of a patient structure. The information about patients, such as their identification ID, name, gender, age, and username with a password for login purposes, are put into this structure. The *Generate-ID* function produces unique consecutive patient IDs to maintain the integrity of the generated patient EHRs and allow ledger traceability. These IDs are constructed using the patient's name, age, and an incremental counter to ensure the uniqueness of each patient's ID registered under this system. The channel (mychannel) deploys this chaincode to store all patients' records on Ledger 1. Algorithm 1 describes a patient chaincode that sends transaction proposals to the Hyperledger Fabric network carrying patients' data, which are used to generate patient IDs automatically.

Algorithm 1: Patient-chaincode (Org1)

```
// Global patient count key
patientCountKey = "patientCount"
// Patient structure
struct Patient {
  id, name, gender, age, username, password
}
Class Patient Contract {

  // Generate a unique patient ID using the patient's name, age, and count
    Function Generate-ID (name, age, count) -> string:
        short name = substring (name, 0, 4)  // Get the first 4 characters of the name
        return short name + toString(age) + toString(count)

    // Add a new patient to the ledger
    Function Add Patient (name, gender, age, username, password):
        count = getPatientCount()       // Get the current patient count
        id = Generate-ID (name, age, count + 1)  // Generate unique ID for the patient
        patient = new Patient (id, name, gender, age, username, password)  // Create a new Patient
instance
        patientJSON = toJSON(patient)      // Convert patient data to JSON format
        PutState(id, patientJSON)       // Store the patient data in the ledger
        setPatientCount(count + 1)    // Update the patient count

    // Retrieve patient details using their unique ID
    Function GetPatient(ctx, id) -> Patient:
        patientJSON = ctx.GetState(id)     // Fetch patient data from the ledger
        return fromJSON(patientJSON)       // Convert JSON data back to Patient structure

    // Get the current patient count from the ledger
    Function getPatientCount() -> int:
        countJSON = GetState(patientCountKey)  // Fetch the patient count from the ledger
        if countJSON exists then
           return toInt(countJSON)
      else
           setPatientCount(0)
     end if

    // Set or update the patient count in the ledger
    Function setPatientCount(count):
        PutState (patientCountKey, toJSON(count))  // Store the updated count in JSON format
}
// Main function to start the chaincode
Function main () -> void:
    chaincode = new Chaincode(PatientContract)
    chaincode.Start()
```

4.2.2 Hospital 1 Chaincode (Organization 2)

A custom Hospital1 chaincode is developed and implemented using the Hyperledger Fabric platform. This chaincode consists of a data structure called a medical record, which is part of a smart contract. Its purpose is to store text data. The structure of a medical record includes several pieces of information such as patient ID, name, age, gender, blood type, medical conditions, admission date, treating doctor, and associated insurance details.

This chaincode is deployed on the hospital1 channel and ensures all records are stored securely on Ledger 2. One important feature of this system is that once the medical record is added to the ledger, it becomes immutable and cannot be changed or deleted. This design choice leverages the inherent immutability and transparency properties of blockchain technology. The Hospital1 chaincode is described in Algorithm 2.

Algorithm 2: Hospital1-chaincode (Org2)

```
// Structure representing a medical record
struct Medical Record {
 id, name, age, gender, blood type, medical conditions, admission date, doctor, insurance, billing amount, room number, admission type, discharge date, medication, test results
}
Class Medical Record Contract {

   // Store a medical record in the ledger
   Function Set Record (id, name, age, gender, blood type, medical conditions, admission date, doctor, insurance, billing amount, room number, admission type, discharge date, medication, test results):
      // Create a new Medical Record object with the provided information
      record = new Medical Record (id, name, age, gender, blood type, medical conditions, admission date, doctor, insurance, billing amount, room number, admission type, discharge date, medication, test results)

      // Convert the Medical Record object to JSON format for storage
      recordJSON = toJSON(record)

      // Store the JSON record in the ledger using the unique ID
      return PutState (id, recordJSON)

   // Retrieve a medical record by its unique ID
   Function GetRecord(id) -> Medical Record:
      // Fetch the record from the ledger using the given ID
      recordJSON = GetState (id)

      // If no record is found, return an error message
      if recordJSON is nil:
         return nil, "Record does not exist"

      // Convert JSON data back to a Medical Record object
      record = fromJSON(recordJSON)
      return record
}
// Main function to start the chaincode
Function main ():
   // Create a new instance of the MedicalRecordContract
   contract = new Medical Record Contract ()

   // Initialize and start the chaincode
   chaincode = new Chaincode(contract)
   chaincode.Start()
```

4.2.3 Hospital 2 Chaincode (Organization 3)

The proposed system for Organization 3 presents a blockchain-based solution to manage healthcare records securely and efficiently using Hyperledger Fabric. The system’s core is the *medical record* structure, designed to store key patient details, including unique ID, name, age, gender, and images’ IPFS hash of associated medical images, ensuring a comprehensive representation of patient data.

This chaincode is deployed on the hospital2 channel and the records are stored on Ledger 3. The Hospital2-chaincode described in Algorithm 3 includes sending a transaction proposal on the Hyperledger Fabric network, which includes patient data and the IPFS hash of the medical image.

Algorithm 3: Hospital2-chaincode (Org3)

```
// Structure representing a record
struct Record {id, name, image-hash   age}
// Record Contract class for managing records
Class Record Contract {
   // Function to create a new record in the ledger
   Function Create Record (id, name, age, image-Hash):
      // Create a new Record object
      record = new Record (id, name, age, image-Hash)
      // Convert the record to JSON format
      recordJSON = toJSON(record)

      // Store the record in the ledger using the record ID
      return PutState(id, recordJSON)

   // Function to retrieve a record by ID
   function Retrieve Record (id) -> Record:
      // Fetch the record from the ledger using the ID
      recordJSON = GetState (id)

      // If the record does not exist, return an error
      if recordJSON is nil:
         return nil, "Record does not exist"

      // Convert JSON data back to a Record object
      record = fromJSON(recordJSON)
      return record
}

// Main function to start the chaincode
Function main ():
   // Create a new instance of the Record Contract
   contract = new Record Contract ()

   // Initialize and start the chaincode
   chaincode = new Chaincode(contract)
   chaincode.Start()
```

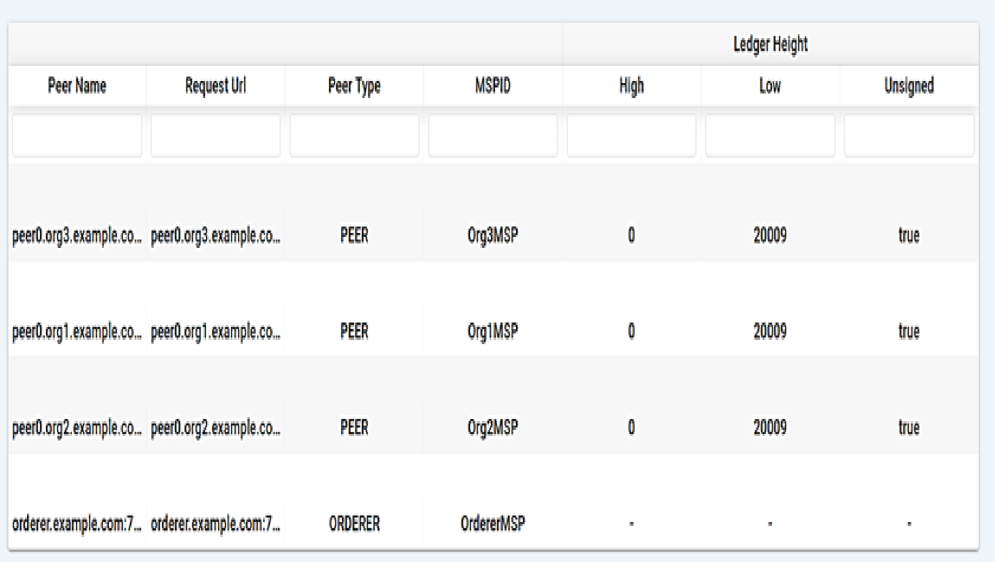
5. Experimental Results and Discussion

The ASUS laptop used to implement the proposed methodology includes the following specifications: 12th Generation Intel(R) Core (TM) i7-12700H 2.30 GHz processor and 16.0 GB of installed RAM. The operating system is Ubuntu 22.04 64-bit. The technologies used in the proposed methodology are listed in Table 2.

Table 2: The technologies used in the proposed methodology.

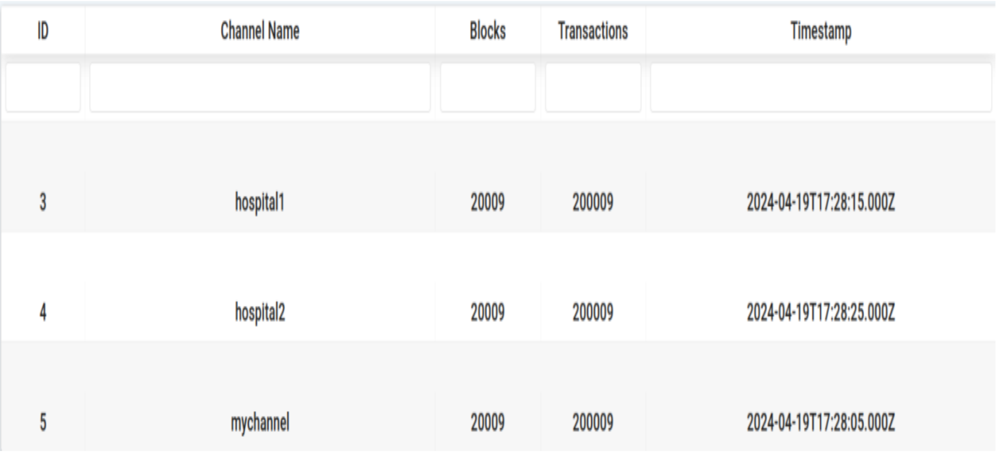
Category	Technology	Purpose/Use
Blockchain Framework	Hyperledger Fabric	The blockchain platform to create a distributed ledger
Containerization	Docker	To run isolated environments for network nodes
Orchestration	Docker Compose	To manage multi-container Docker applications
Programming Language	Go	To develop chaincode and other network components
Monitoring Tool	Hyperledger Explorer Hyperledger Caliper	Visualization tool to monitor and explore blockchain
Development Tool	Visual Studio Code	Code editor for developing and maintaining code
Docker Management Tool	Portainer	Simplified management and deployment of Docker containers

The Hyperledger Fabric network comprises one orderer organization and three peer organizations (Patient, Hospital1, and Hospital2). Each peer or orderer organization has a CA. By utilizing a multichannel, organizations are joined through all channels. The Hyperledger Explorer is used to monitor the blockchain network and to show the experimental results.
Figure 4 displays the nodes (peers) comprising the Hyperledger Fabric network. It shows statistics about each peer and the network structure and members. Three channels are utilized when configuring the Hyperledger Fabric network and establishing connections between peers. These channels consider the MSP identifier, the peer type, and the ledger height corresponding to the number of blocks created. Figure 5 shows the channels.



				Ledger Height		
Peer Name	Request Url	Peer Type	MSPID	High	Low	Unsigned
peer0.org3.example.co...	peer0.org3.example.co...	PEER	Org3MSP	0	20009	true
peer0.org1.example.co...	peer0.org1.example.co...	PEER	Org1MSP	0	20009	true
peer0.org2.example.co...	peer0.org2.example.co...	PEER	Org2MSP	0	20009	true
orderer.example.com:7...	orderer.example.com:7...	ORDERER	OrdererMSP	-	-	-

Figure 4: The Hyperledger Fabric network’s peers and their details



ID	Channel Name	Blocks	Transactions	Timestamp
3	hospital1	20009	200009	2024-04-19T17:28:15.000Z
4	hospital2	20009	200009	2024-04-19T17:28:25.000Z
5	mychannel	20009	200009	2024-04-19T17:28:05.000Z

Figure 5: The Hyperledger Fabric Network channels

5.1 Performance Measurements

The following metrics are used in the result's calculation and generation from Hyperledger Explorer reports and Hyperledger Caliper:

1. Block per hour
2. Block per minute
3. Transaction per hour
4. Transaction per minute
5. Average response time
6. Maximum latency
7. Minimum latency
8. Average latency
9. Throughput
10. CPU usage
11. Memory usage

5.2 Experimental Results of Single-Channel and Multi-Channel Configurations

This part of the analysis compares how single-channel and multichannel methods perform. Based on the findings, it is clear that multi-channel settings are better than single-channel ones as they have separate ledgers for each organization, unlike single channels, where only one ledger is used by all organizations. This becomes more noticeable with growth in network size, where scalability problems often affect single channels. A comparison was made between the two configurations, using Hyperledger Explorer to display the network performance, including approximately 200,000 transactions for each organization (Org1, Org2, Org3). Hyperledger Explorer provided detailed insights into the performance and scalability differences between the two configurations. Figure 6 shows the block/hour for single channel (mychannel) and multichannel (mychannel, hospital1, and hospital2). Figure 7 shows the block generation per minute for a single channel and multichannel.

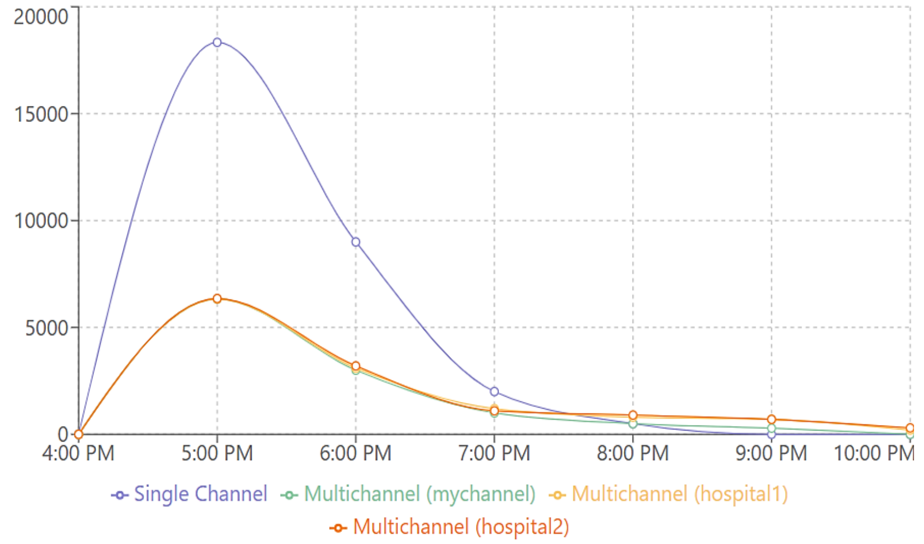


Figure 6: Comparison of blocks/hour for a single-channel vs. multichannel Hyperledger network

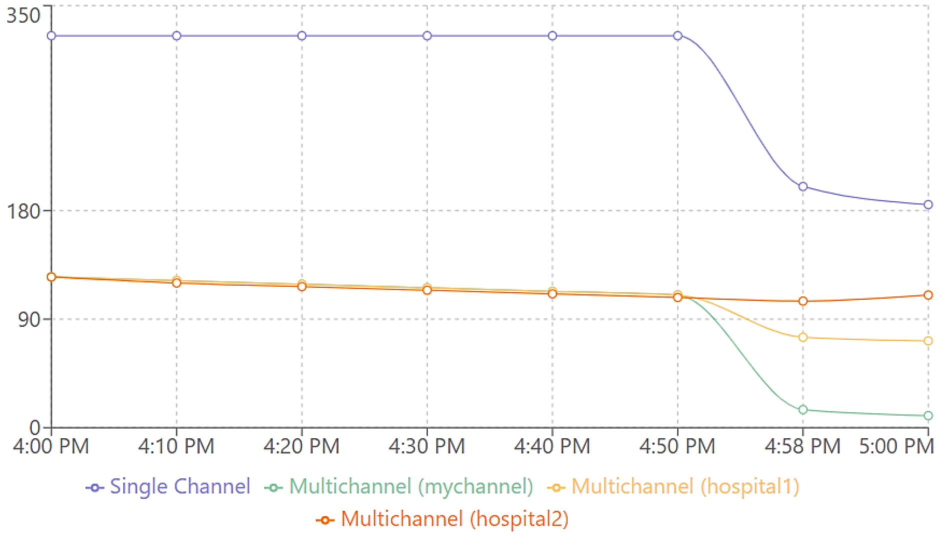


Figure 7: Comparison of blocks/minute for a single-channel vs. multichannel Hyperledger network

Figure 8 displays the transaction/hour rate for single-channel and multichannel Hyperledger network configurations. Figure 9 displays the transaction/minute rate for single-channel and multichannel.

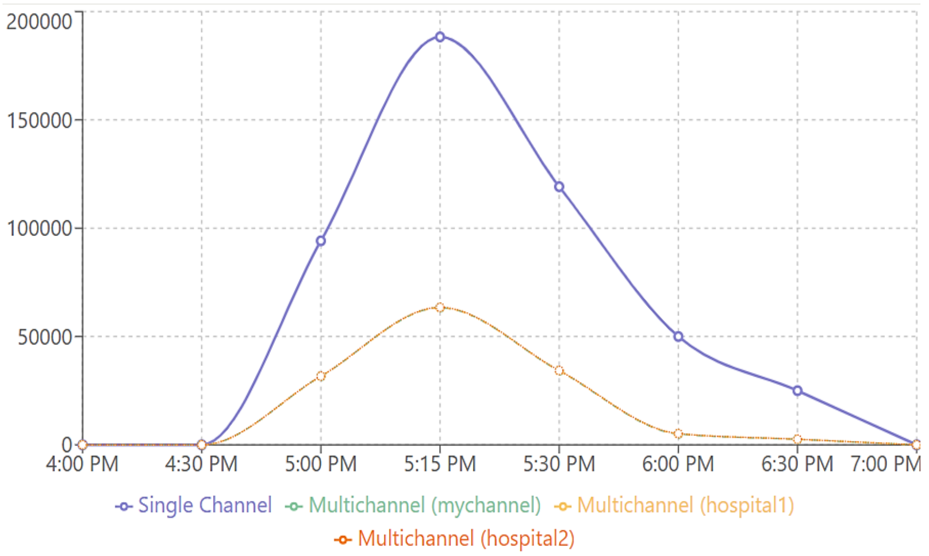


Figure 8: Transaction/Hour rate in a single-channel vs. multichannel Hyperledger network

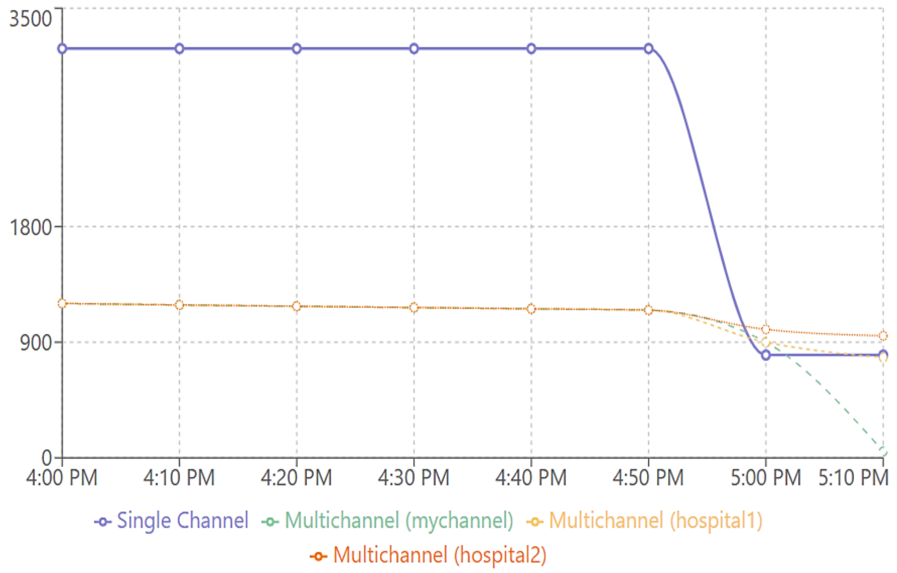


Figure 9: Transaction/Minute rate in a single-channel vs. multichannel Hyperledger network

The multichannel shows better performance over time. This is due to better load distribution and the ability to handle different types of transactions separately, making it a more scalable and flexible solution for many Hyperledger Fabric network use cases. Figure 10 shows the average response time and the total runtime for all organizations under single-channel and multichannel configurations.

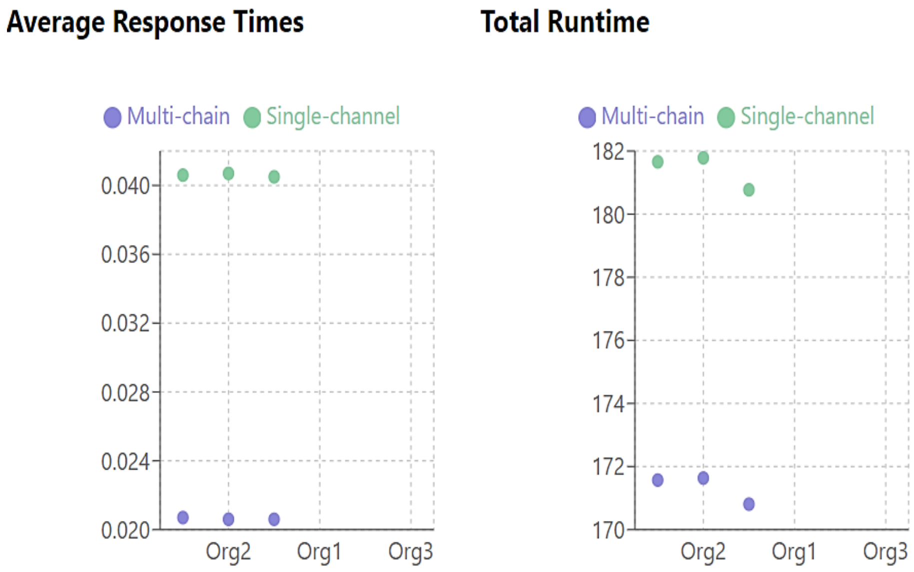


Figure 10: Average response time and total run time of single-channel and multi-channel configurations

Figures 11 and 12 show the performance outcomes of three chaincodes (reg-patient-operations, hospital1-operations, and hospital2-operations) in a proposed multichannel e-health system, evaluated using Hyperledger Caliper. Figure 11 shows the minimum and maximum latency vs. different send rates (25, 50, 100, and 200). Figure 12 shows the average latency and throughput vs. used send rates. Figures 13 and 14 show the CPU and memory usage per organization.

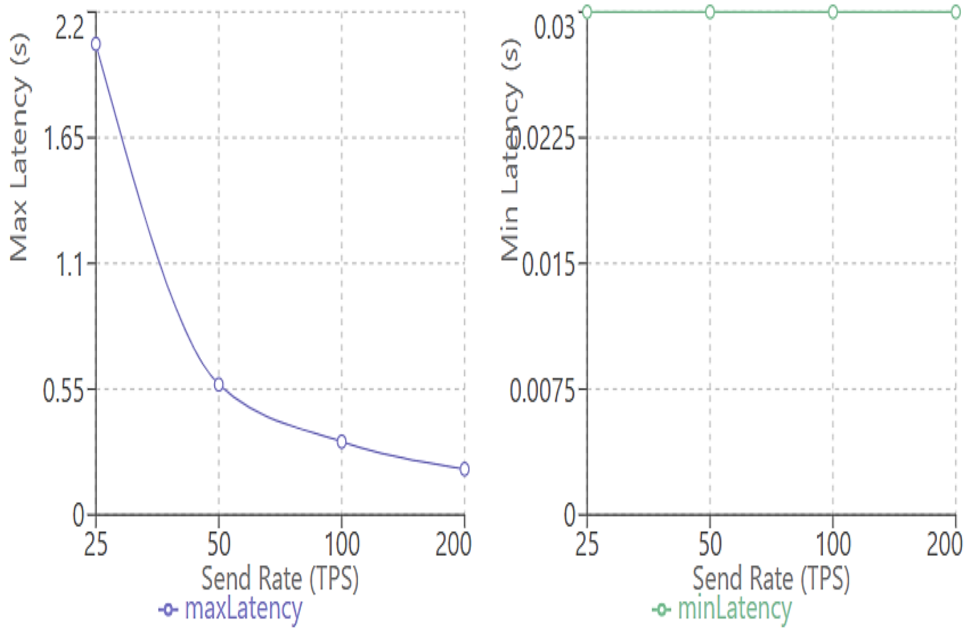


Figure 11: Minimum and maximum latency vs. send rates for the three organizations

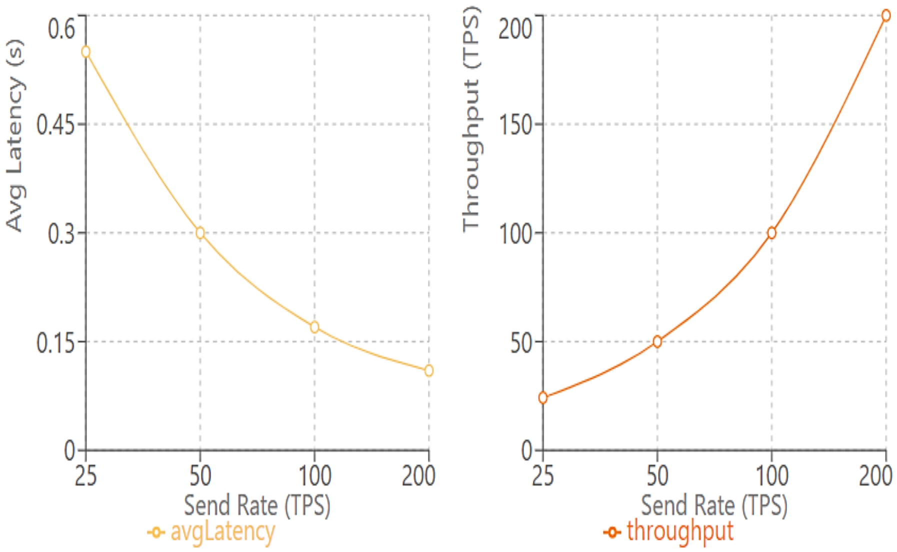


Figure 12: Average latency and the throughput vs. send rates of the three organizations

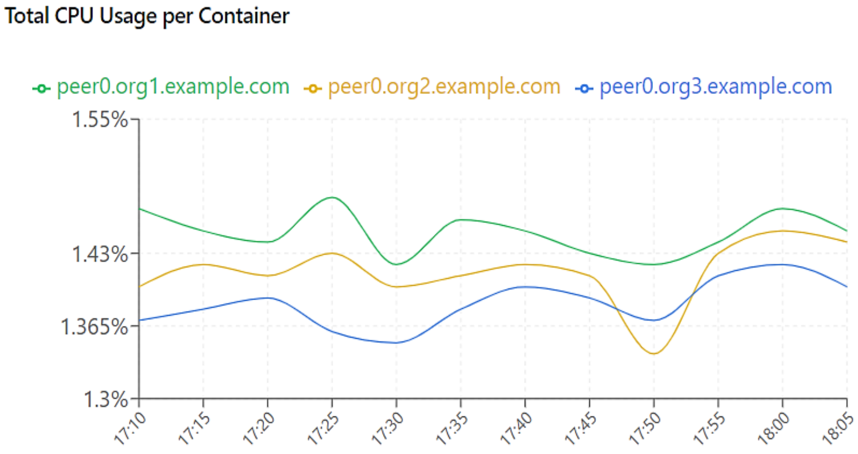


Figure 13: CPU usage per container

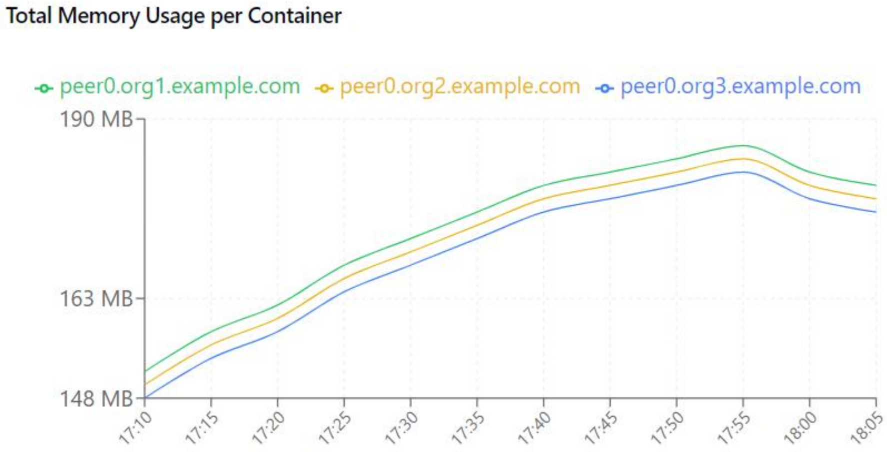


Figure 14: Memory usage per container

5.3 Discussion of Results and Comparison with the Related Work

Multichannels distribute the load across channels, reducing latency and increasing throughput. Single channel shows significantly higher peak values (~18,000) than multichannel (~6,000) for the blocks/hour metric. For blocks per minute, single-channel maintains a stable higher rate (~330 blocks/minute) than multichannel configurations (90-120 blocks/minute). Moreover, single-channel performance peaks at approximately 190,000 transactions per hour and consistently maintains a high rate of around 3,300 transactions per minute. In contrast, multichannel peaks at about 60,000 transactions per hour and shows a lower rate of around 1,200 transactions per minute. Hyperledger Caliper reported detailed performance outcomes of the proposed method. Maximum latency decreases significantly as the send rate increases from 25 to 200 TPS, dropping from 2.06s to 0.2s. Minimum latency remains low and stable (around 0.03s) across all send rates. The average latency consistently decreases from 0.55s to 0.11s as send rates increase. Throughput demonstrates positive scaling across all send rates. Table 3 compares some related work in terms of latency and throughput; the proposed method outperforms the related work.

Table 3: Comparison with some related work

Study	Send Rate (TPS)	# Work-ers	Succ	Max Latency (s)	Min Latency (s)	Avg. latency	Throughput (TPS)
[16]	25	_	_	_	_	_	_
	50	_	1000	_	_	26.22	27
	100	_	1000	_	_	31	53
	200	_	1000	_	_	36.22	101
[17]	25	_	_	_	_	_	_
	50	_	_	_	_	15	50
	100	_	_	_	_	20	100
	200	_	_	_	_	30	130
[20]	25	3	987	4.26	0.71	1.56	25
	50	3	993	4.7	0.74	2.1	48.5
	100	3	988	4.92	0.78	2.31	87.4
	200	3	997	5.03	0.85	3.12	164.4
Proposed method	25	2	1000	2.06	0.03	0.55	24.2
	50	4	1000	0.57	0.03	0.3	50
	100	4	1000	0.32	0.03	0.17	100
	200	4	1000	0.2	0.03	0.11	200

6. Conclusion and Future Work

The outcomes show that multichain systems have great scalability and safety. By distributing transactions across channels, they can manage resources better and work faster in high-load situations. On the other hand, a single-chain system may be less complicated or easier to handle. However, it could become congested when there are many transactions affecting efficiency and causing delays in processing. Based on these findings, companies should adopt multichain setups in their Hyperledger Fabric networks, especially within areas with high transactional volumes or if a volume increase is expected soon. Such designs improve performance and enable security and compliance controls to be tailored for different channels, thus making the network structurally stronger. Expanding the Hyperledger Fabric network to include many organizations and channels and utilizing cloud technologies is suggested as future work for scalability and ease of management. Utilizing cloud services can make it easier for systems to be deployed, scale dynamically, and manage resources efficiently, which is necessary to support wide-ranging healthcare establishments. This growth will enhance more than just scalability or interoperability; it will also encourage wider adoption while fostering collaboration among different e-health institutions, thus maximizing the benefits of blockchain technology in healthcare.

References

[1] I. M. Hasan and R. F. Ghani, “Blockchain for Authorized Access of Health Insurance IoT System,” *Iraqi Journal of Computers, Communications, Control and Systems Engineering.*, vol. 21, no. 3, pp. 76–88, Sep. 2021. Available: doi.org/10.33103/uot.ijccce.21.3.7.

[2] H. H. Saeed, H. Malik, U. Bashir, A. Ahmad, S. Riaz, M. Ilyas, W. A. Bukhari and M. I. Ali Khan, “Blockchain Technology in Healthcare: A Systematic Review,” *PLOS ONE*, vol. 17, no. 4, pp. 1-31, Apr 2022, Available: https://doi.org/10.1371/journal.pone.0266462.

[3] R. F. Ghani, Z. Kamal, and A. Farhan, “Blockchain-based E-Government system using WebSocket protocol,” *Engineering and Technology Journal*, vol. 42, no. 4, pp. 421–429, Apr. 2024. Available: https://doi.org/10.30684/etj.2024.146559.1689.

[4] R. F. Ghani, A. A. Salman, A. B. Khudhair, and L. Aljobouri, “Blockchain-based Student Certificate Management and System Sharing Using Hyperledger Fabric Platform,” *Periodicals of Engineering and Natural Sciences*, vol. 10, no. 2, p. 207, Apr. 2022. Available: https://doi.org/10.21533/pen.v10i2.2839.

[5] N. Radziwill, “Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World.,” *Quality Management Journal*, vol. 25, no. 1, pp. 64–65, Jan. 2018. Available: https://doi.org/10.1080/10686967.2018.1404373.

[6] E. Kane, “Is Blockchain a General Purpose Technology?,” *Economics of Networks eJournal*, 2017, Available: https://doi.org/10.2139/ssrn.2932585.

[7] A. Hasselgren, K. Kralevska, D. Gligoroski, S. A. Pedersen, and A. Faxvaag, “Blockchain in Healthcare and Health Sciences—A Scoping Review,” *International Journal of Medical Informatics*, vol. 134, no. 1, p. 104040, Feb. 2020. Available: https://doi.org/10.1016/j.ijmedinf.2019.104040.

[8] G. Drosatos and E. Kaldoudi, “Blockchain Applications in the Biomedical Domain: A Scoping Review,” *Computational and Structural Biotechnology Journal*, vol. 17, pp. 229–240, 2019. Available: https://doi.org/10.1016/j.csbj.2019.01.010.

[9] E. Chukwu and L. Garg, “A Systematic Review of Blockchain in Healthcare: Frameworks, Prototypes, and Implementations,” *IEEE Access*, vol. 8, pp. 21196–21214, 2020, Available: https://doi.org/10.1109/access.2020.2969881.

[10] C. C. Agbo, Q. H. Mahmoud and J. M. Eklund, “Blockchain Technology in Healthcare: A Systematic Review,” *Healthcare*, vol. 7, no. 2, p. 56, Apr 2019. Available: https://doi.org/10.3390/healthcare7020056.

[11] H. S. A. Fang, T. H. Tan, Y. F. C. Tan, and C. J. M. Tan, “Blockchain Personal Health Records: Systematic Review,” *J. Med. Internet Res.*, vol. 23, no. 4, p. e25094, Apr 2021, Available: https://doi.org/10.2196/25094.

[12] A. A. Siyal, A. Z. Junejo, M. Zawish, K. Ahmed, A. Khalil, and G. Soursou, “Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives,” *Cryptography*, vol. 3, no. 1, p. 3, Jan 2019, Available: https://doi.org/10.3390/cryptography3010003.

[13] A. Bhatti, A. A. Siyal, A. Mehdi, H. Shah, H. Kumar and M. A. Bohyo, "Development of Cost-Effective Tele-Monitoring System for Remote Area Patients," *International Conference on Engineering and Emerging Technologies (ICEET), Lahore, Pakistan*, 2018, pp. 1-7. Available: 10.1109/ICEET1.2018.8338646.

[14] M. S. Houston, J. D. Myers, S. P. Levens, M. T. McEvoy, S. A. Smith, B. K. Khandheria, W. K. Shen, M. E. Torchia, and D. J. Berry, “Clinical Consultations Using Store-and-Forward Telemedicine Technology,” *Mayo Clinic Proceedings*, vol. 74, no. 8, pp. 764–769, Aug 1999. Available: https://doi.org/10.4065/74.8.764.

[15] V. Mani, P. Manickam, Y. Alotaibi, S. Alghamdi, and O. I. Khalaf, “Hyperledger Healthchain: Patient-Centric IPFS-Based Storage of Health Records,” *Electronics*, vol. 10, no. 23, p. 3003, Dec 2021. Available: https://doi.org/10.3390/electronics10233003.

[16] A. P. Singh et al., “A Novel Patient-Centric Architectural Framework for Blockchain-Enabled Healthcare Applications,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020. Available: https://doi.org/10.1109/TII.2020.3037889.

[17] A. Panwar, V. Bhatnagar, M. Khari, A. W. Salehi, and G. Gupta, “A Blockchain Framework to Secure Personal Health Record (PHR) in IBM Cloud-Based Data Lake,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–19, Apr. 2022. Available: https://doi.org/10.1155/2022/3045107.

[18] V. Thakkar and V. Shah, “A Privacy-Preserving Framework Using Hyperledger Fabric for EHR Sharing Applications,” *International Journal of Electrical and Computer Engineering Systems*, vol. 14, no. 6, pp. 667–676, Jul. 2023. Available: https://doi.org/10.32985/ijeces.14.6.6.

[19] R. A. Abutaleb, S. S. Alqahtany, and T. A. Syed, “Integrity and Privacy-Aware, Patient-Centric Health Record Access Control Framework Using a Blockchain,” *Applied Sciences*, vol. 13, no. 2, p. 1028, Jan 2023. Available: https://doi.org/10.3390/app13021028.

[20] Á. Díaz and H. Kaschel, “Scalable Electronic Health Record Management System Using a Dual-Channel Blockchain Hyperledger Fabric,” *Systems*, vol. 11, no. 7, pp. 346–346, Jul 2023. Available: https://doi.org/10.3390/systems11070346.

[21] R. A. Jaafar and S. N. Alsaad, “Enhancing Educational Certificate Verification with Blockchain and IPFS: A Decentralized Approach Using Hyperledger Fabric,” *TEM Journal*, vol. 12, no. 4, pp. 2385–2395, Nov 2023. Available: https://doi.org/10.18421/tem124-51.

[22] L. Abdelgalil and M. Mejri, “HealthBlock: A Framework for a Collaborative Sharing of Electronic Health Records Based on Blockchain,” *Future Internet*, vol. 15, no. 3, p. 87, Mar 2023. Available: https://doi.org/10.3390/fi15030087.

[23] M. N. M. Bhutta, A. A. Khwaja, A. Nadeem, H. F. Ahmad, M. K. Khan, M. A. Hanif, H. Song, M. Alshamari and Y. Cao, “A Survey on Blockchain Technology: Evolution, Architecture and Security,” *IEEE Access*, vol. 9, pp. 61048–61073, 2021. Available: https://doi.org/10.1109/access.2021.3072849.

[24] H. Guo and X. Yu, “A Survey on Blockchain Technology and Its Security,” *Blockchain: Research and Applications*, vol. 3, no. 2, p. 100067, Feb 2022. Available: https://doi.org/10.1016/j.bcra.2022.100067.

[25] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A survey on The Security of Blockchain Systems,” *Future Generation Computer Systems*, vol. 107, pp.841-853, Aug 2017. Available: https://doi.org/10.1016/j.future.2017.08.020.

[26] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu “A Survey of Blockchain Technology Applied to Smart Cities: Research Issues and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2794–2830, 2019. Available: https://doi.org/10.1109/comst.2019.2899617.

[27] A. Iftekhar, X. Cui, Q. Tao, and C. Zheng, “Hyperledger Fabric Access Control System for Internet of Things Layer in Blockchain-Based Applications,” *Entropy*, vol. 23, no. 8, p. 1054, Aug 2021. Available: https://doi.org/10.3390/e23081054.

[28] D. Khan, L. T. Jung, M. A. Hashmani, and M. K. Cheong, “Empirical Performance Analysis of Hyperledger LTS for Small and Medium Enterprises,” *Sensors*, vol. 22, no. 3, p. 915, Jan 2022. Available: https://doi.org/10.3390/s22030915.

[29] J.-J. Lin, Y.-T. Lee, and J.-L. Wu, “The Effect of Thickness-Based Dynamic Matching Mechanism on a Hyperledger Fabric-Based TimeBank System,” *Future Internet*, vol. 13, no. 3, p. 65, Mar 2021. Available: https://doi.org/10.3390/fi13030065.

[30] Y. Liu, J. Zhang, S. Wu, and M. S. Pathan, “Research on Digital Copyright Protection Based on The Hyperledger Fabric Blockchain Network Technology,” *PeerJ Computer Science*, vol. 7, p. e709, Sep 2021. Available: https://doi.org/10.7717/peerj-cs.709.

[31] M. Uddin, “Blockchain Medledger: Hyperledger Fabric Enabled Drug Traceability System for Counterfeit Drugs in Pharmaceutical Industry,” *International Journal of Pharmaceutics*, vol. 597, p. 120235, Mar 2021. Available: https://doi.org/10.1016/j.ijpharm.2021.120235.

[32] A. Lohachab, S. Garg, B. H. Kang, and M. B. Amin, “Performance Evaluation of Hyperledger Fabric-Enabled Framework for Pervasive Peer-To-Peer Energy Trading in Smart Cyber-Physical Systems,” *Future Generation Computer Systems*, vol. 118, pp. 392–416, May 2021. Available: https://doi.org/10.1016/j.future.2021.01.023.

[33] A. M. Alchaqmaqchee and S. N. Alsaad, “Design Scheme for Copyright Management System Using Blockchain and IPFS,” *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 613–618, May 2021. Available: https://doi.org/10.12785/ijcds/100159.