Information Processing and Management 60 (2023) 103160

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Performance analysis of a private blockchain network built on Hyperledger Fabric for healthcare

Ghassan Al-Sumaidaee [a,*], Rami Alkhudary [b], Zeljko Zilic [a], Andraws Swidan [c]

[a] McGill University, 845 Sherbrooke St W, Montreal, H3A 0G4, Canada
[b] LARGEPA, Université Paris-Panthéon-Assas, 12 place du Panthéon, Paris, 75005, France
[c] University of Jordan, (sabbatical year at McGill university), Queen Rania St, Amman, 11942, Jordan

ARTICLE INFO

Keywords:
Blockchain
Healthcare
Hyperledger fabric
Hyperledger caliper

ABSTRACT

The healthcare industry suffers from poor interoperability due to its fragmented communication systems. Each medical institution has a communication system that is not necessarily compatible with the other systems on the same network. Poor communication has serious implications for patients, resources, and costs. The literature proposes blockchain technology as a governance data management system that optimizes the flow of information between multiple organizations by providing a trust component. However, much of the literature is conceptual and addresses blockchain in healthcare in a very general way. This paper contributes to the literature by presenting the use of Hyperledger Fabric in healthcare to improve information flow and solve the fragmentation problem between two medical institutions. In addition, two rate controllers on Hyperledger Caliper are used to evaluate the performance of our network: fixed and linear. We believe that our work will be useful for those working in value chains in healthcare and academia.

1. Introduction

Medical records contain confidential data such as surgical procedures, treatment, and recovery information. These records are still kept in paper form and manually archived. Even with increasingly digitized healthcare supply chains, electronic health records are fragmented and scattered across different organizations and geographic regions where there is no comprehensive system to update and share these records in real-time. This situation can be much more complicated due to privacy and patient data confidentiality issues (Aloqaily, Elayan, & Guizani, 2022; Benzidia, Ageron, Bentahar, & Husson, 2019; Bharimalla, Choudhury, Parida, Mallick, & Dash, 2021; Elayan, Aloqaily, & Guizani, 2021; Hayyolalam, Aloqaily, Özkasap, & Guizani, 2021; Otoum, Al Ridhawi, & Mouftah, 2022).

As a result, critical information is not efficiently shared with other medical institutions, let alone analyzed and used meaningfully to prevent or predict critical emergencies and deadly diseases. Medical data can be generated in real-time using devices from the Internet of Things. However, health data collection is fraught with inaccuracy and limited storage, and health data is not necessarily accessible to patients and other medical institutions on the same network (Al-Sumaidaee, Alkhudary, Zilic, & Féniès, 2021; Margheri, Masi, Miladi, Sassone, & Rosenzweig, 2020; Zhao, Chen, Liu, Baker, & Zhang, 2020).

Poor communication in healthcare is not limited to patient medical records but can extend to insurance and medications, causing significant problems for the healthcare industry. Mislabeled medications, for example, can lead to significant negative impacts

* Corresponding author.
E-mail addresses: ghassan.al-sumaidaee@mail.mcgill.ca (G. Al-Sumaidaee), rami.alkhudary@u-paris2.fr (R. Alkhudary), zeljko.zilic@mcgill.ca (Z. Zilic), sweidan@ju.edu.jo (A. Swidan).

https://doi.org/10.1016/j.ipm.2022.103160
Received 31 July 2022; Received in revised form 31 October 2022; Accepted 7 November 2022
Available online 22 November 2022

Table 1
List of abbreviations.

Term	Full abbreviation
MI_A	Medical Institution A
MI_B	Medical Institution B
C	Channel
CC	Channel Configuration
P_A	Peer Node A
P_B	Peer Node B
L	Ledger
O	Orderer Node
CA	Certificate Authority
ID_s	Identities
MSP	Membership Service Provider
S	Smart Contract
FG	Fabric Gateway
TPS	Transaction per Second
RAFT	Replicated And Fault-Tolerant

on people's lives and serious consequences for healthcare stakeholders, including hospitals, healthcare facilities, manufacturers, and distributors, resulting in a massive loss for the entire industry and its complex supply chains (O'Hagan & Garlington, 2018; Venhuis, Keizers, Klausmann, & Hegger, 2016). These problems are expected in Asia and Africa and the regions of Europe and the Americas (Kovacs et al., 2014; Parker, Sommer, et al., 2011).

The lack of coordination among medical institutions is related to privacy and technical issues, as well as the lack of a consensus mechanism for determining how data should be used or shared when needed, resulting in a waste of valuable resources. In addition, medical institutions are not the only components of a healthcare network. Other users can join and provide valuable data that can be included and shared, such as physicians, insurance providers, government institutions, device manufacturers, and researchers (Chen, Xu, Wang, & Yu, 2021; Uddin, 2021).

The fragmentation problem of health information systems can be solved by using a precise consensus mechanism that respects privacy and optimizes information flow. In this context, blockchain technology comes into play as a digital ledger or record of irreversible blocks of information (Alkhudary, Queiroz, & Féniès, 2022; Berdik, Otoum, Schmidt, Porter, & Jararweh, 2021; Pawar, Parolia, Shinde, Edoh, & Singh, 2022). The blocks are connected by a one-way cryptographic function that makes it impossible to change their contents. The chronological order of transactions forms the ledger, which is distributed among network members, with each member having an identical copy. The ledgers are synchronously updated through an agreement process or consensus algorithms (Alexandridis, Al-Sumaidaee, Alkhudary, & Zilic, 2021; Kebande, Awaysheh, Ikuesan, Alawadi, & Alshehri, 2021). Blockchain networks also include smart contracts, a type of interface that allows users to create new transactions that are permanently stored in the blockchain ledger (Chen, Srivastava, Parizi, Aloqaily, & Al Ridhawi, 2020; Chen et al., 2020; He et al., 2020). Many platforms propose to use blockchain to optimize the information flow between multiple organizations, e.g., Hyperledger Fabric (Xu et al., 2021).

Hyperledger is a non-profit and open source framework from the Linux Foundation that enables the building of decentralized and private blockchain applications to solve the problem of fragmentation and centralization in communication systems. Hyperledger is being proposed in the healthcare industry to address bottlenecks in the industry (Ammi, Alarabi, & Benkhelifa, 2021; Maskey, Badsha, Sengupta, & Khalil, 2021). Hyperledger enables organizations to share data across a distributed database without requiring an individual user to trust other users. It also allows medical transactions to be exchanged in a secure, transparent, and efficient manner. Therefore, we are building and testing a blockchain network between two medical institutions. Although our work is not the first (Figueroa-Lorenzo, Añorga, & Arrizabalaga, 2021; Tanwar, Parekh, & Evans, 2020; Xu et al., 2021; Zaabar, Cheikhrouhou, Jamil, Ammi, & Abid, 2021), our research goal is to illustrate a use case for deploying a private blockchain as a potential solution for integrating medical record systems.

The main contribution of this work can be summarized as follows: First, we illustrate a use case for deploying a private blockchain as a test network built on Hyperledger Fabric to optimize information flow and integrate two medical data systems. Second, our network handles situations where one party does not trust the other. Third, we use two rate controllers to evaluate the network's performance: fixed and linear, and report the best-case scenario in each experiment. Fourth, we explain the difference in parameters with each rate controller change. Finally, we propose a roadmap of future research directions formulating some axes of improvement.

The remainder of this paper is organized as follows: Section 2 describes the main components of the health network we developed. In Section 3, we conduct several experiments to evaluate the network's performance using Hyperledger Caliper and two rate controllers: fixed and linear. Finally, Section 3 concludes with remarks and future research directions.

2. Healthcare network typology built on hyperledger fabric

This section describes the problem our work addresses and the main components of the healthcare network: scenario, ordering service nodes, certificate authority, network administrators, network consortium, communication channel, peer nodes, smart contracts, fabric gateway, and endorsement policy.

2.1. Problem statement

Given the multiplicity and fragmentation of health information systems, it is critical to use a unified system to manage the availability and security of patient medical data (Berdik et al., 2021; Oham, Michelin, Jurdak, Kanhere, & Jha, 2021; Putz, Dietz, Empl, & Pernul, 2021) . Current medical record systems are fragmented, preventing integrated communication over the same network. For example, patients who need immediate and urgent care have to start from scratch with their data when they change hospitals. Therefore, there is an urgent need for flexible and on-demand record systems that enable smooth but secure medical data storage.

Another serious problem challenging existing health information systems is the centralization of patient data storage. This situation makes the entire system and medical data vulnerable to a single point of failure, where a slight manipulation of that data can seriously affect patients' lives. Tampering is not only due to hackers but also to human error. Controlling the procedures by which data is recorded and shared can solve this problem and ensure that data is secure and available relatively quickly.

The literature on blockchain in healthcare is full of simple experiments or conceptual frameworks that still need to be tested. In our paper, we build a test network for blockchain on Hyperledger Fabric and run several experiments to measure the network's performance whenever the system parameters change. An example of such a parameter is transaction rate control, which determines the overall system throughput and latency behavior. For clarity, Table 1 lists all abbreviations used in this document.

2.2. Scenario

Two medical institutions: MI_A and MI_B underpin a permissioned blockchain network via Hyperledger Fabric (Fig. 1). A set of policies (defined rules) governs the blockchain network and allows changes to the entire network, such as adding a new medical institution. We assume that MI_A is the originator of the network (who set up the medical network archetype). MI_A and MI_B need a communication channel C to exchange transactions: a fundamental building block in Hyperledger Fabric that organizes the communication and execution of transactions (Chen et al., 2021).

When we talk about transactions, we primarily mean the execution of smart contracts by the network's peers. This process results in information (outcomes) that, when validated, is permanently appended to the blockchain ledger. The channel is governed by pre-agreed rules specified in the channel configuration CC, which represents a new policy with some features, e.g., versioning for all configuration items so that each change can be uniquely identified.

Suppose MI_A has a client application that can transact over the channel C. MI_B can transact over the same channel. MI_A and MI_B have peer nodes: P_A and P_B, respectively. The peer nodes can be any machines used by the medical institutions to perform transactions. The peer nodes maintain identical copies of the blockchain ledger L, which is associated with the network channel C.

An ordering service is represented by the orderer node O, which has two main tasks: arrange the ordered transactions into blocks distributed to all the peer nodes for validation, and act as a network administrator controlling access to the network channel. It is worth noting that a group of orderer nodes form the ordering service, which is the first administrative point in the network. The network administrator sets up a certification authority CA for each medical institution: CA_A and CA_B, and each issues certificates or identities (IDs) to its employees.

2.3. Ordering service nodes

The first step in creating the blockchain network is to set up the ordering service (Bharimalla et al., 2021; Pawar et al., 2022). Our configuration has a single Replicated And Fault-Tolerant (RAFT) ordering service O. RAFT is the consensus algorithm used in Hyperledger Fabric. It is a replication protocol for state machines proposed to meet data security and other requirements: seamless on-demand access and solving the inconsistency problem of distributed databases. RAFT uses the leader-based mechanism where data can only flow from the leader node to the other server nodes. To achieve consensus when using RAFT, three processes must be considered. First, an immediate election of a network leader can occur if the current leader fails. Second, replication of the log entry across all network nodes is required. Third, there should be a permanent backup of the confirmed log entry in persistent storage (Alexandridis et al., 2021). For simplicity, O is the first management point for the blockchain network. Both can host the ordering service MI_A and MI_B.

2.4. Certificate authority

Other medical institutions need a valid certificate authority to interact with the peer nodes on the blockchain network (Chen et al., 2021; Nie, Long, Zhang, & Lu, 2022). In other words, the peer nodes of the blockchain network must be identified and authenticated; each peer node should maintain a CA. CA_s can identify which peer node belongs to which medical institution and what kind of responsibility it has. CA_s also qualify users to sign transactions to be appended to the blockchain ledger L. In our configuration design, we define two certificate authorities CA_A and CA_B for MI_A and MI_B, respectively. As mentioned earlier, the certificate authorities are responsible for issuing certificates or identities (IDs) to the network users. For an ID to become active, it must be validated by a network component called a Membership Service Provider MSP.

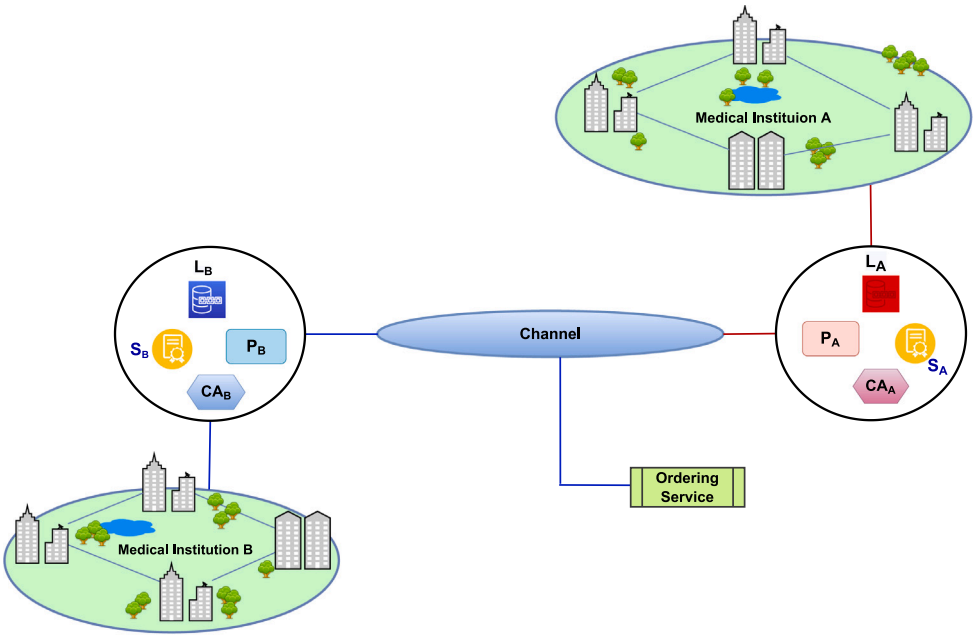


Fig. 1. Healthcare configuration based on Hyperledger Fabric.

2.5. Network administrators

MI_A and MI_B have the same administrative rights, so the users of MI_A and MI_B can manage the health network synchronously, e.g., each medical institution can propose to add new institutions or change policies. As mentioned earlier, the medical network has an ordering service represented by a single Raft node. This is because we built our configuration as a simple blockchain network with two medical institutions and one peer node each.

2.6. Network consortium

One of the most important issues in building permissioned blockchain networks is identifying the key players. A consortium generally refers to a set of actors (organizations) that transact with each other and join the network channel. It is worth noting that there can be more than one consortium within the blockchain network. Once the consortium is defined, the main actors can set up the communication channels. In our healthcare configuration, we have two medical institutions MI_A and MI_B that form the network consortium: the network users, peer nodes, and channels.

2.7. Communication channel

Since we have only two medical institutions, a channel C is established. It is worth noting that there can be more than one channel within a network, depending on the size of the network and the need for private communication between some of the network actors. The connected medical institutions manage the blockchain network through this channel. This configuration defines a set of policies (rules) that both MI_A and MI_B have over the channel, as well as the peer nodes that can participate in this channel. Regarding privacy, channels are important because they allow private communication between key players in the blockchain network (Gao, Lin, Chen, & Liu, 2021; Stamatellis, Papadopoulos, Pitropakis, Katsikas, & Buchanan, 2020).

2.8. Peer nodes

A blockchain network can have many actors. Each represents an independent entity or peer node with its users, e.g., doctors, nurses, etc. A peer node P is a machine with a storage capacity that helps maintain the blockchain network. For example, a doctor in a medical institution can represent a peer node. The peer nodes host a smart contract S and the ledger L (an identical copy of the blockchain database).

The ledger and smart contracts S are inextricably connected as the ledger registers all transactions generated by them. Each peer node must have one ledger L and can host multiple ledgers if it opts to be part of multiple channels. Our configuration includes two medical institutions, each with one peer node. The peer nodes are connected via the network channel C. Since we have only one channel C, the ledger in the peer nodes is identical: P_A and P_B.

2.9. Smart contracts

A smart contract S is a set of parameters (functions) that define its functionality and use on the channel. When multiple organizations join the same network, most organizations must agree to the definition for the actual use phase to begin. Since we have a channel C that includes only two medical institutions, an administrator of MI_A and MI_B takes care of approving the definition. Once the definition (predefined conditions) is approved, the smart contract can be used by client applications to query or update the ledger.

A user, like a doctor, can submit a client application to the blockchain network. The client application must be authenticated with the required certificate that is issued by CA_S and activated by MSP. Although client applications are not considered part of the blockchain network, they must maintain an identity (ID) because they are connected through the network channel. Once the certificate is instantiated and activated by the MSP, users can connect to the network by first connecting to one of the peer nodes. Typically, a doctor, physician, etc., will need to access the ledger periodically to see the latest value of the registered information. The ledger is accessed indirectly through the peer node, which can invoke the smart contract to process a request. In other words, smart contracts define the access rules for the ledger.

MI_A and MI_B must instantiate the functions of a smart contract, install them on their peer nodes, and then agree to them before deploying the smart contract on the channel C. The functions of a smart contract typically involve a business process maintained by both medical institutions. After the smart contract is instantiated, a package with a unique identifier is created by one of the medical institutions. The smart contract S can now be installed on the peer nodes. The installation of S on P_A and P_B does not yet mean that the smart contract is ready for use: the two medical institutions must approve the functionality of S.

2.10. Fabric gateway

A transaction is generated whenever a client application attempts to invoke the smart contract to implement one of the business processes. The transactions are eventually distributed to P_A and P_B for validation. Suppose that a doctor wants to invoke the smart contract S on P_A. He can do so by submitting a transaction proposal to P_A via the fabric gateway FG: a program that forwards transaction proposals to peer nodes for endorsement. In response, P_A invokes the smart contract S to generate an endorsed response. The FG retains the responses from both P_A and P_B, and then transmits them to the client application (the doctor in our example). The client application transfers the responses to the orderer node to pack them into blocks. The orderer node contains the responses and arranges them into a transaction block that is distributed to the peer nodes to update L.

Fabric Gateway FG can contribute to the network efficiency. Any fabric network can be subject to frequent changes that make the network setup changeable over time. Examples of these changes include removing and (or) adding some network components such as peers, ordering nodes, certificate authorities, etc. These changes are more realistic and predictable as most organizations resort to these changes based on need and demand. FG can help simplify how the network handles these changes. An example of these issues that FG can handle is the application that needs to be submitted to the network. In the earlier Fabric network, the application had to be connected to more than one peer on the network before it received enough endorsements. In the latest versions of Hyperledger Fabric, where FG is used, the application only needs to be submitted to one peer in the network, and FG can handle the forwarding of that application to peer nodes throughout the network. In this context, applications can use FG in two ways: statically and dynamically. The difference between these two options is in the configuration. The gateway configuration is fully defined in the static method, while in the dynamic method, the configuration is only minimally defined.

Fig. 2 explains how a transaction is generated and accepted in our network using the Hyperledger framework. (A): A client application, e.g., a doctor, sends a transaction proposal to either CreateRecord (register information) or read it. (B): the fabric gateway, a program that forwards the transaction proposals to the peer nodes and invokes the smart contracts that create a new transaction. (C1–2): when the client application is authorized to create or read the transaction, a positive response is sent back from the smart contracts to FG, which forwards this response to the client application. (D) The client application collects the responses and sends them to the ordering node. (E) The ordering node collects the responses received, arranges them into a block of transactions, and sends them back to the peers for final review. Each peer verifies and confirms the transactions. The transactions are appended to the blockchain ledger if the block has been validated, so the blockchain ledger is updated (F); otherwise, the transactions are rejected.

2.11. Endorsement policy

As mentioned earlier, the definition of the smart contract contains several parameters that determine its functionality. One of the most important parameters is the endorsement policy which specifies which peer nodes must execute transactions. In our medical network, transactions can be considered valid if they are endorsed only by P_A and P_B. The endorsement policy is implemented in the chaincode definition and made available in the network channel C. This allows both P_A and P_B to easily access the endorsement policy.

3. Performance analysis and evaluation

In this section, we describe the setup of the experimental network and the associated performance tool.

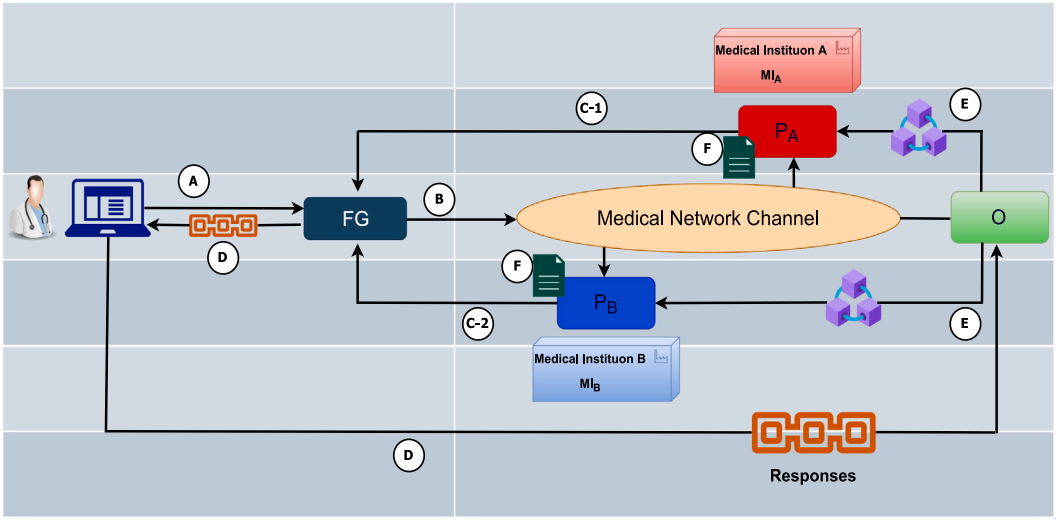


Fig. 2. Transaction overflow in the medical network.

3.1. Network setup

The blockchain network used in our design is Hyperledger Fabric version. The medical network consists of two medical institutions and one ordering node. Each of the two institutions has a peer, and the ordering service is represented by an ordering node that uses RAFT. The network is designed so that both organizations can confirm transactions to be valid. Smart contracts, referred to as "chaincode" in the Fabric network, can be written in various programming languages such as Go, JavaScript, and Java. This work used a smart contract written in JavaScript to test the performance. Fabric has two types of databases: the level and couch databases. The database used in this network is the level database.

3.2. Test environment

As a reminder, Hyperledger Fabric (Fabric, 0000) is an open-source framework that implements the idea of authorized participation in the blockchain network so that all participants are known and can be securely authenticated. Fabric is a modular architecture, meaning its components are designed to meet the needs of businesses and enable rapid transaction processing. Hyperledger Fabric processes transactions in multiple stages, meaning transactions are first executed, then ordered by the consensus mechanism, and finally validated before being stored in the ledger. This mechanism enables fast and efficient processing of a large volume of incoming transactions and improves scalability (Xu et al., 2021).

In this work, the entire components of the Hyperledger Fabric network were deployed as Docker containers and implemented on a local lab computer. Docker (0000) is an open-source project that allows developers to build, deploy, update, manage and run their applications in the cloud. Docker containers are executable software packages that provide developers with a form of isolation that allows them to run multiple applications without needing an entire operating system. The virtual machine on which this network is deployed has 6 CPU cores: Intel R Core(TM) 2.60 GHz, Ram 16.0 GB. It runs the Linux distribution Ubuntu 20.04 with new Fabric version installed.

3.3. Performance measurement

Hyperledger (Caliper, 0000) was used to monitor the network performance. It is a blockchain benchmark tool written in JavaScript that allows users developing blockchain solutions to measure the performance of those networks against a set of predefined use cases. Caliper uses four performance indicators to measure the performance of a network. These indicators include the number of transactions submitted per second TPS, known as "throughput", the rate of successful and failed transactions, the latency of transactions, and the resources used. Table 2 defines these indicators.

Table 2
Presentation of Caliper indicators.

Performance Indicator	Definition
Throughput	The average number of transactions processed per second.
Latency	The overall time the transactions take from issuing to the response.
Successful/Fail	The number of successful and failing transactions.
Resources	Statistics about the utilized resources throughout the measurement.

3.4. Benchmark workspace

A workspace was set up on a virtual machine where all the generated benchmark files were stored. Usually, this workspace contains three main sub-workspaces: Networks, Workload, and Benchmark. The Networks workspace includes a network configuration, a file required for submitting and evaluating transactions on the Hyperledger Fabric network. The file contains the necessary information about the network, such as the organizations, channels, customers, etc. The workload represents the core of Caliper, as it interacts with the deployed smart contract and decides what type of transaction should be submitted at a given time. Finally, the benchmark file is a critical configuration file for the execution of Caliper, since it determines the execution of the benchmark workload and the collection of the results.

3.5. Benchmark set up

Based on our use case and setup, we can use the Caliper tool to measure the performance of the medical network. This tool generates a report with different indicators that we can use to adjust and further improve the network. In this context, our benchmark configuration was initially created to replicate the following parameters:

1. The worker numbers indicate the number of worker processes in which each worker can submit a transaction. This number was set to three different values: 5, 25, and 50 workers.
2. Test rounds: In our benchmark configuration file, we used two types of transactions: CreateRecord and ReadRecord. For CreateRecord, we start by creating the records in the blockchain, while in the ReadRecord, we can start by reading the data from the smart contract.
3. Number of transactions: The number of transactions submitted in each round. This value has been changed several times to get different observations.
4. Rate control: Transactions submitted to the blockchain network can be matched at a fixed rate. The fixed rate can be controlled by users until the system's inflection point is found. For CreateRecord, we used two types of rate controls. First, we used the fixed-rate controller. Second, we used the linear type of rate control. The fixed-rate controller sets a fixed rate of transactions to be sent by the network workers. The linear rate controller has two options: Start TPS, which is the rate of transactions transmitted by workers at the beginning of each round, and Finish TPS, which is the rate of transactions submitted by workers at the end of each round. As with ReadRecord, we used the fixed-type rate control for the experiments. However, it should be noted that the rate controllers' parameters may need to be changed several times until the best utilization rates are found, eventually affecting the network's overall performance. Finally, in both the "ReadRecord" and "CreateRecord" benchmark rounds, we will examine the impact of changing the aforementioned parameters on the network's performance metrics in terms of latency, throughput, and send rate.

3.6. Use cases and results

As mentioned, we will monitor the network's performance in terms of three indicator: Latency, Throughput, and Send rate. For this purpose, several parameters will be tested to find the system's inflection point.

3.6.1. Fixed-rate controller

Experiment 1 (workers number):

In Caliper, workers represent the number of network clients used to run the workload, and are usually represented by a set of keys. In this experiment, we change the number of workers and then observe the impact on the performance indicators. We chose three numbers of workers: 5, 25, and 50 workers. We initially set the number of transactions to 1000 transactions; and since we use a fixed rate controller that sets the number of transactions sent by the specified workers, the transaction rate was set to 75 transactions per second.

1. CreateRecord: We observed stable system performance when we set the number of workers to 25. In other words, network latency increased by no more than 10%, while the network throughput and send rate changed slightly (less than 0.6%). Table 3 shows the performance results.
2. ReadRecord: The above setting shows an increase in the overall throughput of the system when we increase the number of workers. However, an increase in the overall latency was observed (above 100%). Accordingly, we find that setting the number of workers to 25 was the best-case scenario, as the overall system latency remained stable while the send rate moved at the same rate as the throughput (5.9%). Table 4 shows the performance during the experiment with the read entries.

Table 3
CreateRecord performance indicators—Experiment 1.

CreateRecord					
N. of workers	txNumber	TPS	Latency	Throughput	Send rate
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

Table 4
ReadRecord performance metrics—Experiment 1.

ReadRecord					
N. of workers	txNumber	TPS	Latency	Throughput	Send rate
5	1000	75	0.01	5.1	5.1
25	1000	75	0.01	5.4	5.4
50	1000	75	0.02	5.8	5.8

Table 5
CreateRecord performance metrics—Experiment 2.

CreateRecord					
txNumber	N. of workers	TPS	Latency	Throughput	Send rate
1000	5	75	0.1	75.1	75.4
5000	5	75	0.09	75	75.1
15 000	5	75	0.09	75.1	75

Table 6
ReadRecord performance metrics—Experiment 2.

ReadRecord					
txNumber	N. of workers	TPS	Latency	Throughput	Send rate
1000	5	75	0.01	5.1	5.1
5000	5	75	0.01	5.1	5.1
15 000	5	75	0.01	5.1	5.1

Experiment 2 (transactions number):

When running a performance evaluation for a blockchain network, it is necessary to provide the transaction number. This number represents the transactions that must be executed in each round during the evaluation. In this experiment, we test three transaction numbers (1000, 5000, and 15 000) to determine their impact on the network performance. Since we saw the impact of the number of workers in the previous experiment, we set the number of workers to 5 during the transaction number changes. The number of transactions per second was not changed and is still set to 75 TPS.

1. CreateRecord: Latency decreased by 10% when the number of transactions was increased from 1000 to 15 000 (best-case scenario) as seen in Table 5. At the same time, throughput remained about the same with these transaction changes. The send rate decreased by 0.53%. This shows that increasing the transaction rate (15 000) positively affects the system as the latency is reduced while the throughput remains stable.
2. ReadRecord: As seen in Table 6, the increase in transactions has no effect on the overall ReadRecord for latency, throughput, and send rate. We believe that the channel does not suffer overload problems when reading the network records.

Experiment 3 (TPS rate):

In this round, we continue to use the fixed rate controller. In this type of controller, the TPS must be specified because it sets the rate of transactions sent by the network workers. In this experiment, we set the number of transactions to 1000 and leave the number of workers at 5 while we change the TPS and observe the effects of this rate.

1. CreateRecord: Table 7 shows that setting the TPS rate to 150 was the best-case scenario. First, the network latency decreased by 25%. Second, throughput increased proportionally. Third, when TPS was increased to 250 TPS, the throughput and latency of the network were negatively impacted.
2. ReadRecord: Again, and as in Table 8 Increasing TPS has no effect on the overall ReadRecord in terms of latency, throughput, and send rate.

From the experiments conducted in the two rounds (CreateRecord and ReadRecord) and the results obtained with this rate controller (Fixed-rate), we can summarize the main results as follows:

Table 7
CreateRecord performance metrics—Experiment 3.

CreateRecord					
TPS	txNumber	N. of workers	Latency	Throughput	Send rate
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

Table 8
ReadRecord performance metrics—Experiment 3.

RedRecord					
TPS	txNumber	N. of workers	Latency	Throughput	Send rate
75	1000	5	0.01	5.1	5.1
150	1000	5	0.01	5.1	5.1
250	1000	5	0.01	5.1	5.1

Table 9
CreateRecord performance metrics—Experiment 4.

CreateRecord						
N. of workers	txNumber	TPS start	TPS finish	Latency	Throughput	Send rate
5	1000	25	75	0.13	37.3	37.3
25	1000	25	75	0.15	38.2	38.3
50	1000	25	75	0.14	38.7	38.9

Table 10
ReadRecord performance metrics—Experiment 4.

ReadRecord						
N. of workers	txNumber	TPS start	TPS finish	Latency	Throughput	Send rate
5	1000	25	75	0.01	15.1	15.1
25	1000	25	75	0.01	15.4	15.4
50	1000	25	75	0.01	15.8	15.8

Result1: *The higher the number of workers, the more likely the blockchain network will be overloaded.*
Result2: *Transactions are likely to fail as TPS rates increase, which negatively impacts the network's throughput.*

3.6.2. Linear rate controller

For the following experiments, we decided to use a different type of rate controller to investigate how different rate controllers affect the network's overall performance. For this purpose, instead of using a fixed TPS, we used a linear rate controller that splits the TPS into two rates. The first rate represents the transactions submitted by workers at the beginning of the round. The second rate represents the transactions submitted by workers at the end of the round.

Experiment 4 (workers number):

In this experiment, different numbers of workers were tested to observe the performance evaluation. We start with 5 workers, move to 25, and end with 50. With these changes in the number of workers, the number of transactions was set to 1000, and the start and finish TPS were set to 25 and 75, respectively.

1. CreateRecord: When we increased the number of workers from 5 to 25, a 15% increase in network latency was observed. This value decreased when we increased the number of workers from 25 to 50. A slight increase in network throughput and send rate was also observed when we increased the number of workers, as seen in Table 9. We note that it is recommended to set the number of workers to 50 (base-case scenario in our experiment). This is because the network latency did not increase by more than 7.7% even though we increased the number of workers by 900%. However, the throughput and send rate did not increase significantly.
2. ReadRecord: Again, in this round of testing, we tried increasing the number of workers while keeping the other network parameters fixed such as the number of transactions and the TPS rate. As can be seen in Table 10, increasing the number of workers has a negligible effect on the network throughput and the send rate for ReadRecord. However, increasing the number of workers has no discernible effect on network latency, as the latency value did not change throughout the experiments.

Experiment 5 (transaction numbers):

1. CreateRecord: Since we found that network latency increased when we changed the number of workers, we decided to monitor another parameter that could help reduce latency. Therefore, we left TPS's start and finish values at 25 and 75, respectively,

Table 11
CreateRecord performance metrics—Experiment 5.

CreateRecord						
txNumber	N. of workers	TPS start	TPS finish	Latency	Throughput	Send rate
1000	5	25	75	0.13	37.3	37.3
5000	5	25	75	0.12	37.2	37.3
15 000	5	25	75	0.12	37.3	37.3

Table 12
ReadRecord performance metrics—Experiment 5.

ReadRecord						
txNumber	N. of workers	TPS start	TPS finish	Latency	Throughput	Send rate
1000	5	25	75	0.01	15.1	15.1
5000	5	25	75	0.01	15.1	15.1
15 000	5	25	75	0.01	15.1	15.1

Table 13
CreateRecord performance metrics–Experiment 6.

CreateRecord						
TPS start	TPS finish	txNumber	N. of workers	Latency	Throughput	Send rate
25	75	1000	5	0.13	37.3	37.3
50	100	1000	5	0.09	66	66.2
75	150	1000	5	0.08	98.3	98.6

Table 14
ReadRecord performance metrics—Experiment 6.

ReadRecord						
TPS start	TPS finish	txNumber	N. of workers	Latency	Throughput	Send rate
25	75	1000	5	0.01	15.1	15.1
50	100	1000	5	0.01	15.1	15.1
75	150	1000	5	0.01	15.1	15.1

while setting the number of workers to 5. The result showed a change in network latency, which decreased by 8% compared to the latency of the previous experiment. However, increasing the number of transactions did not notably impact the throughput and send rate. Therefore, it is best to set the number of transactions in our experiment to 15 000 transactions (Table 11).

2. ReadRecord: Again, in this round of testing, we tried increasing the number of transactions while fixing other parameters to see how this affected network's performance. As can be notice in the table below (Table 12), we start with 1000 transactions, then increase the number of transactions to 5000 transactions, and finally to 15 000 transactions. Both the number of workers and the TPS start and TPS finish were set to a fixed value. In doing so, we found that increasing the number of transactions from 1000 to 15 000 and even further had no effect on network latency, throughput, or send rate. The table shows the stability of these values with various changes in the number of transactions.

Experiment 6 (TPS rate):

Here we manipulate the values for starting and finishing TPS and then observe the performance. The number of transactions and the number of workers were set to 1000 and 5, respectively. We start by setting the start value of TPS to 25 and finish value to 75. Next, we set the start value of TPS to 50 and finish value to 100. Finally, we set the start value of TPS to 75 and finish value to 150.

1. CreateRecord: We found an inverse relationship between network throughput and latency as we continue increasing the values for TPS's start and finish values. As seen in Table 13, the network latency steadily decreased with each increase in the values of TPS. However, the increase in TPS was accompanied by increased network throughput. Consequently, the best-case scenario, in this case, depends entirely on the user's preference for a decrease in latency or an increase in the network throughput.
2. ReadRecord: When the values of TPS were manipulated, there was no change in the ReadRecord values, as shown in the following Table 14.

From the experiments conducted in the two rounds (CreateRecord and ReadRecord) and the results obtained with this rate controller (Linear-rate), we can summarize the main results as follows:

Result3: *Increasing TPS rates have a positive impact on the network's latency, throughput, and send rate.*

Result4: *The number of workers in the network has a positive effect on the network's throughput.*

4. Discussion

Blockchain has attracted increasing attention in recent years due to its success in the financial sector. This has paved the way for studying blockchain applications in value chains, such as the healthcare supply chain. There has been a significant increase in research papers focusing on blockchain and its underlying architectures (Al-Sumaidaee, Alkhudary, Zilic, & Féniès, 2023; Dursun & Üstündağ, 2021; Haouari, Mhiri, El-Masri, & Al-Yafi, 2022; Putz et al., 2021). This work continues these efforts and proposes a use case for using Hyperledger Fabric to integrate medical data systems. The following paragraphs summarize the main results based on the two rate controllers used in this work, i.e., fixed rate and linear rate controllers. We also discuss the theoretical and practical implications of the obtained results.

Fixed-rate: In the experiments we conducted in this paper, we found that the number of workers in the blockchain network impacts the latency. The higher the number of workers, the more likely the network will be overloaded. From a theoretical point of view, this is logical, as the network was built based on a simple scenario that assumes the existence of two medical institutions. In practice, the network is expected to grow, and this overload can be overcome by expanding the network resources, such as network channels and peer nodes. In addition, the number of transactions per second impacts the throughput of the blockchain network. If we increase TPS further, transactions may fail, and consequently, the throughput of the network will be negatively affected. This is also understandable because we use a fixed rate of a certain number of transactions injected into the network. Some networks may need to process a large amount of data, so the fixed rate can be adjusted to the volume of data received.

Linear-rate:

As mentioned, Hyperledger Caliper provides multiple rate controllers to control the send rate of transactions in a blockchain network. Selecting the appropriate rate plays a vital role in benchmarking blockchain network performance. Caliper also supports network developers in creating their rate controllers, which we believe should be considered in other value networks. We have chosen two rate controllers in this work: fixed and linear. In the experiments performed, it has been shown that the number of workers in the network affects the latency when the rate controller is changed. As for the practical impact, our results depend on the network volume and the number of workers in the blockchain network. For networks dealing with urgent cases, latency is a critical factor in determining the use of blockchain in healthcare value chains. For example, a trade-off between the number of workers and network latency should be carefully considered. In our experiments, we changed the start and finish rates. It has been shown that increasing these rates can positively impact the latency, throughput, and send rate of the blockchain network. Large value chains can benefit from this rate controller as it provides more flexibility in processing input transactions.

5. Concluding the article with remarks and future research

Our paper proposes a blockchain test network built on Hyperledger Fabric to optimize the information flow between two medical institutions, decentralize the network, and improve the management of shared information even if one of the users does not trust the others. Hyperledger is a robust and open-source framework from the Linux Foundation that facilitates the development of decentralized applications based on blockchain. We also use Hyperledger Caliper as a blockchain benchmark tool to evaluate the performance of the developed network. Although our work is not one of the first works to use Hyperledger to improve healthcare nor one of the first works to analyze the performance of blockchain networks (Spataru, Pungila, & Radovancovici, 2021), it has contributed to the literature by using two rate controllers to evaluate the performance of the blockchain network: fixed and linear. Our paper contributes to the literature by using and comparing the fixed and linear controllers in several experiments, explaining the variations when changing the parameters, and picking out the best-case scenarios.

The blockchain network presented in this paper offers a solution to the centralization and fragmentation of healthcare information systems. This is because each medical institution in the network has an identical copy of the blockchain ledger. These ledgers are updated in real-time and do not follow the client–server communication model. However, it is worth noting that only images or the key values of medical information, such as patient records, are registered in the blockchain ledger in a decentralized manner. The source data cannot be stored in the network's ledgers. The current trend in the industry is to host the data source of network communications in the cloud (Li, Liang, Zhang, Wang, & Luo, 2022; Li, Wu, Jiang, & Srikanthan, 2020; Zhang, Yang, Xie, & Liu, 2021).

In future research, we urge researchers to investigate the use of the InterPlanetary File System (IPFS), a peer-to-peer storage system (Cangir, Cankur, & Ozsoy, 2021; Khalid et al., 2021), to ensure network decentralization. We also propose using other rate controllers on Hyperledger Caliper that are more sophisticated than the fixed and linear controllers. We can use a number of different rate controllers: fixed feedback rate, fixed load, maximum rate, composite rate, zero rate, record rate, and replay rate. Each of these controllers can affect performance as the parameters of the network change. Future research could investigate the differences between these rates and their impact on performance. Finally, we performed several experiments on a virtual machine. We believe that repeating the experiments to test the added value of our presentation statistically is artificial. Future research could statistically test our approach with a relatively large blockchain network.

CRediT authorship contribution statement

Ghassan Al-Sumaidaee: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Rami Alkhudary:** Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Zeljko Zilic:** Writing – original draft, Writing – review & editing. **Andraws Swidan:** Writing – original draft, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgments and have given us their written permission to be named. If we have not included an Acknowledgments, then that indicates that we have not received substantial contributions from non-authors. All authors approved the version of the manuscript to be published.

References

Al-Sumaidaee, G., Alkhudary, R., Zilic, Z., & Féniès, P. (2021). A blueprint towards an integrated healthcare information system through blockchain technology. In *HEALTHINFO 2021, the sixth international conference on informatics and assistive technologies for health-care, medical support and wellbeing* (p. 32).

Al-Sumaidaee, G., Alkhudary, R., Zilic, Z., & Féniès, P. (2023). Configuring blockchain architectures and consensus mechanisms: The healthcare supply chain as a use case. In *Blockchain driven supply chains and enterprise information systems* (pp. 135–150). Springer.

Alexandridis, A., Al-Sumaidaee, G., Alkhudary, R., & Zilic, Z. (2021). Making case for using RAFT in healthcare through hyperledger fabric. In *2021 IEEE international conference on big data (big data)* (pp. 2185–2191). IEEE.

Alkhudary, R., Queiroz, M. M., & Féniès, P. (2022). Mitigating the risk of specific supply chain disruptions through blockchain technology. In *Supply chain forum: an international journal* (pp. 1–11). Taylor & Francis.

Aloqaily, M., Elayan, H., & Guizani, M. (2022). C-healthier: A cooperative health intelligent emergency response system for C-ITS. *IEEE Transactions on Intelligent Transportation Systems*.

Ammi, M., Alarabi, S., & Benkhelifa, E. (2021). Customized blockchain-based architecture for secure smart home for lightweight IoT. *Information Processing & Management*, *58*(3), Article 102482.

Benzidia, S., Ageron, B., Bentahar, O., & Husson, J. (2019). Investigating automation and AGV in healthcare logistics: a case study based approach. *International Journal of Logistics Research and Applications*, *22*(3), 273–293.

Berdik, D., Otoum, S., Schmidt, N., Porter, D., & Jararweh, Y. (2021). A survey on blockchain for information systems management and security. *Information Processing & Management*, *58*(1), Article 102397.

Bharimalla, P. K., Choudhury, H., Parida, S., Mallick, D. K., & Dash, S. R. (2021). A blockchain and NLP based electronic health record system: Indian subcontinent context. *Informatica*, *45*(4).

Caliper 0000. Caliper, https://hyperledger.github.io/caliper/.

Cangir, O. F., Cankur, O., & Ozsoy, A. (2021). A taxonomy for blockchain based distributed storage technologies. *Information Processing & Management*, *58*(5), Article 102627.

Chen, Q., Srivastava, G., Parizi, R. M., Aloqaily, M., & Al Ridhawi, I. (2020). An incentive-aware blockchain-based solution for internet of fake media things. *Information Processing & Management*, *57*(6), Article 102370.

Chen, Z., Xu, W., Wang, B., & Yu, H. (2021). A blockchain-based preserving and sharing system for medical data privacy. *Future Generation Computer Systems*, *124*, 338–350.

Docker 0000. Dcoker, https://www.docker.com.

Dursun, T., & Üstündağ, B. B. (2021). A novel framework for policy based on-chain governance of blockchain networks. *Information Processing & Management*, *58*(4), Article 102556.

Elayan, H., Aloqaily, M., & Guizani, M. (2021). Digital twin for intelligent context-aware IoT healthcare systems. *IEEE Internet of Things Journal*, *8*(23), 16749–16757.

Fabric, H. 0000. Hyperledger Fabric, https://www.hyperledger.org/use/fabric.

Figueroa-Lorenzo, S., Añorga, J., & Arrizabalaga, S. (2021). Methodological performance analysis applied to a novel IIoT access control system based on permissioned blockchain. *Information Processing & Management*, *58*(4), Article 102558.

Gao, Y., Lin, H., Chen, Y., & Liu, Y. (2021). Blockchain and SGX-enabled edge-computing-empowered secure IoMT data analysis. *IEEE Internet of Things Journal*, *8*(21), 15785–15795.

Haouari, M., Mhiri, M., El-Masri, M., & Al-Yafi, K. (2022). A novel proof of useful work for a blockchain storing transportation transactions. *Information Processing & Management*, *59*(1), Article 102749.

Hayyolalam, V., Aloqaily, M., Özkasap, Ö., & Guizani, M. (2021). Edge intelligence for empowering IoT-based healthcare systems. *IEEE Wireless Communications*, *28*(3), 6–14.

He, D., Deng, Z., Zhang, Y., Chan, S., Cheng, Y., & Guizani, N. (2020). Smart contract vulnerability analysis and security audit. *IEEE Network*, *34*(5), 276–282.

Kebande, V. R., Awaysheh, F. M., Ikuesan, R. A., Alawadi, S. A., & Alshehri, M. D. (2021). A blockchain-based multi-factor authentication model for a cloud-enabled internet of vehicles. *Sensors*, *21*(18), 6018.

Khalid, A., Iftikhar, M. S., Almogren, A., Khalid, R., Afzal, M. K., & Javaid, N. (2021). A blockchain based incentive provisioning scheme for traffic event validation and information storage in VANETs. *Information Processing & Management*, *58*(2), Article 102464.

Kovacs, S., Hawes, S. E., Maley, S. N., Mosites, E., Wong, L., & Stergachis, A. (2014). Technologies for detecting falsified and substandard drugs in low and middle-income countries. *PLoS One*, *9*(3), Article e90601.

Li, C., Liang, S., Zhang, J., Wang, Q.-e., & Luo, Y. (2022). Blockchain-based data trading in edge-cloud computing environment. *Information Processing & Management*, *59*(1), Article 102786.

Li, J., Wu, J., Jiang, G., & Srikanthan, T. (2020). Blockchain-based public auditing for big data in cloud storage. *Information Processing & Management*, *57*(6), Article 102382.

Margheri, A., Masi, M., Miladi, A., Sassone, V., & Rosenzweig, J. (2020). Decentralised provenance for healthcare data. *International Journal of Medical Informatics*, *141*, Article 104197.

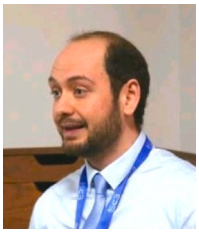
Maskey, S. R., Badsha, S., Sengupta, S., & Khalil, I. (2021). ALICIA: Applied intelligence in blockchain based VANET: Accident validation as a case study. *Information Processing & Management*, *58*(3), Article 102508.

Nie, Z., Long, Y., Zhang, S., & Lu, Y. (2022). A controllable privacy data transmission mechanism for internet of things system based on blockchain. *International Journal of Distributed Sensor Networks*, *18*(3), Article 15501329221088450.

O'Hagan, A., & Garlington, A. (2018). Counterfeit drugs and the online pharmaceutical trade, a threat to public safety. *Forensic Research & Criminology International Journal*, *6*(3), 151–158.

Oham, C., Michelin, R. A., Jurdak, R., Kanhere, S. S., & Jha, S. (2021). B-FERL: Blockchain based framework for securing smart vehicles. *Information Processing & Management*, *58*(1), Article 102426.
Otoum, S., Al Ridhawi, I., & Mouftah, H. (2022). Realizing health 4.0 in beyond 5G networks. In *ICC 2022-IEEE international conference on communications* (pp. 2960–2965). IEEE.
Parker, R. G., Sommer, M., et al. (2011). *Routledge handbook of global public health*. Routledge Abingdon, UK.
Pawar, P., Parolia, N., Shinde, S., Edoh, T. O., & Singh, M. (2022). EHealthChain—a blockchain-based personal health information management system. *Annals of Telecommunications*, *77*(1), 33–45.
Putz, B., Dietz, M., Empl, P., & Pernul, G. (2021). Ethertwin: Blockchain-based secure digital twin information management. *Information Processing & Management*, *58*(1), Article 102425.
Spataru, A. L., Pungila, C.-P., & Radovancovici, M. (2021). A high-performance native approach to adaptive blockchain smart-contract transmission and execution. *Information Processing & Management*, *58*(4), Article 102561.
Stamatellis, C., Papadopoulos, P., Pitropakis, N., Katsikas, S., & Buchanan, W. J. (2020). A privacy-preserving healthcare framework using hyperledger fabric. *Sensors*, *20*(22), 6587.
Tanwar, S., Parekh, K., & Evans, R. (2020). Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications*, *50*, Article 102407.
Uddin, M. (2021). Blockchain medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry. *International Journal of Pharmaceutics*, *597*, Article 120235.
Venhuis, B. J., Keizers, P. H., Klausmann, R., & Hegger, I. (2016). Operation resistance: a snapshot of falsified antibiotics and biopharmaceutical injectables in europe. *Drug Testing and Analysis*, *8*(3–4), 398–401.
Xu, X., Sun, G., Luo, L., Cao, H., Yu, H., & Vasilakos, A. V. (2021). Latency performance modeling and analysis for hyperledger fabric blockchain network. *Information Processing & Management*, *58*(1), Article 102436.
Zaabar, B., Cheikhrouhou, O., Jamil, F., Ammi, M., & Abid, M. (2021). HealthBlock: A secure blockchain-based healthcare data management system. *Computer Networks*, *200*, Article 108500.
Zhang, G., Yang, Z., Xie, H., & Liu, W. (2021). A secure authorized deduplication scheme for cloud data based on blockchain. *Information Processing & Management*, *58*(3), Article 102510.
Zhao, Q., Chen, S., Liu, Z., Baker, T., & Zhang, Y. (2020). Blockchain-based privacy-preserving remote data integrity checking scheme for IoT information systems. *Information Processing & Management*, *57*(6), Article 102355.

Ghassan Al-Sumaidaee was born and raised in Iraq. Before arriving at McGill University, he earned a master's degree at Al-Ahliyya Amman University, focusing on the Machine Learning field, more precisely, protecting networks using hybrid Artificial Neural Networks. Ghassan plans to continue his work in related areas in Canada (currently a Ph.D. candidate at McGill) and is interested in investigating the possible techniques that increase blockchain security in the Canadian healthcare industry. Ghassan's research has been recognized by several international conferences and published by IEEE and Springer. ghassan.al-sumaidaee@mail.mcgill.ca ResearchGate: https://www.researchgate.net/profile/Ghassan-Al-Sumaidaee.



Rami Alkhudary is an Assistant Professor of Logistics and Information Systems at Paris-Panthéon-Assas University, where he recently completed his Ph.D. with a thesis on the use of blockchain technology in supply chains. Rami's current research focuses on new technologies in supply chains and project management. His research has been recognized by several international conferences (Best Research Paper at PROLOG 2019) and published by Springer, International Journal of Project Management, European Business Review, Supply Chain Forum, and Harvard Business Review France. He is also a reviewer for several international journals and conferences. Rami.Alkhudary@u-paris2.fr ResearchGate: https://www.researchgate.net/profile/Rami-Alkhudary-2. .

Zeljko Zilic is Professor at McGill University, Montreal, QC, Canada, undertaking research on Internet of Things and Blockchain and their applications. Prior to McGill, he was a Member of Technical Staff at Lucent Technologies in Allentown, PA, USA, involved in architecting, designing, verifying and testing ORCA FPGAs. He has a Ph.D. and M.Sc. in Electrical and Computer Engineering from University of Toronto. Over the years, he researched various areas of computer engineering, resulting in 300+ peer-reviewed publications and three books. He has received 5 Best Paper Awards and several Honorary Mention Awards from international conferences. He has supervised over 95 graduate students and postdocs.

Andraws I. Swidan is a professor at the computer engineering department at University of Jordan and visiting professor at McGill University, Montreal Canada. Earned all his graduate degrees from Leningrad electro-technical institute, Leningrad (Saint-Petersburg) Russia (Soviet Union) 1979 and 1982. He has authored and co-authored tens of papers in international peer-reviewed journals and attended several international conferences. His research interests are modular arithmetic and information security.