

ORIGINAL RESEARCH

Scalability Performance Analysis of Blockchain Using Hierarchical Model in Healthcare

Lipsa Sadath, MSc, MCA¹, Deepti Mehrotra, PhD², Anand Kumar, PhD³

¹Computer Science Engineering Department, School of Engineering, Amity University, Dubai, UAE; ²Computer Science Engineering Department, Amity School of Engineering and Technology, Amity University; Uttar Pradesh, Noida, India; ³Electronics Engineering Department, School of Engineering, Amity University, Dubai, UAE

Corresponding Author: Lipsa Sadath; Email: lsadath@amitydubai.ae

DOI: <https://doi.org/10.30953/bhty.v7.295>

Keywords: blockchain technology, database management, hyperledger, Hyperledger Caliper, network performance, privacy, security, scalability

Abstract

Blockchain technology has become crucial in improving the privacy and security of enterprise applications in the cyber world. However, scalability has become a significant concern for researchers in large organizations, especially those with complex hierarchies and access privileges. As a result, the existing models and consensus algorithms suffer from various issues. Medical centers and healthcare providers are particularly affected by this problem due to the vast amount of data, making it a critical weakness of traditional database management systems. To address this issue, the authors propose a hierarchical model within the Hyperledger Fabric enterprise application, focusing on the healthcare sector as a use case. This model includes multiple organizations at different levels of the hierarchy, such as hospitals, hospital governance, and insurance companies. The initial implementation of this model includes two levels of hierarchy, demonstrating networks of hospitals joining an insurance company. The primary objective of the experiment is to test and improve the network's performance using this model. The model's performance is evaluated by manipulating and scaling environmental factors such as the number of organizations, transaction numbers, channels, block intervals, and block sizes. The benchmarking tool used for this assessment is Hyperledger Caliper, which measures indicators such as success and failure rates, throughput, and latency. Currently, the research focuses only on testing the model's scalability using patient data.

Submitted: December 5, 2023; Accepted: January 20, 2024; Published: April 17, 2024

The healthcare sector suffers from data privacy, security, and integrity issues as each patient's data flows through different hospitals under a specific hospital network (HN). The biggest challenge is faced when insurance claims are rejected due to mismatches in data. The prime reason could be tampered data submitted to insurance companies by the HNs. Such database registries can seriously affect claims submitted to insurance companies.

When a private organization's application works with blockchain, a guarantee for better transparency and security exists, but issues related to scalability and latency arise. This is seen mainly in supply chain systems, healthcare applications, etc. One of the main reasons is the

number of organizations participating in transactions and different contracts running through the network. Many enterprise applications are considered sensitive to network delays because of the latency caused by the vast number of transactions. Current studies revolve around changing the batch time or the network's block size in a permissioned environment. This problem in the healthcare sector is more complex and requires more attention due to substantial patient data.

The primary interest of research in this paper is to understand scalability performance and create a model in the healthcare sector using Hyperledger Fabric enterprise-grade open-source on the Linux platform. The contributions of the research are listed here:

- A performance analysis uses a linear model of Hyperledger Fabric with varying parameters and fixed rate controllers.
- Performance analysis is used to analyze the effect of more channels to compare the difference between single-channel and multi-channel networks with rate controllers.
- The hierarchical model analyzes the average latency and throughput of the model, and how they overcome the effect with multiple organizations, clients, and channels.
- A sequence of experiments shows the results of the model with changes in block sizes and block intervals.
- Performance and bottlenecks are identified to evaluate the capacity of the model.

The flow diagram in Figure 1 shows the order of topics discussed. A glossary of key terms used throughout the article is presented in the Appendix.

Blockchain Technology (BCT): Security, Privacy, Scalability

Security and Privacy

Blockchain combines technological features, including cryptography, peer-to-peer networking, distributed system, transparency, access identity permissions, open source, autonomy, and immutability. This makes the technology secure as the transactions cannot be tweaked or tampered with, and all the entities/parties that make transactions are kept anonymous.¹ Thus, blockchain offers integrity, protected distributed ledgers, and transparency of the entire process by upholding a set of global states.

The participating nodes agree upon the existence, value, and histories of all states. Each state contains multiple transactions. Hence, blockchain is a way to manage distributed transactions. The concerned entities preserve replicas of the data and jointly agree on a transaction execution

order. This high-performance blockchain is a means to monitor a system completely as if under the surveillance of a legal third party. This is very important, especially in the healthcare systems, as the amount of patient data that floats from laboratories to insurance claims is immense. This data transparency is required to understand genuine patient claims. The network does not provide an opportunity for data tampering by any organization.

Scalability Issues

At the same time, concerns about data replication² exist as immutable records are stored with every peer. This affects the throughput of the blockchain network.³ Many frameworks are designed to handle this at production levels and industry applications. However, research is underway to understand the effect of different consensus mechanisms and their role in maintaining the immutability, stability, privacy, and security of the system.⁴

Block size causes performance issues when changed, resulting in a change in the maximum message counts the blocks can hold.⁵ Whereas, the scalability use case in the Ethereum network discusses sharding (the process of separating large databases into smaller, faster, more easily managed parts)⁶ as a solution that reduces the load in the entire network.

Machine learning through proof-of-information consensus is discussed by Kuo et al.,⁷ but the model lacks a discussion on scalability. In contrast, a model by Zhang et al.⁸ discusses data transparency between owners and users of clinical data addresses and, to an extent, scalability and security issues. Another model published by Ylonen and Lonvick⁹ that discussed scalability has an architecture with the Secure Shell Protocol (SSH)⁹ that has an identity preservation method with an absolute model view controller pattern with pointers for data access from the organization's data pool. Smart contracts in web and data immutability are the highlight in a genomic study by Glicksberg et al.¹⁰ that attempts to understand the identification of late cancer stages using blockchain, but they lacked clear scalability research evidence. Other research by Lee et al.¹¹ using blockchain to handle patient records studied data sensitivity, privacy, integrity, and authorization using blockchain technology, but it didn't address scalability issues. This challenge of scalability issues exists, especially when dealing with large clinical data records,¹² and other types of records that need privacy and consistency. All research speaks about careful data handling with reliable smart contracts.

Hyperledger Fabric Framework

The Hyperledger Fabric framework has peer machines with the same data. Technically, there is no administrator in a blockchain system, but enterprise applications already have authenticated peers. The blocks hold the data, which

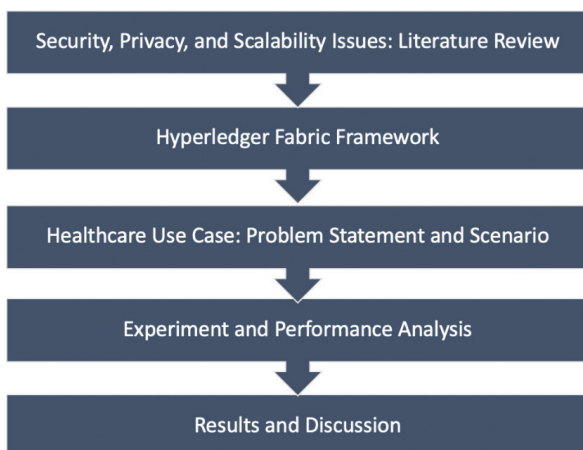


Fig. 1. Process flow diagram.

are otherwise known as immutable ledgers. The respective chaincode, or the smart contract, is common for a particular channel and the peers who join that transaction. The world state stores the other updates of the block data.¹³ Scalability issues are discussed in various models, like the hierarchical model through various abstract models.¹⁴⁻²⁰

This basic network architecture is shown in Figure 2, where the organizations are part of the data layer, which includes the fabric. The client application part is the business layer, which interacts with the fabric blockchain and hosts the application for the user. The Membership Service Provider (MSP) are credentials used by the peers (P) and committers to participate in the Hyperledger Fabric network as they are the organization's identity. The genesis block contains all the MSPs and the policies. Clients authenticate transactions, and peers authenticate the results or endorse them using these credentials. The orderers have a shared communication channel with peers and clients.

The broadcast service for messages and transactions takes place here. For scalability, they are implemented as docker containers. The peers maintain the read/write operations. The ordering services are collective nodes that order transactions into a block. The service is common for the overall network and supports pluggable implementations. Each member has a cryptographic identity material that is contained within orderers.

Whenever the client has a transaction request (Figure 3), an endorsement by the respective endorsers takes place according to the policy. Then, the orderer orders the block and copies the committing peers so all peers get a copy of the latest transaction.

Healthcare Use Case

This section describes the use case in a healthcare sector that involves an insurance company (IC) handling patient data from multiple HNs.

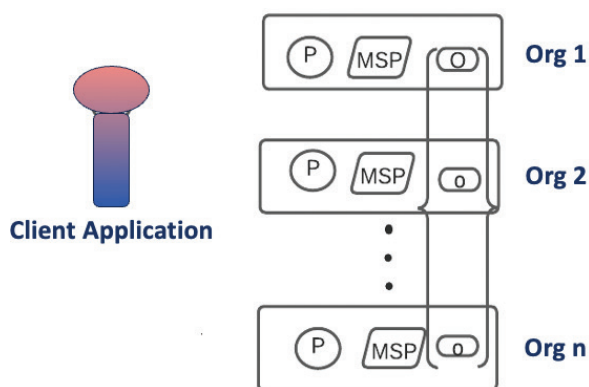


Fig. 2. Hyperledger Fabric Network Architecture. MSP: Membership Service Provider; O: orderers; Org: organization 1,2,...,n; P: Peers.

Statement of the Problem

When ICs need data on patients during a claim, there is a lot of missing information or gaps in the data provided by hospitals. This is due to the traditional relational database management system used to store information in hospitals and related networks. Data tampering and human-caused data errors are common problems in such situations. Most literature thus far has conceptual ideas that could be implemented in the blockchain. Another related issue is scalability as more organizations or hospitals join the network. Certain experiments in healthcare²¹ have only a very basic structure of blockchain implementation with a maximum of two medical organizations joining a single channel. Therefore, there is no proof of absolute latency test performed in networks implemented in this sector.

Hence, the main aim of this paper is to design a model to test the integration of medical records between insurance companies and a large network of hospital groups where patient data mount quickly.

Use Case Scenario

The key terminologies of the fabric framework are described through the use case. The experiment consists of five organizations with four HNs (HN1, HN2, HN3, HN4) and one IC (Figure 4). The benchmarking used for the tests is in Hyperledger caliper. Ideally, many data transfers take place between the insurance companies and the hospitals independently.

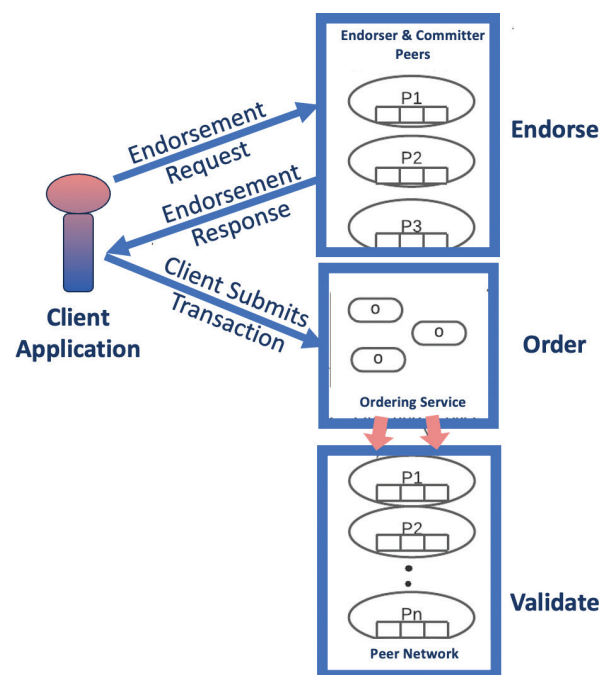


Fig. 3. Hyperledger Transaction Flow. MSP: Membership Service Provider; O: orderers; P: peers (1, 2,...,n).

1. Node: IC and each HN is considered as a node where each node joins for transactions through a peer in the network. They are also endorsers of the policy for the transactions. Each peer maintains their respective ledger after every transaction. We consider the one HN as a group that has data from their hospital branches (e.g.: Chicago, Dubai etc.)
2. Chaincode: The chaincode or the smart contract contains the business logic between the IC and the HN. There are four chaincodes or smart contracts that are deployed separately for each channel that joins the IC.

We consider the basic details of the patients that are needed from hospitals by insurance companies in the following way (Figure 5). The basic data structure

consists of patient ID, department, age, patient name, address, phone number, and bill for services.

For ease of use and populating the data, the chaincodes are renamed for deploying into the channels. The patient data are duplicated (Figure 6) to handle up to 15,000 transactions at times to check the scalability and performance of the network.

3. Endorsement Policy: Only a few peers are eligible to endorse the transaction proposal. The endorsement policy specifies the required peers for endorsement. In our business logic, we have implemented “OR” endorsement where not all peers are required to endorse the request. Figure 7 shows that policy as either Org1 or Org3 can be endorsed through Channel 2.

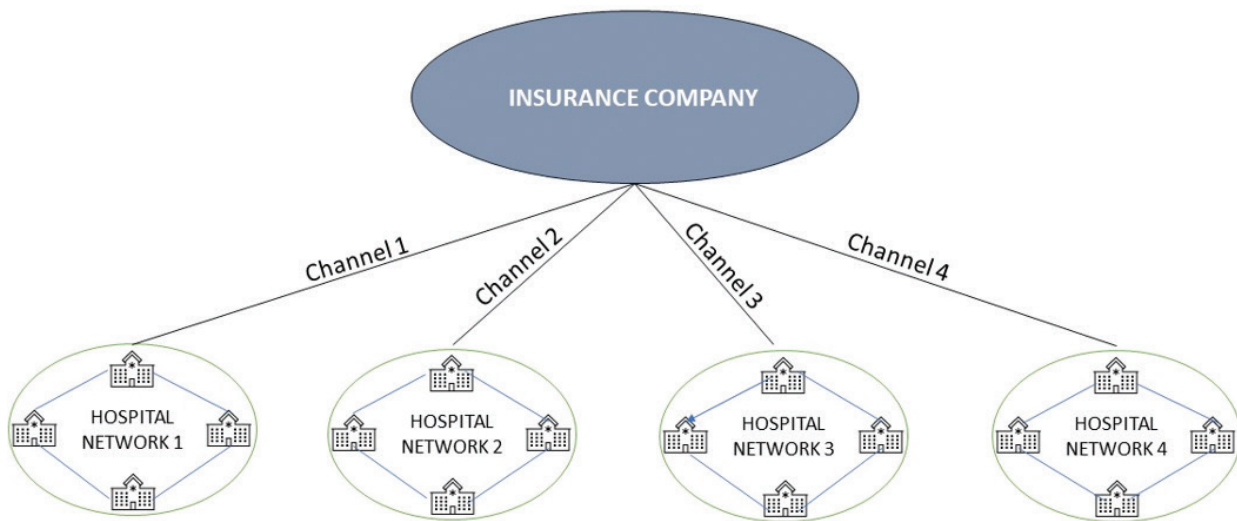


Fig. 4. Healthcare sector use case diagram.

```
asset := Patient {
ID: id,
Department: department,
Age: age,
Name: name,
Address: address,
PhoneNumber: phoneNumber,
BillAmount: billAmount,
}
```

Fig. 5. Presentation of patient data structure in smart contract.

4. Channel: Four channels connect the peers of the five organizations (the IC and the four HNs). Each channel has a chaincode deployed on it. Hence, each channel is governed by the respective pre-agreed business logic or policy as per the smart contract. Then, the ledger functions are initiated on these smart contracts. When a specific organization wants to conduct business transactions privately, Its peers join the channels separately.
5. Ordering Service: The model uses three orderers for ordering services to give more capacity to the network. This is to prevent any unforeseen situation where the ordering service of one orderer stops. In such situations other orderers take over to avoid any delay in the network.


```

const departments = ["Xray","BloodCheck","Consultation","MRI","Inpatient","Outpatient"];
const ages = [23,29,22,34,21,11,24,5,32,34,54,43,22,23,24,25,65,61,62];
const names = ["Tomoko","Bradd","Jin","Max","Adriana","Michel"];
const phoneNumbers = ["9999999999","9999449999", "9999999555", "9999999991","2999999999", "6599999999", "4499999999"];
const addresses = ["Dubai", "Delhi", "Banglore", "Mumbai", "Kolkata", "Tokyo"];
const billAmounts = [234,2345,35,4456,456,36,234,467,567,678,678,678,678,476,456,456,346345,345,3454]

```

Fig. 6. Presentation of data types and data in smart contract.

```

$ deployChaincode2.sh
41 CHANNEL_NAME="mychannel2"
42 CC_RUNTIME_LANGUAGE="golang"
43 VERSION="1"
44 SEQUENCE="1"
45 CC_SRC_PATH="./artifacts/src/github.com/contract2"
46 CC_NAME="contract2"
47 CC_POLICY="OR('Org1MSP.member','Org3MSP.member')"
48
49 packageChaincode() {
50     rm -rf ${CC_NAME}.tar.gz
51     setGlobalsForPeer0Org1
52     peer lifecycle chaincode package ${CC_NAME}.tar.gz \
53         --path ${CC_SRC_PATH} --lang ${CC_RUNTIME_LANGUAGE} \
54         --label ${CC_NAME}_${VERSION}
55     echo "===== Chaincode is packaged ===== "

```

Fig. 7. Presentation of endorsement policy in smart contract.

6. Certificate Authority (CA): Each peer of an organization (IC and HNs) will have their certificate authority. It is the certificate authority service that is first up in the network to establish all the organizations.²⁵
7. Ledger: The data is stored in CouchDB for the respective organizations. According to the GDPR (General Data Protection Rule)²⁵ personal data should not be revealed in transactions, so the advantage of blockchain is that it is the hash of the data that is stored in the blocks. Other data that the transactions can pass on are the treatment and diagnostic details of the patient that need to be checked by the IC.
8. Workers: The current experiment considers workers as the clients. The clients can be from the IC or any of the HNs. They could be doctors or agents of insurance companies, agents who try to create a transaction, or read a transaction as the business logic or the endorsement policy.
9. Fabric Gateway: They are a set of libraries that help to invoke the business transaction through smart contracts and the peers for endorsement. After endorsement, the transaction is saved and distributed to other peers through the fabric gateway and the channels.

Scalability in Healthcare Use Case

There are two ways to handle scalability: vertical scaling and horizontal scaling. In horizontal scaling, the load is distributed at peak hours to different temporary

servers from the application process interfaces, whereas in vertical scaling, we assume that the capacity of the network must be increased as more organizations join and more transactions are incorporated (TPS or transactions per second). However, here in this experiment, we use vertical scaling to understand the performance of the network.

Experiment and Performance Analysis

Experimental Setup

The fabric blockchain test bed is set up with five organizations carrying one peer each. The Raft consensus algorithm is used with three orderers to avoid any delay even when one orderer is down. The endorsement policy requires at least one peer to be an endorser from each organization. The network is set up on a virtual machine with Linux Ubuntu 20.04 with 6 CPU cores at different instances to check the performance with 16 GB RAM.

To check scalability and performance, our fabric experiment focuses on the number of organizations, channels, number of TPS, number of clients or workers, block intervals, and block sizes checked against the latency and throughput (number of successful transactions). The benchmarking tool used is Hyperledger Caliper, which calculates the time to create and read the transactions. Most experiments^{22,23} consider the latency analysis using the transaction arrival rates using Hyperledger Caliper as the benchmarking tool.

Channel Distribution in the Network

Table 1 lists the channel distribution in the network with the IC and HN (HN1, HN2, HN3, HN4).

Bench Marking Performance

The performance of the network is measured by varying one parameter as a rate controller and keeping other parameters constant. This is done to understand the maximum capacity of the network. The experiment is benchmarked against two popular experiments by Xu et al.²² and Al-Sumaidae et al. Al-Sumaidae et al.²¹ have performed a Hyperledger Fabric experiment using the concept of just two medical institutions and performed latency analysis of the network.

The experiment did not consider features such as the inclusion of more than two organizations in the network, varying the number of channels, or varying other parameters like block size and block intervals. The major benchmarking we followed from Al-Sumaidae et al.²¹ is mainly varying the number of clients or workers, and steadily increasing the number of transactions and TPS to analyze the network capacity. Additionally, our experiment follows other criteria set by Xu et al.²² to check latency in terms of varying number of channels, block size, and block interval.

The benchmarking is performed using Hyperledger Caliper, which gives four performance indicators in terms of TPS. The indicators include the success and failure rates of transactions, latency (average time taken to complete the response),

Table 1. Presentation of channel distribution details in the network according to Figure 4

Channel Number	Channel Distribution
1	IC- HN1
2	IC- HN2
3	IC-HN3
4	IC-HN4

IC: insurance company; HN: hospital network.

Table 2. Experiment 1: Changing Transaction Numbers: Create operation

Tx (n)	Channels (n)	Transactions (per second)	Workers (n)	Throughput	Latency (ms)
1000	1	75	5	72.5	0.56
5000	1	75	5	74.7	0.56
15,000	1	75	5	74.8	0.58

Tx: transactions.

Table 3. Experiment 1: Changing Transaction Numbers: Read operations

Tx (n)	Channels (n)	Transactions (per second)	Workers (n)	Throughput	Latency (ms)
1000	1	75	5	74.5	0.01
5000	1	75	5	74.9	0.01
15,000	1	75	5	75	0.01

Tx: transactions.

and throughput (average number of transactions/second). This section has the complete experimental setup and analysis with varying parameters. The readings are shown in table format, and the results are compared with Al-Sumaidae et al.²¹ for the first three experiments. From Experiment 4 onwards, we do not compare the results with any benchmarking as we are observing our own implementation with five organizations. Each parameter impact is shown below through analysis graphs as well. Each experiment is performed at least three times, and an average is taken to get a final reading

Results and Discussion

The performance of the network shall be measured in terms of three indicators such as latency, throughput, and send rates. We record the latency, throughput and send rates for creating and reading records. In the below experiments, first to test the network, we keep one, that is, Channel 1—between IC and HN1 connected to test the impact of other parameters in the network. Each experiment that we perform is mentioned with fixed and variable rate controllers, and recorded for their latency, throughput, and send rates to measure the success and failure of the network.

Rate Controllers

In the first set of three experiments, we fixed the Block Interval = 1s and Block Size = 50, Number of Channels = 1, TPS = 75, Workers = 5. From Experiment 1 to Experiment 3, we have only one channel with variable rate controllers such as transaction numbers, number of workers, and the TPS. For the remaining experiments, we increase the number of channels to understand the channel impact along with TPS, block size, and block interval.

Experiment 1: Impact of Transaction Numbers

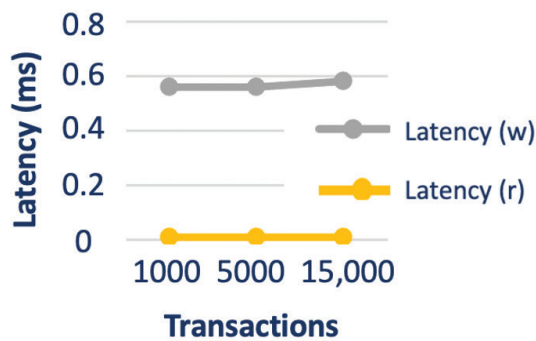
Only transaction numbers (Table 2 for create operations and Table 3 for read operations) are changed as variable rate controller with five clients or end users as workers.

Observations

The latency is high for create operation. It is observed that both the throughput and latency are higher as the transaction number increases (Figure 8).

Benchmarking

Benchmarking is against experiments conducted using a similar platform by Al-Sumaidae et al.²¹ with similar indicators but with two organizations only. It is observed that when the number of organizations increases, throughput and latency (Figure 8) are affected more when compared to results shown in Table 4.



Experiment 2: Impact of Number of Workers

The variable rate controller is the number of workers or clients that join to process the transactions. Table 5 shows create operations, and Table 6 shows read operations.

Observations

There is a steady decrease in throughput and latency as the number of clients or workers increase for both create and read operations (Figure 9). This is a typical situation when end users are more at the network, and access write and read operations are requested by all 5 to 50 clients at the same time.

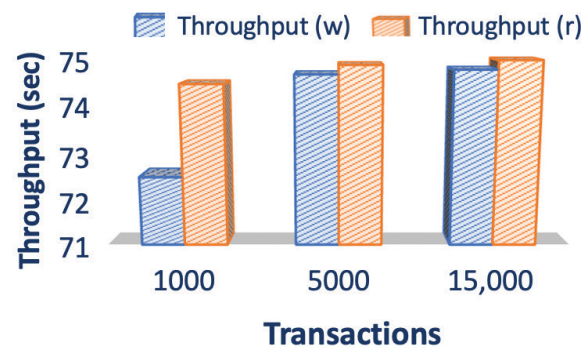


Fig. 8. Evaluation of the impact of transaction numbers, TPS (transactions per second) = 75 from Tables 2 and 3, Org (organization) numbers = 5: Latency and Throughput in Create and Write operations.

Table 4. Evaluation of the impact of transaction numbers.²¹ Org (organization) numbers = 2

Create Record					
Workers (n)	txNumber	Transactions (per second)	Latency (ms)	Throughput	Send Rate
5	1000	75	0.1	75.1	75.4
5	5000	75	0.09	75	75.1
5	15,000	75	0.09	75.1	75

TxNumber: The total number of transactions that must be sent.

Table 5. Changing transaction numbers of workers: Create operations

Tx (n)	Channels (n)	Transactions (per second)	Workers	Throughput	Latency (ms)
1000	1	75	5	72.2	0.56
1000	1	75	25	64.0	0.99
1000	1	75	50	43.6	2.25

Tx: transactions.

Table 6. Changing Transaction Numbers of workers: Read operations

Tx (n)	Channels (n)	Transactions (per second)	Workers	Throughput	Latency (ms)
1000	1	75	5	74.8	0.01
1000	1	75	25	67.4	0.01
1000	1	75	50	56	0.01

Tx: transactions.

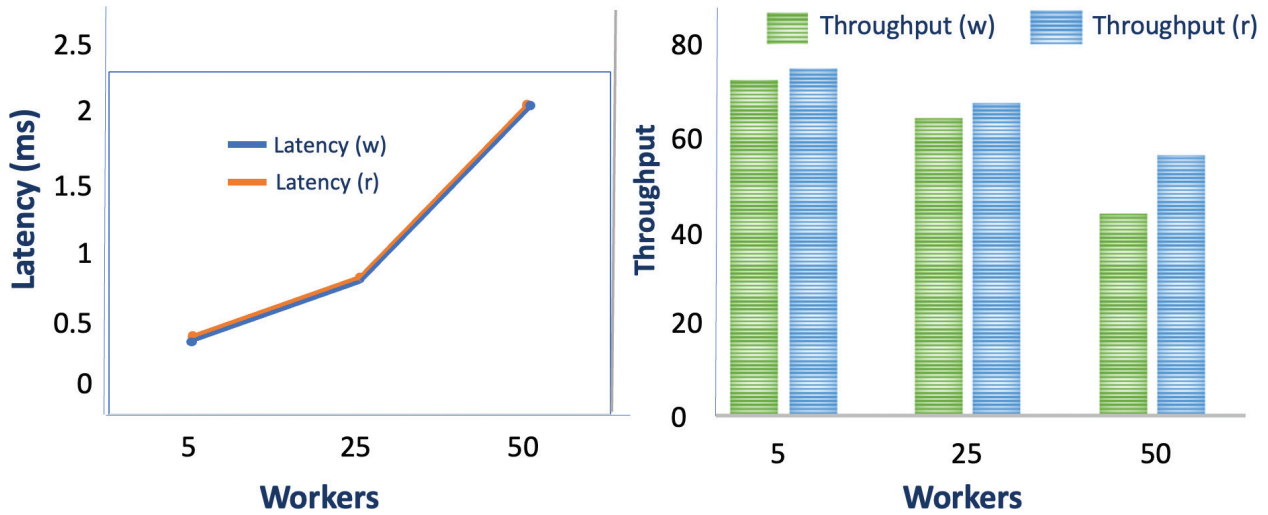


Fig. 9. Evaluation of impact of transaction numbers, TPS = 75 from Tables 5 and 6, Org numbers = 5, Latency and Throughput in Create and Read operations.

Table 7. Evaluation of the impact of workers,²¹ Org numbers = 2

Create Record					
Workers (n)	txNumber	Transactions (per second)	Latency (ms)	Throughput	Send Rate (n)
5	1000	75	0.1	75.1	75.4
25	1000	75	0.11	75.6	75.8
50	1000	75	0.76	76.4	76.7

TxNumber: The total number of transactions that must be sent.

Table 8. Changing the transactions per second: Create operations

Tx (n)	Channels (n)	Transactions (per second)	Send Rate (n)	Workers (n)	Throughput	Latency (ms)
1000	1	75	73.3	5	72.4	0.60
1000	1	150	140.3	5	122.5	1.83
1000	1	250	190.3	5	119.0	3.41

Tx: transactions.

Benchmarking

We compare our results with experiments performed by Al-Sumaidae et al.,²¹ where the number of organizations (Table 7) is just two, whereas our network supports five organizations. Hence, the resources and services are shared between all the five organizations compared to two organizations. It is observed that the latency and throughput of our experiment are severely affected due to the greater number of organizations and workers.

Experiment 3: Impact of Increase in Transactions Per Second

From Experiment 1 and Experiment 2, we found that throughput and latency are affected as more number of organizations join the network. Experiment 3 is conducted by increasing the TPS and keeping other parameters

steady. Send Rate is referred to as the number of transactions that are actually sent, though we set the TPS to a few values as 75, 150, and 250, as in Tables 8 and 9 during the experiment.

Observations

As the TPS increases, the latency increases (Figure 10) mainly for Write operations. Read operations give a good throughput and less latency which is almost negligible compared to the write.

Benchmarking

The experiment is benchmarked against Al-Sumaidae et al.²¹ and it is observed that the throughput and send rate are less when the number of organizations increase compared to the

two organizations in the benchmarked experiment. txNumber is the total number of transactions that must be sent.

The throughput in transaction creations fell drastically compared to the benchmarking Al-Sumaidae et al. (Table 10).²¹

Experiment 4: Impact of Channel Numbers (TPS = 75)

Here, the rate controller is set as the number of channels. We add more channels to make sure step by step, the IC

communicates in parallel with HN1, HN2, HN3, and HN4, respectively. For this, we do not keep any benchmarking with Al-Sumaidae et al.²¹ as they do not have multiple channels. From Experiment 4 onwards, we escalate the experiment (Tables 11 and 12) to check more performance according to our experiment model only.

Observations

Though all readings were taken each time a channel joined the IC, we observed the difference in the performance of Channel

Table 9. Changing transactions: Read operations

Tx (n)	Channels (n)	Transactions (per second)	Send rate (n)	Workers (n)	Throughput	Latency (ms)
1000	1	75	73.3	5	74.6	0.01
1000	1	150	140.3	5	146.7	0.01
1000	1	250	190.3	5	234.9	0.01

Tx: transactions.

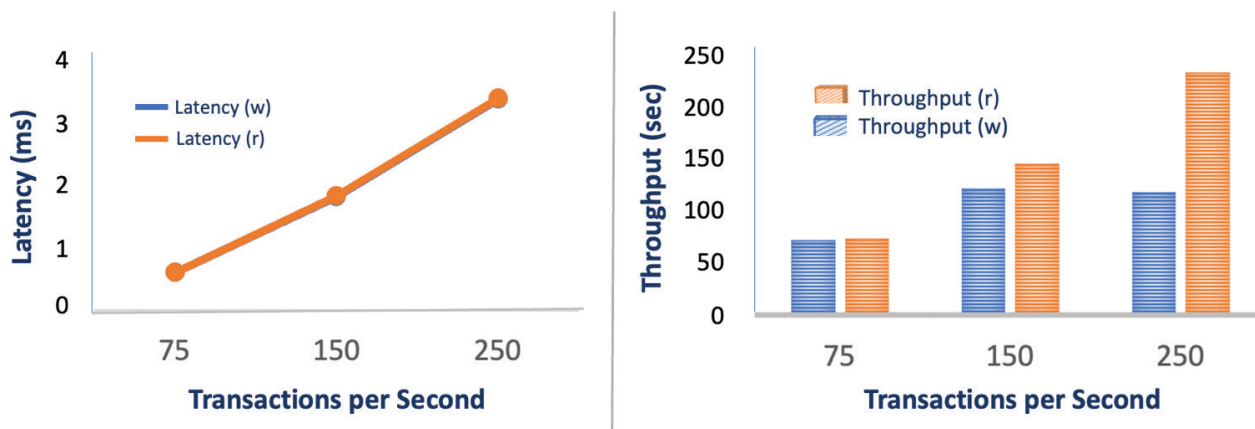


Fig. 10. Evaluation of impact of transaction numbers, Tx = 1000 from Tables 8 and 9, Org numbers = 5, Latency and Throughput in Create and Read operations.

Table 10. Evaluation of transactions per second increase,²¹ Org numbers = 2

Transactions (per second)	txNumber	Workers (n)	Latency (ms)	Throughput	Send Rate (n)
75	1000	5	0.1	75.1	75.4
150	1000	5	0.08	149.6	150.8
250	1000	5	0.1	164.6	251.1

TxNumber: The total number of transactions that must be sent.

Table 11. Changing the number of channels: Create operations

Tx (n)	Channels (n)	Transactions (per second)	Send rate (n)	Workers (n)	Throughput	Latency (ms)
5000	1	75	74.6	5	74.4	0.67
5000	2	75	72.9	5	49.0	28.24
5000	3	75	74.8	5	74.6	0.65
5000	4	75	74.8	5	74.5	0.70

TxNumber: The total number of transactions that must be sent.

1 each time for Create (Table 11) and Read (Table 12). In this scenario, we try to check if the addition of more channels affects the performance of Channel 1, though any channel can go slow in the process. It is observed that as Channel 2 [HN2] joined to IC, there is a sudden latency (28.24 ms) experienced in the network (Table 11) and a fall in throughput to 49.0. This could be due to several clients' requests being processed at the same time and the network sharing the resources parallelly all of a sudden. As always, the latency is higher in Create than in Read operations.

Experiment 5: Impact of Channel Numbers (With TPS = 100)

The next effort is to understand how the addition of channels affect the network as TPS changes to 100 from 75 (Experiment 4). We conduct the experiment for both create (Table 13) and read (Table 14) operations.

Observations

Similar to the pattern that is observed for TPS = 75 (Table 11), it is noted that during the write operation, while the second channel joins there is a sudden latency experienced (16.43 ms) in the network. Overall latency increases as the number of channels increases from one through four. It is also observed that a high rate in TPS also affects the network (TPS = 75 to TPS = 100) (Figure 11).

Table 12. Changing the number of channels: Read operations

Tx (n)	Channels (n)	Transactions (per second)	Send Rate (n)	Workers (n)	Throughput	Latency (ms)
5000	1	75	74.7	5	74.7	0.01
5000	2	75	74.7	5	74.6	0.02
5000	3	75	75.0	5	74.9	0.01
5000	4	75	74.9	5	74.9	0.01

Table 13. Changing the number of channels: Create operations

Tx (n)	Channel	Transactions (per second)	Send Rate (n)	Workers	Throughput	Latency (ms)
5000	1	100	99.3	5	98.8	0.95
5000	2	100	97.6	5	60.5	16.43
5000	3	100	99.7	5	99.4	0.74
5000	4	100	99.6	5	97.4	1.39

Table 14. Changing the number of channels. Read operations

Tx (n)	Channel	Transactions (per second)	Send Rate (n)	Workers	Throughput	Latency (ms)
5000	1	100	99.7	5	98.8	0.95
5000	2	100	99.0	5	60.5	16.43
5000	3	100	99.7	5	99.4	0.74
5000	4	100	99.7	5	97.4	1.39

Experiment 6: Impact of Block Interval (TPS = 75)

Block interval is the time taken to create a new block. With all the changes being observed, it is essential to now learn if there is any impact on the block intervals. Therefore, we set the block interval to 2s from 1s, with the block size continuing to be 50 itself for create (Table 15) and read (Table 16) operations.

Observations

Compared to Experiment 4 with BT = 1s, it is observed that there is less latency in the network even with more number of channels joining when the Batch time or block interval increases to 2s (Figure 12). But there is no evidence of a heavy change in throughput. Figure 12 shows a comparison in latencies for write operation for BT = 1s and BT = 2s for channel numbers increasing from 1 through 4.

Experiment 7: Impact of Block Size (TPS = 75, Block Time = 2s)

It is essential to understand the impact of block size in the network. Hence, we keep block size as the rate controller in experiment 7 for create (Table 17) and read (Table 18) operations.

Observations

It is observed that with an increase in BS (100), the throughput is steady. And the latency reduces compared

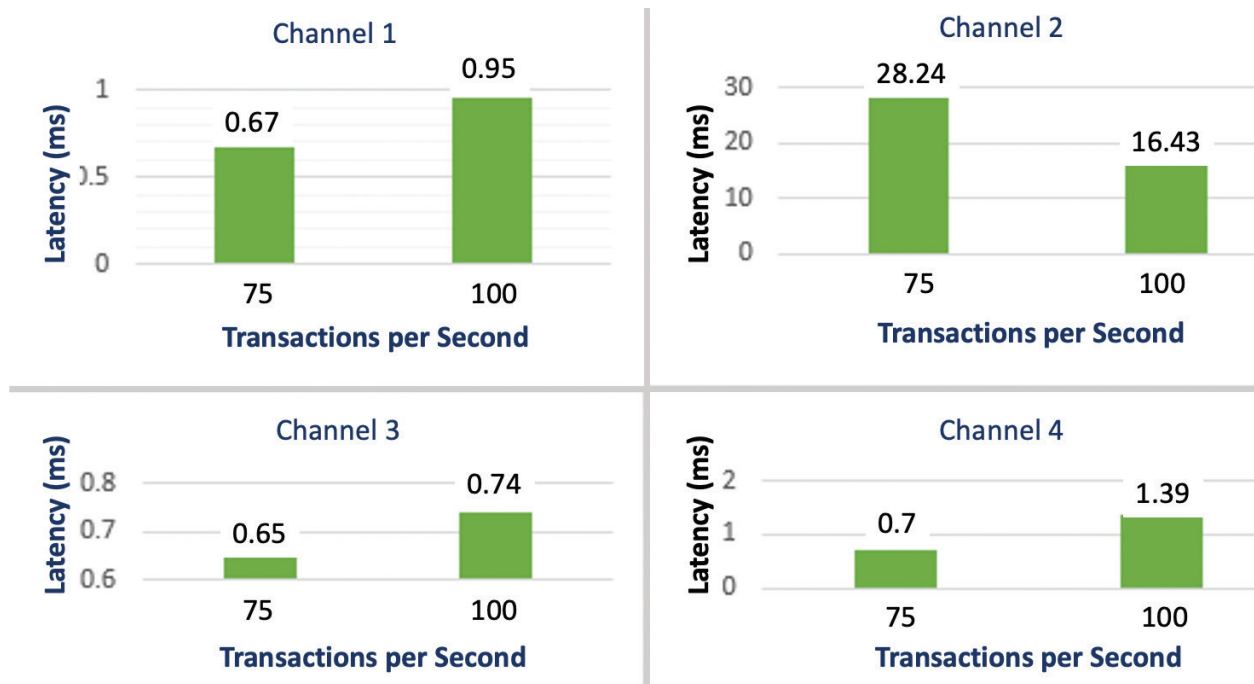


Fig. 11. Evaluation of latency for TPS (transactions per second) = 75 and TPS = 100 (Tables 11 and 13) in Write operation with the number of channels increased from 1 through 4.

Table 15. Changing block interval = 2s, transactions per second = 75: Create operations

Tx (n)	Channel	Transactions (per second)	Send Rate (n)	Workers	Throughput	Latency (ms)
5000	1	75	74.6	5	74.5	0.63
5000	2	75	74.8	5	74.6	0.62
5000	3	75	74.9	5	74.6	0.57
5000	4	75	74.6	5	74.4	0.61

Tx: transactions.

Table 16. Changing block interval = 2s: Read operations

Tx (n)	Channel	Transactions (per second)	Send Rate (n)	Workers	Throughput	Latency (ms)
5000	1	75	74.8	5	74.8	0.01
5000	2	75	74.9	5	74.9	0.01
5000	3	75	74.9	5	74.7	0.01
5000	4	75	74.8	5	74.8	0.01

Tx: transactions.

to BS = 50. Even when more number of channels joined, the throughput is not much affected when the block size is increased, but there is an increase in latency when the block size increases (Figure 13).

This section completes a detailed analysis of the proposed two-layer hierarchical model. We experiment with the entire network with a total of five organizations. The prime one is the IC, and all the four HNs them. Each channel is deployed with its own smart contract. The performance and latency analysis of

the network are benchmarked and evaluated against a two-organization network that has only two channels. The study varies the parameters and uses different rate controllers during the experiment. It is observed that as the number of organizations, channels, workers, and transactions increases, the latency of the network and the performance gradually falls. Effectively we conclude that vertical scaling is recommended for any organization that needs to improve the scalability of their network.



Fig. 12. Evaluation of block interval (1s and 2s) for TPS = 75 (Tables 11 and 15) in Write operation with the number of channels increase from 1 through 4. TPS: transactions per second.

Table 17. Changing block size = 100, block interval or block time = 2s: Create operations

Tx (n)	Channel	Transactions (per second)	Send Rate (n)	Workers (n)	Throughput	Latency (ms)
5000	1	75	74.8	5	74.5	0.67
5000	2	75	74.9	5	74.6	0.65
5000	3	75	74.7	5	74.5	0.67
5000	4	75	74.9	5	74.4	0.66

Tx: transactions.

Table 18. Changing block size = 100, block interval or block time = 2s: Read operations

Tx (n)	Channel	Transactions (per second)	Send Rate (n)	Workers (n)	Throughput	Latency (ms)
5000	1	75	74.9	5	74.9	0.01
5000	2	75	74.8	5	74.8	0.01
5000	3	75	74.9	5	74.9	0.01
5000	4	75	74.8	5	74.8	0.01

Tx: transactions; TPS: transactions per second.

Discussion

Impact on Healthcare

While we evaluate the performance of the entire model in terms of scalability, we would like to comment on how this entire exercise is useful for the healthcare community. As more users, clients, transactions, and patient data increase in the network, it becomes important to understand the capacity of the organization’s network (IC in the experiment). This study does not cover trust issues as we already understand BCT guarantees the security of patient data that is abundantly floating between insurance companies and different HNs. Hence,

we deal with scalability issues only in such systems that require vertical scaling, that is, increasing the capacity of the network. So, if more clients (workers in the experiment) post more patient transactions, the network could be scaled to handle such situations. Otherwise, as shown in the experiment, there could be severe latency as more hospital groups join, and the network might crash.

Related Works in Blockchain

The technology is being widely tested across many use cases like supply chains for agriculture sustainability²¹

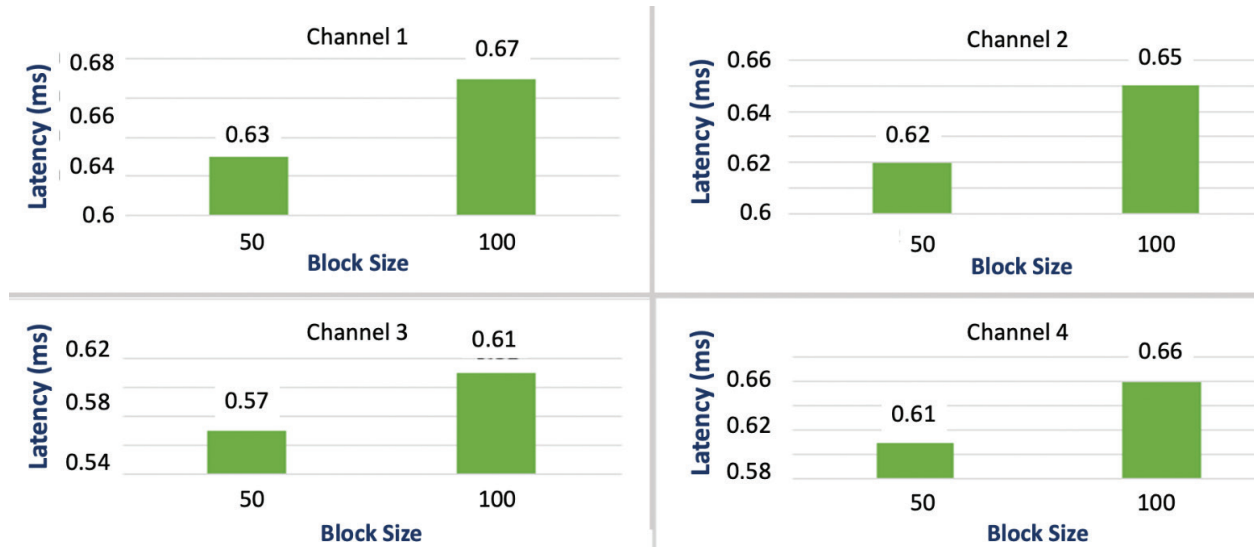


Fig. 13. Evaluation of block size (50 and 100) for TPS = 75 (Tables 13 and 15) in Write operation with the number of channels increase from 1 through 4. TPS: transactions per second.

and semiconductor industry²⁶ and other areas²⁷ for performance analysis. Bag et al. conducted a study on how BCT can be effective in small and medium enterprises (SMEs).²⁸ It is not always the performance analysis, but as research is delving more into this area, there is confidence now shown by organizations to implement blockchain technology. Another sector that recently made an impact was research in the automotive industry.²⁹ While most studies are still progressing in this technology, keeping track of sustainability and supply chain, interesting use cases from different parts of the world like Jordan are impressive.³⁰ Enormous studies have also been conducted in healthcare supply chain systems using blockchain technology.³¹ Though all sectors are trying to learn the implementation of blockchain, it is the supply chain that is mainly trying to research from all ends to check the resilience of the technology.³²

Conclusion

In this work, a two-level hierarchical model is implemented using Hyperledger Fabric as a solution to the scalability issues faced in BCT in the healthcare sector. Detailed performance analysis of the model is done by varying the number of transactions, workers or clients, channels, block size, and block intervals. When more channels are added, the major performance analysis on latency is made on the records creation operations rather than on read operations. This is because the read operations mostly show a negligible amount of latency. The benchmarking tool used for the analysis is Hyperledger Fabric. Each channel has a patient contract deployed on it. The entire experiment tries

to analyze how more patient data can be managed at hospitals while interacting with insurance companies for insurance claims. When block interval increases, there is less latency experienced, but it is studied that the block size increase is not recommended as there is higher latency experienced. Though more readings and experiments were conducted in this study, limited tables are included to show the performance of the network through worker size, channels, block size, and block intervals. The current model has only one smart contract implemented per channel.

The main limitation of the experiment is memory capacity. During the experiment at different stages, we scaled our network vertically, and it was observed that as we scale more, more organizations can be added to the network. Hence, the study emphasizes doing further experiments to check the maximum ability of a healthcare system network.

Future Work

One major challenge that might occur in a running real-time network could be that the network might crash if the scalability issues during heavy transactions are not foreseen. The initial overhead and usage of more memory during major transactions could be high. However, this is required to protect patient data and make sure smooth and effective implementations are deployed to handle healthcare data well in advance.

Future work will extend to further test the scalability of the network to the maximum, and the implementation of more organizations in the proposed hierarchical model to maintain more levels of HNs.

Funding

The was no funding for the preparation of the article.

Financial and non-Financial Relationships and Activities – no Financial Relationships

None are reported by the authors.

Contributors

All authors contributed to this paper. Lipsa Sadath developed the prototype, wrote the article and performed research. Deepti Mehrotra supervised the research, article and gave feedback. Anand Kumar gave feedback.

References

- Zhang S, Lee JH. Analysis of the main consensus protocols of blockchain. *ICT Express*. 2020 Jun 1;6(2):93–7. <https://doi.org/10.1016/j.ict.2019.08.001>
- Zhang R, Preneel B. Publish or perish: a backward-compatible defense against selfish mining in bitcoin. In *Topics in Cryptology–CT-RSA 2017: the cryptographers’ track at the RSA Conference 2017, San Francisco, CA, USA, February 14–17, 2017, Proceedings 2017* (pp. 277–292). Springer International Publishing. Available from: <https://www.esat.kuleuven.be/cosic/publications/article-2746.pdf>.
- Pahlajani S, Kshirsagar A, Pachghare V. Survey on private blockchain consensus algorithms. 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), Chennai, India, 2019, (pp. 1–6). <https://doi.org/10.1109/ICIICT.2019.8741353>
- Bonneau J, Narayanan A, Miller A, Clark J, Kroll JA, Felten EW. Mixcoin: anonymity for bitcoin with accountable mixes. In: R. Safavi-Naini & N. Christin (Eds.). *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Revised Selected Papers*. Springer Verlag. 2014. (pp. 486–504). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)). https://doi.org/10.1007/978-3-662-45472-5_31
- Zheng Z, Xie S, Dai H, Chen X, Wang H. An overview of blockchain technology: architecture, consensus, and future trends, 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 2017, (pp. 557–5640). <https://doi.org/10.1109/BigDataCongress.2017.85>
- Chauhan A, Malviya OP, Verma M, Mor TS. Blockchain and scalability. 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 2018, (pp. 122–128). <https://doi.org/10.1109/QRS-C.2018.00034>
- Kuo TT, Ohno-Machado L. Modelchain: decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks. *arXiv preprint arXiv:1802.01746*. 2018. <https://doi.org/10.48550/arXiv.1802.01746>
- Zhang P, White J, Schmidt DC, Lenz G, Rosenbloom ST. FHIR-Chain: applying blockchain to securely and scalably share clinical data. *Computational and structural biotechnology journal*. 2018 Jan 1;16:267–78. <https://doi.org/10.1016/j.csbj.2018.07.004>
- Ylonen T, Lonvick C. The secure shell (SSH) protocol architecture. 2006 Jan. <https://doi.org/10.17487/RFC4251>
- Glicksberg BS, Burns S, Currie R, Griffin A, Wang ZJ, Haussler D, et al. Blockchain-authenticated sharing of genomic and clinical outcomes data of patients with cancer: a prospective cohort study. *J Med Internet Res*. 2020 Mar 20;22(3):e16810. <https://doi.org/10.2196/16810>
- Lee HA, Kung HH, Udayasankaran JG, Kijisanayotin B, Marcelo A, Chao LR, et al. An architecture and management platform for blockchain-based personal health record exchange: development and usability study. *J Med Internet Res*. 2020 Jun 9;22(6):e16748. <https://doi.org/10.2196/16748>
- Bosworth HB, Zullig LL, Mendys P, Ho M, Trygstad T, Granger C, et al. Health information technology: meaningful use and next steps to improving electronic facilitation of medication adherence. *JMIR Med Inform*. 2016 Mar 15;4(1):e4326. <https://doi.org/10.2196/medinform.4326>
- Kakei S, Shiraiishi Y, Mohri M, Nakamura T, Hashimoto M, Saito S. Cross-certification towards distributed authentication infrastructure: a case of hyperledger fabric. *IEEE Access*. 2020 Jul 22;8:135742–57. <https://doi.org/10.1109/ACCESS.2020.3011137>
- Sahoo S, Fajge AM, Halder R, Cortesi A. A hierarchical and abstraction-based blockchain model. *Appl Sci*. 2019 Jun 7;9(11):2343. <https://doi.org/10.3390/app9112343>
- Cousot P. Abstract interpretation. *ACM Computing Surveys (CSUR)*. 1996 Jun 1;28(2):324–8. <https://doi.org/10.1145/234528.234740>
- Hassani H, Huang X, Silva E. Big-crypto: big data, blockchain and cryptocurrency. *Big Data Cogn Comput*. 2018 Oct 19;2(4):34. <https://doi.org/10.3390/bdccc2040034>
- Jana A, Halder R, Abhishekh KV, Ganni SD, Cortesi A. Extending abstract interpretation to dependency analysis of database applications. *IEEE Transac Softw Eng*. 2018 Jul 31;46(5):463–94. <https://doi.org/10.1109/TSE.2018.2861707>
- Blanchet B. Security protocol verification: symbolic and computational models. In *International conference on principles of security and trust 2012* Mar 24 (pp. 3–29). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Sadath L, Mehrotra D, Kumar, V. Scalability in Blockchain-Hyperledger Fabric and Hierarchical Model. 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT), New Delhi, India, 2022, (pp. 1–7). <https://doi.org/10.1109/GlobConPT57482.2022.9938147>
- Jiang L, Chang X, Liu Y, Mišić J, Mišić VB. Performance analysis of Hyperledger Fabric platform: a hierarchical model approach. *Peer-to-Peer Netw Appl*. 2020 May;13:1014–25. <https://doi.org/10.1007/s12083-019-00850-z>
- Al-Sumaidae G, Alkhudary R, Zilic Z, Swidan A. Performance analysis of a private blockchain network built on Hyperledger Fabric for healthcare. *Inform Process Manag*. 2023 Mar 1;60(2):103160. <https://doi.org/10.1016/j.ipm.2022.103160>
- Xu X, Sun G, Luo L, Cao H, Yu H, Vasilakos AV. Latency performance modeling and analysis for Hyperledger Fabric blockchain network. *Inform Process Manag*. 2021 Jan 1;58(1):102436. <https://doi.org/10.1016/j.ipm.2020.102436>
- Kuzlu M, Pipattanasomporn M, Gurses L, Rahman S. Performance analysis of a Hyperledger Fabric blockchain framework: throughput, latency and scalability. 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, (pp. 536–540). <https://doi.org/10.1109/Blockchain.2019.00003>
- Thakkar P, Nathan S, Viswanathan B. Performance benchmarking and optimizing Hyperledger Fabric blockchain platform. In 2018 IEEE 26th international symposium on modeling, analysis, and simulation of computer and telecommunication systems (MASCOTS) 2018 Sep 25 (pp. 264–276). IEEE. Available from: <https://api.semanticscholar.org/CorpusID:44088292>
- Sirur S, Nurse JR, Webb H. Are we there yet? Understanding the challenges faced in complying with the General Data Protection

- Regulation (GDPR). In Proceedings of the 2nd international workshop on multimedia privacy and security 2018 Jan 15 (pp. 88–95). Available from: <https://arxiv.org/abs/1808.07338v1>
26. Zkik K, Belhadi A, Rehman Khan SA, Kamble SS, Oudani M, Touriki FE. Exploration of barriers and enablers of blockchain adoption for sustainable performance: implications for e-enabled agriculture supply chains. *Int J Logist Res Appl*. 2023 Nov 2;26(11):1498–535. <https://doi.org/10.1080/13675567.2022.2088707>
 27. Tan CL, Tei Z, Yeo SF, Lai KH, Kumar A, Chung L. Nexus among blockchain visibility, supply chain integration and supply chain performance in the digital transformation era. *Ind Manag Data Syst*. 2023 Feb 3;123(1):229–52. <https://doi.org/10.1108/IMDS-12-2021-0784>
 28. Bag S, Rahman MS, Gupta S, Wood LC. Understanding and predicting the determinants of blockchain technology adoption and SMEs' performance. *Int J Logist Manag*. 2023 Dec 1;34(6):1781–807. <https://doi.org/10.1108/IJLM-01-2022-0017>
 29. Kamble SS, Gunasekaran A, Subramanian N, Ghadge A, Belhadi A, Venkatesh M. Blockchain technology's impact on supply chain integration and sustainable supply chain performance: evidence from the automotive industry. *Ann Oper Res*. 2023 Aug;327(1):575–600. <https://doi.org/10.1007/s10479-021-04129-6>
 30. Jum'a L. The role of blockchain-enabled supply chain applications in improving supply chain performance: the case of Jordanian manufacturing sector. *Manag Res Rev*. 2023 Jan 31. <https://doi.org/10.1108/MRR-04-2022-0298>
 31. Vishwakarma A, Dangayach GS, Meena ML, Gupta S, Luthra S. Adoption of blockchain technology enabled healthcare sustainable supply chain to improve healthcare supply chain performance. *Manag Environ Qual*. 2023 May 17;34(4):1111–28. <https://doi.org/10.1108/MEQ-02-2022-0025>
 32. Li G, Xue J, Li N, Ivanov D. Blockchain-supported business model design, supply chain resilience, and firm performance. *Transp Res E: Logist Transp Rev*. 2022 Jul 1;163:102773. <https://doi.org/10.1016/j.tre.2022.102773>

Copyright Ownership: This is an open access article distributed in accordance with the Creative Commons Attribution Non Commercial (CC BY-NC 4.0) license, which permits others to distribute, adapt, enhance this work non-commercially, and license their derivative works on different terms, provided the original work is properly cited and the use is non-commercial. See: <http://creativecommons.org/licenses/by-nc/4.0>.

Appendix. Key terminologies of the fabric framework are described through the use case.*

Terminology	Defined	Application
Node	<ul style="list-style-type: none"> The IC and each HN are considered as a node where each node joins for transactions through a peer in the network. They also endorse the policy for the transactions. 	<ul style="list-style-type: none"> Each peer maintains their respective ledger after every transaction. We consider the one HN as a group that has data from their hospital branches (e.g., Chicago, Dubai, etc.).
Chaincode	<ul style="list-style-type: none"> The chaincode (aka smart contract) contains the business logic between the IC and the HN. Four chaincodes are deployed separately for each channel that joins the IC. 	<ul style="list-style-type: none"> The basic data structure consists of patient ID, department, age, patient name, address, phone number, and bill amount (Table 5). For ease of use and populating the data, the chaincodes are renamed for deployment into the channels. Patient data are duplicated (Figure 6) to handle up to 15,000 transactions at times in order to check the scalability and performance of the network.
Endorsement policy	<ul style="list-style-type: none"> The endorsement policy specifies the required peers needed for endorsement. 	<ul style="list-style-type: none"> In our business logic, we implemented “OR” endorsement, where not all peers are required to endorse the request. Figure 7 illustrates a policy that either Org1 or Org3 can endorse through Channel 2. Only a few peers are eligible to endorse the transaction proposal.
Channel	<ul style="list-style-type: none"> Four channels connect the peers of the five organizations (i.e., the IC and the four HNs). 	<ul style="list-style-type: none"> Each channel is assigned a chaincode. Hence, each channel is governed by the respective pre-agreed business logic or policy as per the smart contract. Then, the ledger functions are initiated on these smart contracts. When a specific organization wants to transact business privately, its peers join the channels separately.
Ordering Service	<ul style="list-style-type: none"> In the model, three orderers request services to provide greater capacity to the network. 	<ul style="list-style-type: none"> This prevents unforeseen situations where the ordering service of one orderer stops. In such situations, other orderers take over to avoid network delays.
Certificate Authority	<ul style="list-style-type: none"> Each peer of an organization (IC and HNs) has their CA. 	<ul style="list-style-type: none"> CA is first up in the network to establish all the organizations.
Ledger	<ul style="list-style-type: none"> The data are stored in Couch DB for the respective organizations. 	<ul style="list-style-type: none"> According to the GDPR, 25 personal data should not be revealed in transactions. Accordingly, the advantage of blockchain is that it is the hash of the data stored in the blocks. Other data the transactions can pass on are the treatment and diagnostic details of the patient that must be checked by the IC.
Workers	<ul style="list-style-type: none"> The current experiment considers workers as the clients which come from the IC or any of the HNs. 	<ul style="list-style-type: none"> Clients can be doctors or IC agents who attempt to create a transaction or read a transaction as the business logic or the endorsement policy.
Fabric Gateway	<ul style="list-style-type: none"> A set of libraries that help invoke the business transaction through SC and the peers for endorsement. 	<ul style="list-style-type: none"> Following endorsement, the transaction is saved and distributed to other peers through the fabric gateway and the channels.

*The authors propose a hierarchical model within the Hyperledger Fabric enterprise application, focusing on the healthcare sector as a use case.

**Patient data structure in a smart contract is illustrated in Figure 5. CA: certificate of authority; Couch DB: a clustered database that allows running a single logical database server on any number of servers; GDPR: Data Protection Rule; HN: hospital network; IC: insurance company; ID: identification; OR: endorsement where not all peers are required to endorse the request. Org: organization number (e.g., ORG1). Figure 7 shows that policy as either Org1 or Org3 can be endorsed through Channel 2.; SC: smart contract.