cryptography

Article

Connected Blockchain Federations for Sharing Electronic Health Records

Faiza Hashim [1], Khaled Shuaib [1,*] and Farag Sallabi [2]

[1] Department of Information Systems and Security, College of Information Technology, United Arab Emirates University, Al Ain P.O. Box 15551, United Arab Emirates

[2] Department of Computer and Network Engineering, College of Information Technology, United Arab Emirates University, Al Ain P.O. Box 15551, United Arab Emirates

* Correspondence: k.shuaib@uaeu.ac.ae

Abstract: With the growing utility of blockchain technology, the desire for reciprocal interactions among different blockchains is growing. However, most operational blockchain networks currently operate in a standalone setting. This fragmentation in the form of isolated blockchains creates interoperability difficulties, inhibiting the adoption of blockchains in various ecosystems. Interoperability is a key factor in the healthcare domain for sharing EHRs of patients registered in independent blockchain networks. Each blockchain network could have its own rules and regulations, obstructing the exchange of EHRs for improving diagnosis and treatments. Examples include patients being treated by healthcare providers in different countries or regions, or within one country but with a different set of rules per state or emirate. By contrast, a federation of blockchain networks can provide better communication and service to stakeholders in healthcare. Thus, solutions for facilitating inter-blockchain communication in such a blockchain federation are needed. However, this possibility has not been fully explored, and further investigations are still being conducted. Hence, the present study proposes a transaction-based smart contract triggering system for inter-blockchain communication, enabling EHR sharing among independent blockchains. We use local and global smart contracts that will be executed once a transaction is created in the blockchain. Local smart contracts are used for EHR sharing within the blockchain, whereas global smart contracts are used for EHR sharing among independent blockchains. The experimental setup is conducted using the Hyperledger Fabric blockchain platform. Inter-blockchain communication between two independent fabric networks is conducted through a global smart contract using Hyperledger Cactus for EHR sharing in a health federation setup. To the best of our knowledge, our study is the first to implement an inter-blockchain communication model in the healthcare domain.

Keywords: blockchain integration; blockchain communication; smart contracts; electronic health records (EHRs); blockchain federation

Citation: Hashim, F.; Shuaib, K.; Sallabi, F. Connected Blockchain Federations for Sharing Electronic Health Records. *Cryptography* **2022**, *6*, 47. https://doi.org/10.3390/cryptography6030047

Academic Editor: Kentaroh Toyoda

Received: 10 July 2022
Accepted: 12 September 2022
Published: 16 September 2022

1. Introduction

An electronic health record (EHR) is an important asset of a patient in the healthcare environment and is solely owned by the healthcare entities visited by that patient. An EHR comprises critical and highly sensitive health data for patient diagnosis and treatment [1,2]. One report [3] stated that, on average, a patient in the United States visits 18.7 different caregivers (CGs) and holds nearly 19 unique medical records during their lifetime. Hence, patient data are scattered at different locations. Such data need to be shared among healthcare providers to maintain patients' unique records. This can help healthcare individuals to easily access patient data to avoid repeated laboratory tests when visiting different healthcare providers, thereby providing patients with better treatment and diagnosis [4,5]. However, each healthcare entity has its own rules and regulations, representing the main obstacle in sharing patient records among the concerned entities to facilitate patient treatment and diagnosis, thus, causing the process of sharing to be excessively difficult and lengthy. Therefore, technology that facilitates the sharing of patients' EHRs among healthcare providers is needed.

Recently, blockchain has been investigated in the research community as a technology for addressing interoperability issues in healthcare. Blockchain technology can provide numerous benefits to the healthcare system, including a decentralized data-sharing setup allowing interoperability, security, authentication, and integrity [6,7]. Therefore, several researchers have adopted and investigated the use of blockchain technology in healthcare [8–14]. As a result, many interoperability issues among various healthcare providers have been resolved through the development of a blockchain network. However, at present, many blockchains operate independently within the same state, country, and continent. Patients traveling between these locations need to share their EHRs across different blockchains. Given the high demand and adoption of blockchain in the healthcare environment, the research community has recently focused on investigating interoperability among independent blockchains.

Interoperability is a key factor that allows multiple blockchains to exchange data, even if they have different consensus rules and platforms in a blockchain federation. Such interoperability is highly desirable in a healthcare blockchain federation for improved diagnosis and treatment of patients, considering that EHRs are scattered across different blockchains of the healthcare environment. Blockchain federation is a model for integrating multi-blockchain functionalities. Two types of federations are found in the literature: homogeneous and heterogeneous blockchains. Homogeneous refers to a federation comprising independent blockchains of the same type (platform, consensus protocol, public/private) that are built according to the same architectural rules but with each blockchain developed on different business logic. By contrast, heterogeneous pertains to a federation comprising independent blockchains of different types that are built based on different rules and operating mechanisms. In both cases, no interaction exists among blockchains, with each operating independently. Developing methods for interoperability among independent healthcare blockchains is an active field of research. However, high barriers exist between homogeneous and heterogeneous blockchains for sharing data. Inter-blockchain communication achieves secure and effective solutions for interoperability in a blockchain federation. Although multi-blockchain technology is in its infancy, initial frameworks, models, and architectures can be found in the literature, including sidechain-based [15–17], router-based [18–20], and smart contract-based solutions [21–23]. Each solution possesses challenges to be addressed and provides a functional inter-blockchain communication model for the healthcare domain.

Sharing health records among the various stakeholders of the healthcare ecosystem is important, which includes individuals (patients and their doctors) and other stakeholders (insurance companies/research centers). EHR sharing is an imperative step in escalating the interoperability of healthcare providers and ensuring that the healthcare system is smart and efficient. Interoperability is a key factor in the healthcare domain for sharing EHRs of patients registered in independent blockchain networks. Each blockchain network could have its own rules and regulations, which could be a barrier towards exchanging EHRs for improved diagnosis and treatments when needed. Examples include patients being treated by healthcare providers in different countries or regions, or within one country but with a different set of rules per state or emirate. Therefore, a system for inter-blockchain communication is necessary to better serve the needs of stakeholders.

Inter-blockchain communication allows independent blockchains within a federation to communicate directly with each other and trade assets. Connected blockchains do not have to communicate directly with each other; instead, they send information packages (transactions) via dedicated channels using smart contracts. Smart contracts are used by each blockchain. The sender and receiver blockchain deploys a transport, authentication, and ordering (TAO) module, which implements and operates based on the configured smart contracts at each blockchain. The inter-blockchain communication process utilizes relayers to transfer transactions via secure established communication channels, which are dedicated links configured based on smart contracts between interacting blockchains. The TAO module transmits and authenticates the transaction flow from one blockchain to another.

Currently, many advancements are in place for inter-blockchain communication, and the research community is achieving progress with the proposal of various solutions. However, these solutions do not completely address inter-blockchain communication. In particular, implementation of the frameworks and architectures is lacking in the proposed solutions. Thus, our study contributes to developing state-of-the-art inter-blockchain communication by proposing a possible solution to the challenges of interoperability in healthcare blockchain federations for EHR sharing. The main contributions of this research include the following:

- Development of a novel healthcare blockchain integration model using transaction-based inter-blockchain communication for EHR sharing in a federation of independent blockchains.
- The use of local and global smart contracts to establish communication links and transaction flow in a blockchain federation.
- Implementation of independent blockchains in healthcare, which represent the first example where an inter-blockchain communication model is implemented for healthcare data sharing. Two Hyperledger Fabric networks are used that operate independently, with each running different business logic. Both networks are integrated for inter-blockchain communication to enable EHR sharing among them.
- Defining a set of metrics used to evaluate the performance of independent blockchains and the derived inter-blockchain communication model while demonstrating improvements in performance results compared with previous work.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 presents the proposed inter-blockchain communication model. Section 4 discusses performance evaluation and implementation details. Section 5 presents the experimental results and discussion. Finally, the study is summarized and concluded in Section 6.

2. Literature Review

Since the advent of bitcoin in 2009 [24], we have seen a massive adoption of blockchain in many domains, including banking and finance [25,26], supply chain management [27,28], IoT [29,30], healthcare [31–33], and cloud computing [34,35]. These blockchains operate in closed-ended silos, unable to communicate with each other. However, with the growing utility and advancement of blockchain technology, inter-blockchain communication opens up a literal world of possibilities, allowing blockchains to interoperate and transfer value, interchange assets and services, and connect for sharing data. In this section, research trends in inter-blockchain communication are outlined. The inter-blockchain communication solutions are categorized as sidechains, blockchain routers, and smart contract solutions.

Sidechains are secondary blockchains that are connected to other blockchains, and the main chain allows the bidirectional transfer of data among different blockchains [36]. Reference [37] was the first to introduce the idea of a sidechain to facilitate transactions between bitcoin and other cryptocurrencies, and vice versa, using two-way peg chains. Thus, users have the flexibility to access various cryptocurrency systems by using the assets they already own. A two-way peg is a technique for the bidirectional transfer of data between the main chain and the side chains at a predefined exchange rate. A Rootstock (RSK) platform [16] was developed and operated as a bitcoin sidechain. In this implementation, when bitcoins are transferred to RSK blockchains, they become “smart bitcoins” and are equivalent to bitcoins. They can be transferred back into bitcoins for a standard transaction fee. Both blockchains (main chain and sidechain) use proof-of-work (PoW) to perform combined mining and generate blocks. However, the federated pegs used in RSK suffer from political centralization. Moreover, being a sidechain, RSK does not operate as an independent blockchain. Reference [15] proposed a blockchain architecture of satellite chains that form interconnected but independent subchains of a single blockchain system. In this design, nodes can join a given satellite chain according to their choices and requirements. Each satellite chain maintains its own private ledger, which cannot be accessed by other satellite chains in the network. Reference [38] proposed a federated two-way peg mechanism in a sidechain solution. In this configuration, the entire federation collectively maintains custody of locked funds and mutually validates the fund transfers between the main blockchain and its sidechain with a majority consensus. This work provides increased security to the fund's transfer processes. However, the federated two-way peg mechanism increases the time for validating the transactions. Plasma [17] is a sidechain developed for Ethereum. Each sidechain has independent rules and constraints imposed through smart contracts. Plasma chains used the proof-of-stake (PoS) consensus algorithm. However, the mining process is performed on the main chain, making Plasma dependent on its main chain for the mining process.

A blockchain router approach requires some of the network participants to act as routers to transmit data among diverse blockchain networks [39]. In this setup, the requests are sent and received via designated router nodes of each blockchain in the network. Reference [40] proposed a blockchain router that allows communication among various blockchains via their router nodes. Their proposed model comprises four participants: validator, surveyor, nominator, and connector. The validator verifies, concatenates, and forwards blocks to the correct destinations. The nominator contributes their own funds to the validator and is then rewarded. The surveyor monitors the blockchain router behavior. The connector links the blockchain router with subchains. A design was presented in this report, but technical details of the technique were not provided. Reference [41] introduced a private token-based inter-blockchain communication without any mediators using a routing algorithm and practical byzantine fault tolerance (PBFT) protocol. However, the network throughput was degraded as the connected blockchains have diverse topologies. Reference [18] proposed interactive multiple blockchain architecture to support inter-blockchain communication among heterogeneous blockchains using routing management and message transfer protocol. This model operates in four layers: basic, blockchain, multi-chain communication, and application layers. The paper also introduced a unified packet for the transaction and routing among multiple blockchains. Reference [19] proposed interchain as a framework for inter-communication among any pair of blockchains. The proposed framework comprises subchain, interchain, validating, and gateway nodes. A three-way handshaking technique was employed for asset transfer among the connected blockchains. However, the paper did not mention the consensus algorithm adopted by the framework.

Next, smart contract-based models to create interoperable protocols among multiple blockchains are discussed. Reference [20] proposed a solution consisting of a smart contract to allow data sharing among various independent blockchains. As a proof of concept, their model transacts on Ethereum public and private blockchain networks. However, the authors did not apply their solution to two hybrid systems. Reference [42] proposed a cross-chain atomic swap for asset exchange among various participants across multiple Ethereum blockchains. Reference [43] provided a mechanism for cross-blockchain data transfer, smart contract interaction, and currency transfer. They proposed transferring the same type of token to any number of blockchains simultaneously. However, their protocol can only be operated in the same kind of environment (Ethereum). Reference [22] proposed a smart contract-based interoperability solution between independent blockchains (public and private) without intermediaries. However, the authors did not apply their solution to two hybrid systems.

In this section, various solutions for inter-blockchain communications for homogenous and heterogeneous blockchains along with their limitations are analyzed in Table 1. Sidechain solutions are widely adopted in the literature. However, the major drawback of techniques using sidechains is the one-to-one communication among homogeneous blockchains. Furthermore, in the implementation of a sidechain, security vulnerabilities in the blockchain federation increase when a sidechain in the network is compromised. Blockchain routers provide connectivity solutions for heterogeneous blockchain networks, but none have been implemented yet. Furthermore, such implementations require that the architecture of the router nodes is configured to function as routers. Another limitation of the blockchain router technique is the single point of failure issue. Hence, when any router node fails, communication among any participating networks will be compromised. The healthcare domain is a highly in-demand field that requires solutions for inter-blockchain communication in a blockchain federation. However, this area of research has not been fully explored, and further investigations are needed. Therefore, this research proposes a novel "transaction-based inter-blockchain communication" technique based on global and local smart contracts in a healthcare federation to address the interoperability challenges among independent blockchains.

Table 1. Summary of the projects reviewed for inter-blockchain communication solutions.

Index	Consensus	Features (+/−)	Solution Type	Shortcomings of the Solution
[15]	PoW	(+) Works as sidechain pegged to bitcoin. Faster transaction validation, lower transaction fee. (−) Mining is performed on the main chain, completely dependent on the main chain for the mining process.	Side chain solution	Supports 1–1 communication among sidechains and main chain. Focus on homogenous blockchains. Higher computational cost and complex Sidechain blockchains cannot operate independently.
[17]	PoS	(+) Each sidechain has its own independent rules and constraints. (−) Mining is performed on the main chain, completely dependent on the main chain for the mining process.		
[38]	Heterogeneous consensus algorithms	(+) Sidechains use independent consensus algorithms. Maintains private ledger, which provides faster block generation. (−) The private ledger is not shared with all participants.		
[39]		(+) Uses a federated two-way peg mechanism, provides increased security to the funds transferred among sidechains and main chain. (−) The federated two-way peg mechanism increases the transaction validation time.		
[18]	Delegated Stake-PBFT	(+) Provides communication among heterogeneous blockchains. Can dynamically add blockchain routers. (−) Communication via blockchain router only. One-point failure issue can compromise communication.	Blockchain router solution	Design and frameworks available but are not yet implemented. The configuration of blockchain node needs to be changed to function as router node. One-point failure issue. Communication is affected as the router node fails or compromised.
[19]	PBFT	(+) Different blockchain systems communicate without any intermediaries. Using ANN-router-based network architecture, a part of the blockchain can function as router, however, configuration details of such setup are required. (−) The connection mechanism is not provided. Based on each blockchain topology, throughput is affected. Implementation details are missing.		
[20]		(+) Created a dynamic blockchain network called router blockchain, which includes router nodes from each blockchain. (−) One-point failure issue due to communication via a single node. The configuration setting of router node is not provided.		
[22]	PoS	(+) Smart contract-based interoperability solution between independent blockchains (public and private) without intermediaries. (−) The authors did not apply their solution between two hybrid networks.	Smart contract solutions	Available solutions operate in homogeneous blockchains. Smart contract solutions in infancy and implementations not available. Smart contract sharing not available.
[43]		(+) Cross-blockchain data transfer, smart contract interaction, currency transfer. Transfer same kind of token any number of blockchain simultaneously. (−) Proposed protocol operates in same environment only among homogeneous blockchains.		
[23]		(+) A cross-chain atomic swap is used for assets transfer across multiple participants between multiple Ethereum blockchains. (−) Need to implement atomic swaps on and with other blockchains.		

3. Methods

This section proposes a transaction-based smart contract triggering system in inter-blockchain communication for EHR sharing among independent blockchains, as shown in Figure 1. In this setup, each blockchain holds a unique blockchain ID (e.g., B1, B2, B3, and B4) that is preregistered with an overarching global authority (GA), such as the Ministry of Health. Our system consists of several nodes that can take any of the following roles: hospitals that are full nodes for executing transactions (requesting and granting access to a patient record), patients who can only view their medical record, allied health professionals who can request patients' EHRs, validators who participate in the consensus process, and regulators who enforce policies and handle registration of nodes to establish connections (e.g., the certification authority (CA) for each blockchain) without necessarily participating in the consensus process.

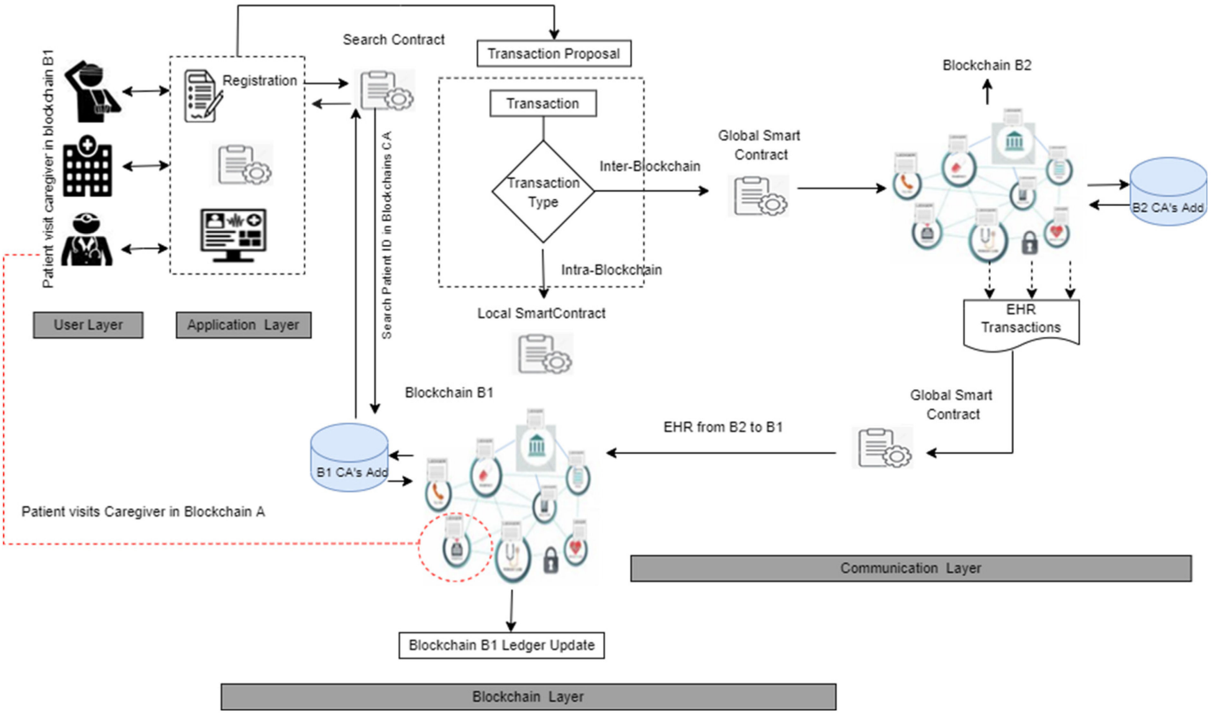


Figure 1. Proposed blockchain federation model for EHR sharing.

The proposed architecture consists of four layers: user, application, blockchain, and communication layers. The user layer consists of clients who interact with the blockchain network using a decentralized application, where each node will conduct transactions directly with its peers in the network. The application layer monitors the registration process of the participants in the blockchain and generates the encryption keys of registered users. In this architecture, the application layer triggers a search contract after registration of a patient to search for their record in other blockchains of the federation. The blockchain layer comprises the network's core components, including network participants, consensus mechanism, and smart contracts. The communication layer uses global smart contracts to create communication links to other blockchains for data sharing in a blockchain federation.

3.1. User Layer

Participants of a blockchain network need to install the used decentralized application to access the functionality of the blockchain network using their own devices. New participants will need to provide their credentials for registration to the network, while registered participants can use their login information to sign into the network.

3.2. Application Layer

Registration of all users in the blockchain network is required. After registration, public and private key pairs are generated for each user through a CA of the user's blockchain. If the user is already a registered node in the blockchain network, its key pair and registration details would be already stored with the CA and will be used for proof-of-identity of the network users. Hence, the CA of each blockchain network stores the details of registered nodes in the blockchain. In the proposed technique, each blockchain maintains a registry of addresses for all the CAs of the blockchain federation network. When a patient is registered in the blockchain, after completing the registration process, a search contract is triggered to check the patient's ID in CAs. The patient's ID is sent to the CAs of the blockchain entities within the federation. The blockchain CA checks if the patient's ID exists in its database and responds accordingly. The search contract uses the patient's ID because each blockchain CA generates a separate public/private key pair for the same patient. Hence, the application layer automates the process of finding the blockchains where a patient's EHRs exist. Then, the communication layer enables the creation of needed communication links to the targeted blockchains within the federation.

3.3. Blockchain Layer

A consortium blockchain is a federated blockchain, where multiple healthcare entities (including hospitals, pharmacies, insurance companies, laboratories) govern the network. A collaborative environment is formed, such that every entity contributes to the network, which facilitates the sharing of EHRs among multiple entities within the blockchain network. The proposed system is based on multiple consortium blockchains running independently in a healthcare blockchain federation. In this case, we consider homogenous blockchains that can communicate within the federation to access the EHRs of a patient registered in one or more blockchains. All involved blockchains are permissioned blockchains that are preregistered with the GA. The network participants are considered trusted authorities of the federation to participate in various functions, including the consensus process and accessing and updating patients' EHRs. Consensus is the core of any blockchain network and in a federation. The choice of a consensus algorithm depends on the network platform being used for the implementation of the blockchain network. Local and global smart contracts are used in this architecture to access patients' EHRs within the same blockchains and across the independent blockchains of the federation.

3.3.1. Smart Contracts

Smart contracts play a vital role in blockchain operations. Smart contracts are programable modules stored on a blockchain that are triggered when predetermined conditions are met. They can also automate a workflow, triggering the next action when conditions are met. Smart contracts automate the execution of a condition or an agreement so that all network nodes can be promptly acknowledged of the outcome without the involvement of any mediators. The proposed blockchain layer entails three types of smart contract: search, global smart, and local smart contracts, as shown in Figure 2.

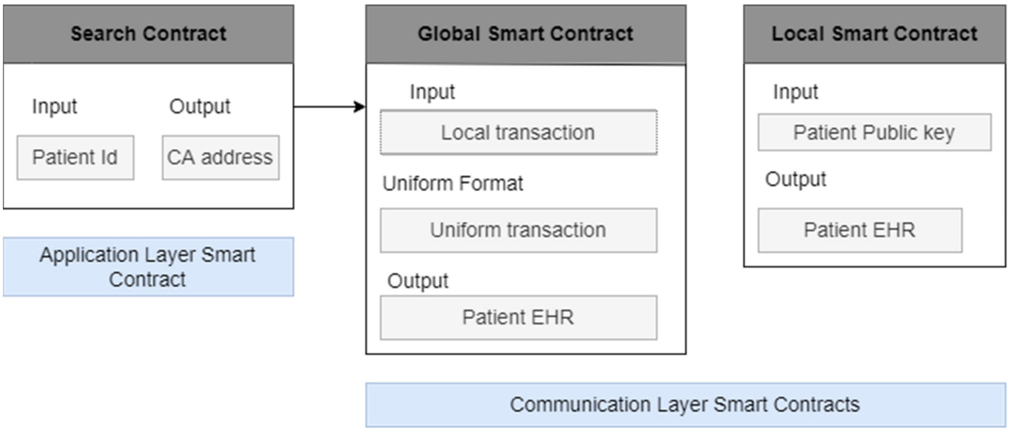


Figure 2. Proposed smart contract architecture.

Search Contract: The search contract is triggered at the application layer after the patient is registered in the blockchain network. The functionality of this contract involves searching for a patient ID in the CA address registry to identify the blockchain in which their EHRs exist. The input in this contract is the patient ID, as it is identical in all blockchains within the federation. The CA of each blockchain generates a public/private key pair based on this ID.

Global Smart Contract: Global smart contract is triggered at the communication layer when the transaction type in the transaction proposal prepared by the current CG is identified as "inter-blockchain." This contract allows communication among independent blockchains in a federation to share the EHRs of patients under observation.

Local Smart Contract: In the proposed architecture, a local smart contract is triggered when the transaction type to access a patient's EHRs within the same blockchain being currently visited by the patient is "intra-blockchain."

Data Contract: The data contract provides CGs with the functionality to add data to the blockchain. The EHRs of patients are stored in IPFS and the hash of these records is stored in a data contract that can be easily accessed by authorized nodes.

3.3.2. Decentralized Off-Chain Storage

Based on their storage capacity, blocks in blockchain are unable to store a huge volume of data on-chain [25]. Therefore, the proposed model uses IPFS, a decentralized file system, for off-chain EHR storage. IPFS provides a distributed storage structure to store copious amounts of medical records and shield the system from DoS attacks, one-point failure, and enhance data integrity. It uses the content-addressing hash to uniquely identify each file stored in the system. In the blockchain network, the CGs add the EHR of the patient to the IPFS, and the content-addressing hash of the EHR is then stored on-chain in a blockchain block that is used for accessing the EHR from IPFS.

3.4. Communication Layer

Based on the output of the search contract, the communication layer establishes communication links between the current blockchain and other blockchains of the federation through a secure communication channel where patients' EHRs are found.

In the example shown in Figure 1, patient P visits blockchain B1 for an appointment with a CG. Here, we assume that patient P has previous EHRs registered in another blockchain, B2, as provided by the search contract in the application layer. To provide treatment, the current CG requires their previous EHRs for better diagnosis and to avoid repeated tests that have already been performed in their previous visits. In the proposed architecture, patient centricity is provided using a consent form signed by the patient (or their attendee in case of an emergency, on behalf of the patient). The signed consent form is encrypted using the current CG's private key. On the receiving side, the consent form will be validated using the CG's public key, from blockchain B1, before providing the record by nodes in B2. A transaction proposal refers to executing a specific function on the smart contract, for example, invoking a "ReadPatient" function to access the EHR of the patient in the blockchain network. The current CG prepares a transaction proposal, setting the transaction type as inter-blockchains, as shown in Figure 1. This type of transaction, "inter-blockchain", triggers the global smart contract to create a communication link with blockchain B2 using the B2 CA address.

3.4.1. CA Chain

Blockchain networks use public key cryptography for encrypting and decrypting information on the distributed network. Public Key Infrastructure (PKI) is a public key management environment in a public key cryptographic system. PKI uses two mathematically related keys for encryption and decryption. In public key cryptography, one key is used to encrypt/decrypt the information, and the second key carries out the reverse operation. The private key is kept secret, whereas the public key can be handed out to any member of the network. In the PKI, a certificate is generated to bind a specific identity to a specific public key and information about how the public key may be used. CAs are trusted entities that issue certificates to PKI users and provide information on the status of certificates issued by the CA.

Within the blockchain federation, each blockchain operates with its PKI. In this study, we need to communicate with different blockchains, so cryptographic functions will be carried out among different public keys for the participants of each blockchain. According to [44], isolated CAs can be combined to form larger PKI. This CA combination is created at the communication layer during the integration of B1 and B2 for sharing patient records. In the proposed architecture, a superior–subordinate relationship, referred to as a hierarchical PKI, is used to create a single PKI. In this relationship, a CA is defined as a root CA, and all users of the hierarchical PKI begin certification with the root CA. In such a scenario, choosing the root CA is a daunting task. In this study, we approach this problem by selecting the current blockchain CA as the root CA, which will establish a link with other blockchains in the federation. The connected blockchains will function as subordinate CAs and will use the root CA certificates and public keys, as shown in Figure 3.

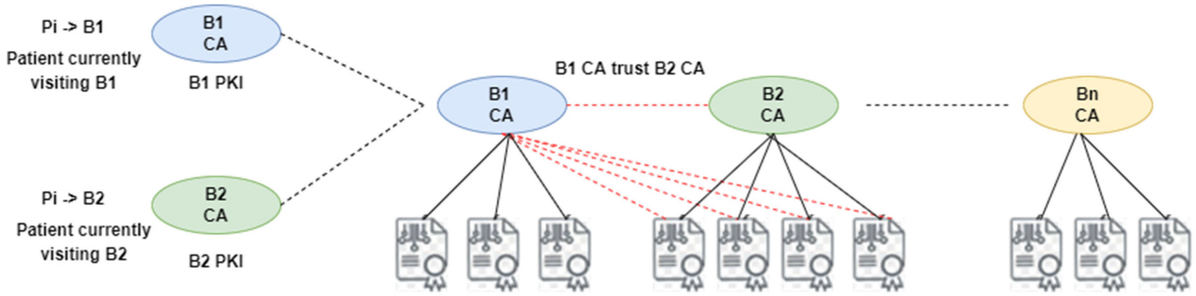


Figure 3. Proposed hierarchal PKI architecture.

3.4.2. Patient Record Retrieval

The global smart contract of B1 transfers the patient record request (transaction) to blockchain B2 through the communication module. B1′s query transaction is broadcasted within B2 nodes, where each node verifies the patient's consent form using B1′s CG public key, which was sent from the requesting node in B1 (CG). After verification, nodes search their database for the patient's EHRs and broadcast a transaction of the patient's EHRs to the network. Each node in the B2 blockchain that holds the patient's EHRs in its registry uses the public key of the current CG of B1 to encrypt the patient's EHRs. The encrypted EHRs are added to the InterPlanetary File System (IPFS), and a hash of the EHR address in IPFS is generated and added to the transaction. The process for retrieval of the patient's EHR is provided in Algorithm 1.

Algorithm 1. Patient Electronic Health Record Retrieval

```
Smartcontract: DataRequest ($P_{id}$)
DataRequest ($P_{id}$)                          # $P_{id}$ = Patient id
if (msg.sender = Authorized $B_N$) then
    $P_{pk} \leftarrow P_{id}$
    Search ($P_{pk}$)
        if $P_{pk}$ == found then
            return ($PR$)                       #PR = Patient Record
    else
        AbbortSession
    end if
end if
$PR \rightarrow CG_{pk}.Encrypt$
$E(PR) \rightarrow IPFS$
$IPFS \rightarrow H(E < PR >)$
$H(E < PR >) \rightarrow Trans$
$Trans \rightarrow broadcast$
```

Byzantine fault tolerance (BFT) consensus algorithms are widely used in permissioned consortium blockchains. They rely on the message-based scheme for faster block generation compared with PoW [45,46]. In this research, we assume that the blockchains in a healthcare federation are using BFT-based consensus protocols, such as PBFT [47] and proof-of-authority [12,48]. Therefore, a leader node is selected based on the protocols adopted by each blockchain in the network for the transaction's validation process. For example, the leader in B2 adds transactions with the query field to a block and sends the block to validators to validate the transactions within the B2 blockchain network. The leader then waits for commitment messages from the validating nodes. After receiving the required confirmations, the block is validated based on the consensus algorithms used by the blockchains. Moreover, the leader adds a transaction type as "inter-blockchain" and will replace the "query" field with "Ack", specifying to the receiver's CA that the request is fulfilled. The transaction type triggers the global smart contract of blockchain B2. The global smart contract sends the transaction to blockchain B1′s CA address. The CA of B1 receives the transactions from B2 and broadcasts in B1. In B1, we assume that the current CG is the leader node for the current appointment, as in [12]. The current CG adds all the transactions (local and global) of the patient under observation in a block, runs the blockchain B1 consensus protocol, and updates its ledger. Once the encrypted patient's EHRs from B2 are received by the current CG in blockchain B1 ledger, the patient's EHRs are decrypted using the CG's private key. A step-by-step process of the communication layer is shown in Algorithm 2.

Algorithm 2. Inter-Blockchain Communication

```
 Patient $P_i$ visits Healthcare Professional $D_i$ in blockchain $B_i$ having EHR in blockchain $B_j$
$D_i$ issue transaction $\langle$T$_{ran}$ type, $P_{ID}$, $D_{PK}$, C_form, $S\_CA_{ADD}$, R_$CA_{ADD}$, TS, DS $\rangle$ $in$B$_i$
$B_i$ nodes validate $T_{ran}$
Wait for $\{T_{ran}\}$ from $B_j$
if ($T_{ran}$ type == inter blockchain) then
    Trigger GS                 # Global Smart Contract
    Procedure Request _Connection ($S\_CA_{ADD}$, R_$CA_{ADD}$)
        Create connection $S\_CA_{ADD} \rightarrow$ R_$CA_{ADD}$
    Procedure Request Transaction ($T_{ran}$, $S\_CA_{ADD}$, R_$CA_{ADD}$)
        Procedure Request Transaction ($T_{ran}$, $S\_CA_{ADD}$, R_$CA_{ADD}$)
end if
$CA\_$B$_j$ $Validate$ T$_{\mathrm{ran}}$
if (T$_{\mathrm{ran}}$ ==TRUE) then
    $Tran \rightarrow$ < T$ran$ + $query$>
end if
CA_$B_j$ $\langle$ T$_{ran}$ $\rangle$ $\rightarrow$B$_j$ Nodes
Procedure Search$record$ (T$_{ran}$, $B_j$ Nodes)
  Each node in $B_j$ search ($P_{ID}$ in PR)
    if ($P_{ID}$ found) then
        Issue $Tran$ $\langle$T$ran$ type, $P_{ID}$, $S\_CA_{ADD}$, R_$CA_{ADD}$, Hash(EHR), TS, DS, Ack>
  end if
$B_j$ Nodes validate $T_{ran}$
if ($T_{ran}$ type == inter blockchain) then
  Trigger GS
  if (Query == Ack) then
      Procedure Transfer $T_{ran}$ ($T_{ran}$, R_$CA_{ADD}$)
  end if
end if
$\forall$T$_{ran}$ $\rightarrow$T$_{ran}$ pool of $B_i$
Wait = NULL
$D_i$ adds $T_{ran}$ to block
Procedure Consensus (block, $B_i$ nodes)
  Consensus protocol
Update ledger $B_i$
Appointment = NULL
```

4. Performance Evaluation

In this section, we conducted several experiments to individually evaluate the performance of both blockchains (B1 and B2) and then evaluate the average response time of query transactions from blockchain B1 to B2. The following sections discuss the details of the performance evaluation of the proposed model in terms of evaluation metrics and experimental environment.

4.1. Evaluation Metrics

The proposed model performance is evaluated through the following evaluation metrics, which are relevant to the implemented blockchain scenarios and are typically used when looking at evaluating a blockchain network:

Scalability: The scalability of blockchain networks is the platform's ability to support the increasing transaction load, including the increasing number of nodes on the network. It indicates the acceptability of the network performance while varying the number of nodes and transaction load.

Throughput: Throughput refers to the total number of transactions confirmed per second in the blockchain network.

Transaction Latency: Latency measures the time for an issued transaction to be completed and for a response to be available to the application issuing the transaction.

CPU Utilization: CPU utilization measures CPU consumption on participating nodes in a network.

Average Elapsed Time (ET): Average elapsed time measures the average elapsed time for query transactions from one blockchain network to another, that is, from blockchain B1 to B2. For example, B1′s ET can be calculated from the start time (T_s) of a client request initiated from B1 to the time the client received the response from B2 (T_R), such that

$$ET = T_R - T_s$$

In the communication process, the ET depends on the query processing time (QT) by B2's round trip communication time (CT) from B1 to B2. Hence,

$$ET = CT + QT_{B2}.$$

The QT at blockchain B2 can be calculated as follows:

$$QT_{B2} = RT_{B2} - QT_s$$

where RT_{B2} represents the response time for the query by B2, and QT_s represents the query start time in B2. Then, B1 ET can be calculated as follows:

$$ET = (RT_{B2} - QT_s) + CT. \quad (1)$$

4.2. Experimental Environment (The Source Code of the Implementation Used in this Paper Will Be Provided by the First Author upon Request)

The blockchain-based framework Hyperledger Fabric [49] is used to develop two independent private blockchain consortiums for efficient data sharing in healthcare, where several health entities form a peer-to-peer consortium network. Hyperledger Fabric is a scalable blockchain platform that is widely used in a variety of contexts, including healthcare [50], IoT traceability [51], self-sovereign identity [52], digital couponing [53], and supply chain management [27]. Hyperledger Fabric is an open-source permission-based distributed ledger technology, where all the participants know each other. Therefore, the network is fully trusted and secure. In fabric architecture, all the participating healthcare entities and their end-users are notorious and registered by a CA using a membership service module. The roles in the fabric network include the following:

CA: Fabric CA takes care of the registration, issuance of electronic certificates, role assignment, renewals, and revocation to different nodes before they can start communicating online.

Peer Nodes: Peers are fundamental nodes in the network and perform multiple roles, including executing smart contracts, validating transactions provided by the clients, and maintaining a copy of the ledger.

Client Nodes: These nodes submit the transaction proposal to endorsing peer nodes and broadcast the transaction proposal to ordering nodes. Transaction proposal refers to executing a specific function on the smart contract, for example, ReadPatient, AddPatient, and RegisterPatient.

Endorser Nodes: These nodes execute a smart contract upon receiving a transaction and comply with the endorsement policies of the network.

Orderer Nodes: These nodes maintain the consistency of the state of the ledger. Orderer accepts the endorsed transaction from the client, orders them into a group of blocks with cryptographic signatures of the ordering peers, and finally broadcasts the blocks to the blockchain network.

The implementation and experimental part of the proposed model is conducted using two PCs with the following hardware and software configuration:

4.2.1. Hardware Environment

The experiments are conducted using two systems with the following hardware specifications:

- 2 Core CPU (Intel (R) Core ™ i5-4570 CPU @ 3.20 GHz);
- 8 GB RAM;
- Ubuntu OS (version 20.04.1 (TS)).

4.2.2. Software Environment

To facilitate experiments and eliminate other interference factors, both computers use the same software configuration, as given below:

- Hyperledger Fabric V2.x;
- Git 2.9+;
- Python 2.7.x;
- Npm V 5.x;
- Docker Engine 17.037;
- Docker Compose 1.8+;
- VS code;
- Hyperledger Caliper;
- Hyperledger Cactus.

The main contribution of this research is the integration of independent blockchains in a healthcare federation. To accomplish this, two independent blockchains (B1 and B2) were developed using the above configuration. Different networks will have a different number of entities in a federation, based on their requirements. For developing the test networks, we started with a minimum number of healthcare entities, owing to the limited CPU power of our system. However, we used a different number of healthcare entities in both blockchains to track the performance of both networks with a different number of nodes. B1 comprises three healthcare entities (we named them hospital-A, hospital-B, and hospital-C). Each healthcare entity has at least two peers (peer0 and peer1), one orderer, a CA for each entity, and a peer node as an endorser in the network. Blockchain 2 consists of four healthcare entities (we named them hospital-1, hospital-2, hospital-3, and hospital-4) with the same settings as blockchain B1. Both blockchains execute transactions independently in the testbed environment.

5. Experimental Results and Discussion

This section focuses on evaluating the performance of blockchains B1 and B2 with different scenarios to determine how the throughput and latency of the system change with a varying number of transactions. Second, we performed experiments for inter-blockchain communication to determine the average ET for query transactions from blockchains B1 and B2. Then, we compared the average latency for inter-blockchain and intra-blockchain transactions. We conducted a series of experiments to achieve these goals. We ran each test five times and took the average of the five tests in each scenario for our final results.

B1 and B2 are Hyperledger Fabric networks for EHR sharing in a healthcare federation. Hyperledger Caliper [54] is an open-source benchmarking tool that is used for the performance measurement of blockchain networks. In Hyperledger Caliper, the workloads or benchmarks generate the transaction that is broadcasted to the blockchain network. Caliper uses a set of independent workers to send the transaction requests to the blockchain network and monitor the response. Every worker process executes the workload generation independently. When the tests are completed, Caliper generates a performance report consisting of average throughput and maximum/minimum/average latency throughout the tests. Table 2 shows the evaluation setup environment for Caliper. We used the same setup to evaluate both blockchain networks (B1 and B2).

Table 2. Experimental parameter configuration.

Parameters	Configuration
Workers	5
Test Duration	50 sec
Rounds	5
Transaction Load per Round	500, 1000, 1500, 2000, 3000, 4000, 5000
Transactions Mode	Read
Network Size	3 Healthcare entities, 6 peers/4 Healthcare entities, 8 peers
Varied Factor	Block time

5.1. Peak Performance

We measured the peak performance of blockchains B1 and B2 while keeping the transaction load stable at 500. Figure 4 shows the peak throughput and latency of B1 and B2. The results are taken on an average of five rounds, and each round has a rate of 500 transactions. From Figure 4a, B1 and B2 test networks reach a throughput of 496.32 and 494.6 tps, respectively. Figure 4b shows an average latency of B1 and B2. Both networks exhibit a low latency of 0.868 and 0.922 s, respectively. In this basic network performance experiment, no failed transactions are recorded because of the minimal CPU utilization of the used transaction rate of 500. The results demonstrate that adding a single healthcare entity to the consortium (as in the B2 test network) reflects a minor decrease in performance of the network, that is, a difference of 1.72 tps throughput and latency of 0.054 s, which is higher in B2 than in B1.

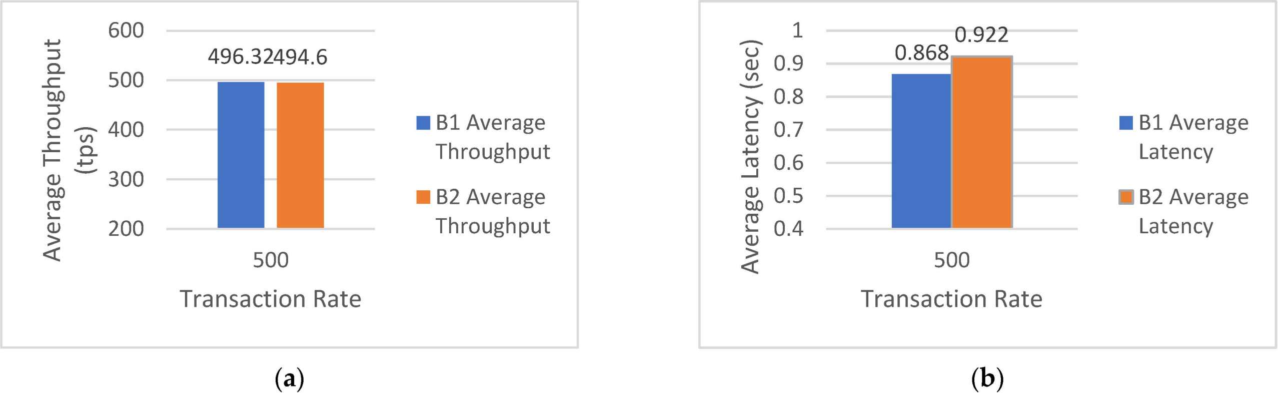
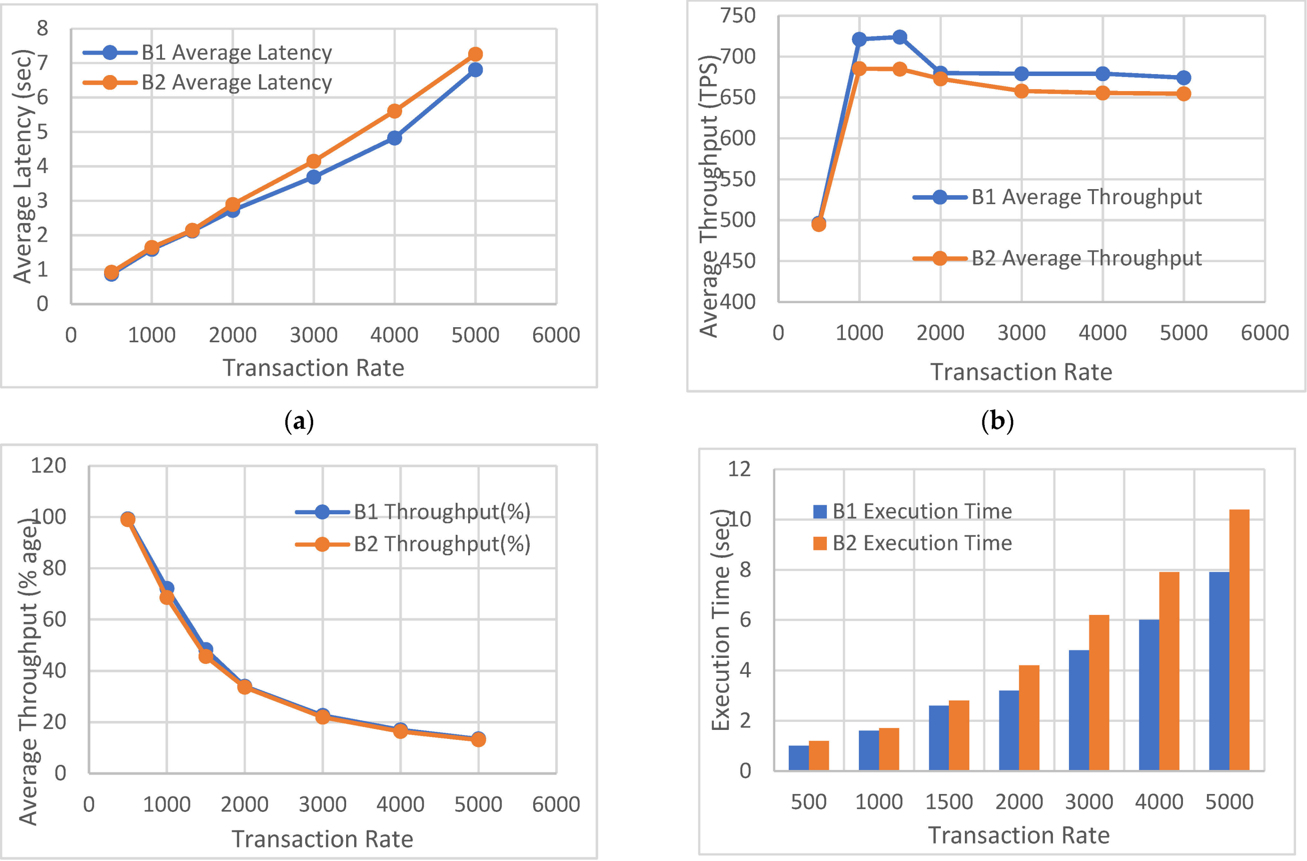


Figure 4. B1 and B2 peak performance at a transaction rate of 500 (**a**) average throughput (**b**) average latency.

5.2. Scalability Performance

In this section, we examined the scalability performance of our test networks with increasing network traffic, that is, the transaction load at a given time. Second, we used two test networks, B1 with three healthcare entities and B2 with four healthcare entities. Each network runs a consortium blockchain model. Hence, we examined the test network's performance at varying transaction loads, starting from 500, 1000, 1500, 2000, 3000, 4000, and 5000 in each round. Figure 5 shows the average throughput, latency, and execution time of both networks. Figure 5a illustrates a monotonic increase in latency of B1 and B2 as the number of transactions increases. B1 and B2 showed the same latency for transaction loads of 500, 1000, and 1500, whereas at a transaction load of 2000 or greater, B2 showed an incremental increase in latency. Figure 5b shows the throughput performance of both networks. Notably, throughput increases can be observed for transaction loads of 500 and 1000, which then drop as the CPU utilization increases to reach the maximum. Summing up the results of throughput for all transaction loads, both test networks process an average of 650–725 transactions per second. Figure 5c illustrates a monotonic decrease in transaction throughput percentage with an increasing transaction load, with performance of both networks being almost the same. However, as shown in Figure 5d, the average execution time for each transaction rate shows a gradual increase to a higher transaction rate for both networks. B2 execution time is increasing after transaction rate 1500, with the highest execution time for transaction rate 5000. However, as we increase the transaction load, B1 reports no failed transactions, whereas B2 reports an average of 339 failed transactions when reaching a maximum CPU utilization saturation point at a transaction load of 5000.



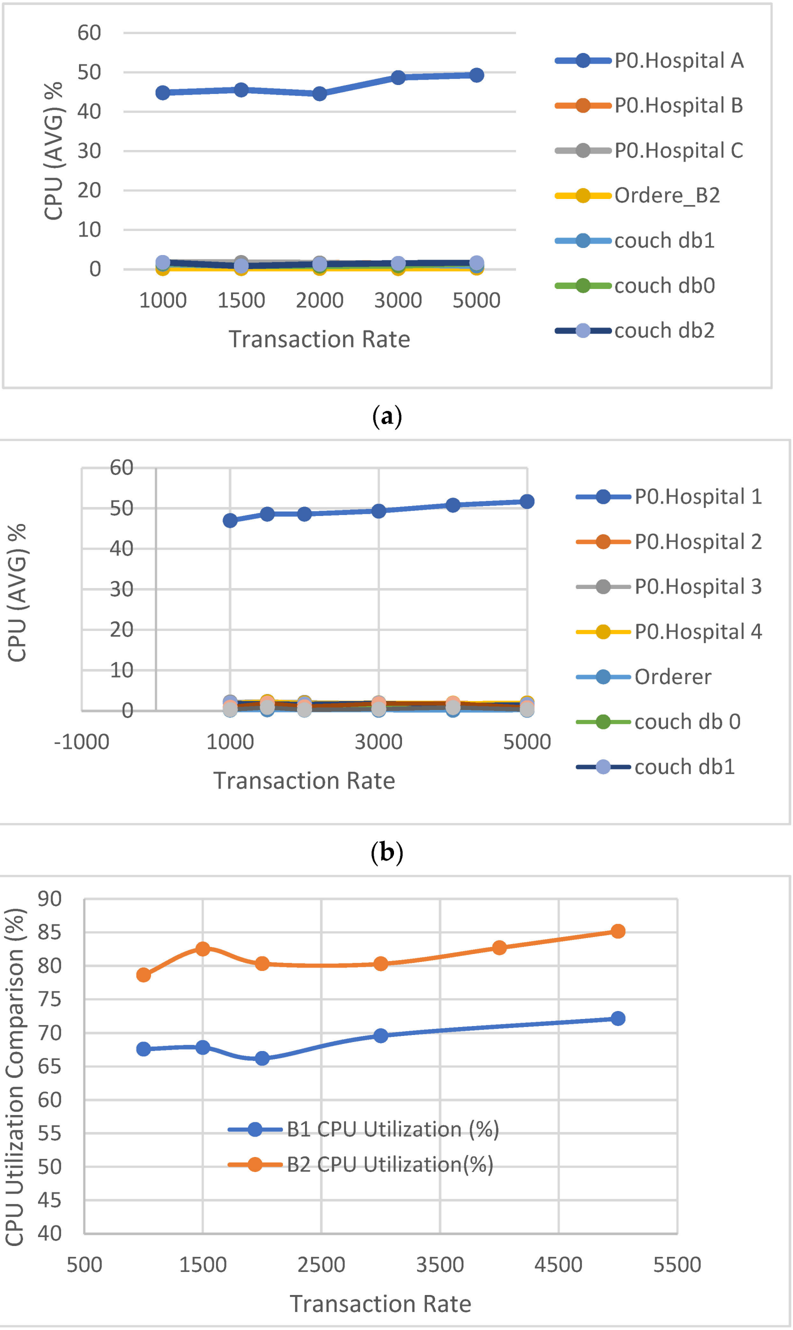
(**a**) (**b**)

(**c**) (**d**)

Figure 5. Scalability performance of test networks: (**a**) B1 and B2 latency vs. the number of transactions; (**b**) B1 and B2 throughput vs. the number of transactions; (**c**) throughput percentage vs. the number of transactions; (**d**) B1 and B2 average execution time.

5.3. CPU Utilization

Figure 6 illustrates the CPU utilization of B1 and B2 as obtained by the Hyperledger Caliper report. Notably, P0.Hospital A (Figure 6a) and P0.Hospital 1 (Figure 6b) show higher CPU utilization because these peers are in charge of running and verifying the smart contracts and the endorsement policy. Figure 6c illustrates the total CPU utilization for both test networks, B1 and B2. B1 consumes approximately 68% of the CPU, and it increases as the transaction load increases and reaches approximately 72% as the highest value. In comparison, running four healthcare entities, B2 consumes approximately 79% of the CPU, which is 10% more than B1 at a transaction load of 500. As the transaction load increases, B2 CPU consumption increases, and at a transaction load of 5000, it consumes approximately 85% of the CPU, resulting in an average of 339 failed transactions. Therefore, we conclude that increasing the number of healthcare entities will result in a higher CPU utilization. Moreover, upon reaching the saturation point, some of the transactions may fail to commit. Hence, the CPU power of any used system needs to be considered and appropriately configured for running a scalable healthcare blockchain network.



(**a**)

(**b**)

(**c**)

Figure 6. Test network CPU utilization: (**a**) blockchain B1 resource utilization, (**b**) blockchain B2 resource utilization, and (**c**) comparison of B1 and B2 CPU utilization.

5.4. Inter-Blockchain Communication Performance

In this study, we tested the interoperable operations between two independent Hyperledger Fabric blockchains B1 and B2, such that the client application running on B1 sends a request for EHRs of patients to blockchain B2. We used Hyperledger Cactus [55] for the integration of the developed blockchains. The workflow is depicted in Figure 7.

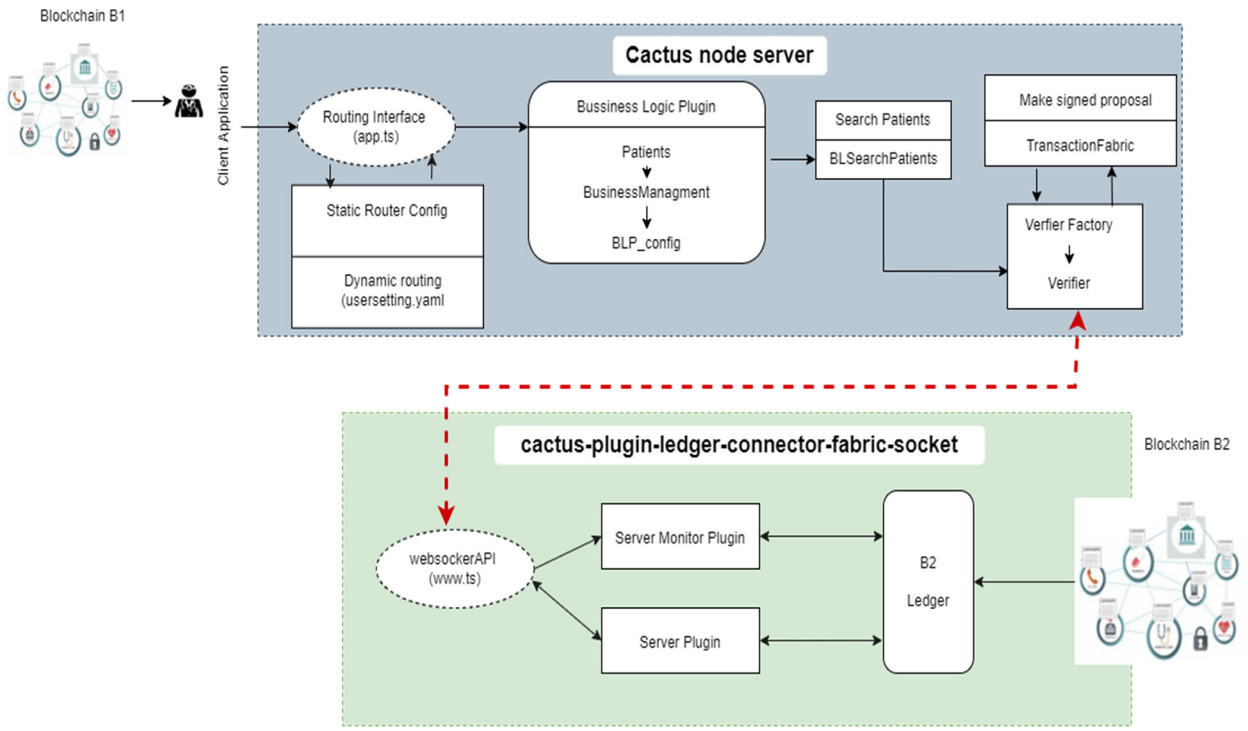


Figure 7. Blockchain B1 to B2 communication workflow using Hyperledger Cactus.

As shown in Figure 7, first, the client application requests the patient record via the Cactus node server and connects blockchain B1 to blockchain B2 for the retrieval of the patient's EHRs. The service bus of the Cactus node server transfers the query from B1 to B2. The B2 ledger is accessed for the patient's EHRs via the Cactus–plugin–ledger–connector–fabric–socket. From the B2 ledger, the hash of the EHR of patient (that resides in the decentralized off-chain storage) is added to the service bus. The service bus is responsible for transmitting the query from B1 to B2 and response from B2 to B1. The patient's record is sent back to the blockchain B1 client application via the service bus from the Cactus node server.

In this setup, we evaluated the average ET (Equation (1)) for client requests in blockchain B, as shown in Figure 8. Notably, the ET for the first query transaction is very high because of the initial connection to blockchain B2. After the connection is created, we see a gradual drop in ET for the second client request and onwards. Figure 9 shows a comparative analysis of blockchain B1 ET with blockchain B2 QT. B1 initiates a client request and waits for a response from the B2 blockchain. The B2 nodes process the query in time t and send the requested record to B1. Therefore, the ET of B1 depends on the QT of B2, as given in Equation (1). The entire experiment was performed in the same lab; therefore, the CT is very low, and the network delay is negligible in this test run.

Average Latency Comparison

In this experiment, we compared the latency of our proposed approach for transferring EHRs between blockchains with a previous solution reported in [56]. The paper used a trusted execution environment for asset transfers among blockchains in a supply chain domain. We compared our results with the work of the supply chain management domain, as the results of inter-blockchain communication implementation are extremely limited to date. Particularly, our work constitutes the first example of implementation in the healthcare domain, to the best of our knowledge. Therefore, we looked at the available literature results from other domains for comparison. Figure 10 shows the results of a comparative analysis of both methods. The results show that our proposed transaction-based inter-blockchain communication technique has significantly minimized latency for inter-blockchain transfer. According to the results in Figure 10, the previous solution [56] takes approximately 3.05 s to perform a transaction between two blockchains. However, in our case, the latency of inter-blockchain transactions was 1.0125 s. Here, the latency depends on the performance of the connected blockchains and the connection methodology used. Our proposed technique provides better results in the healthcare domain, where delays in record access cannot be tolerated, particularly in the case of an emergency.

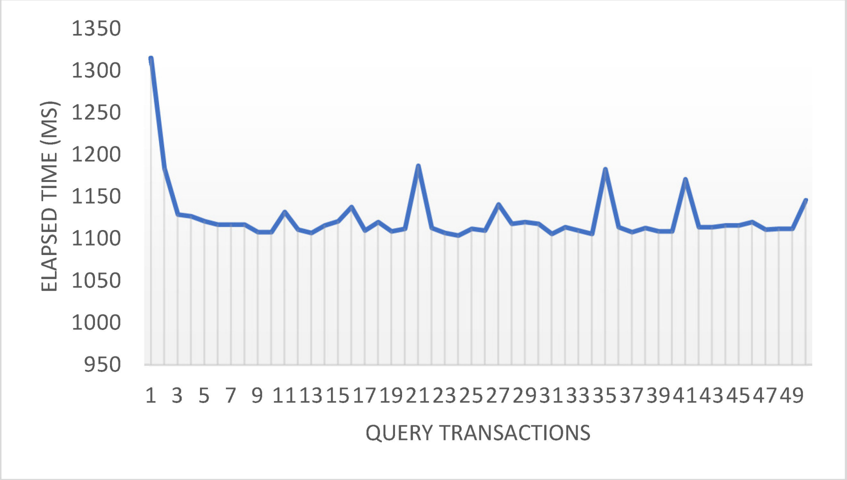


Figure 8. B1 elapsed time for query transactions.

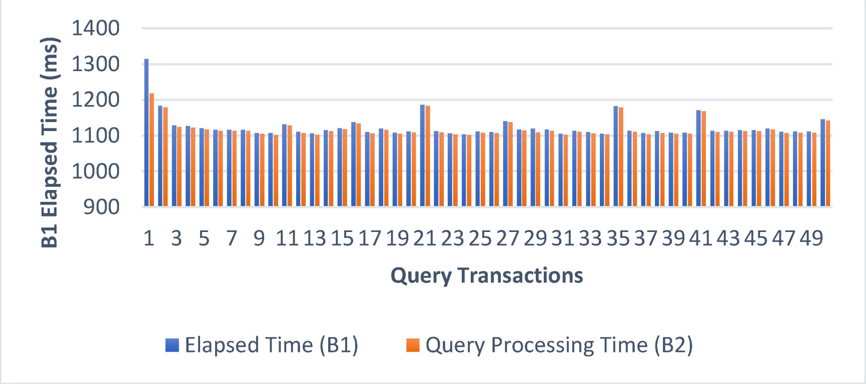


Figure 9. B1 elapsed time vs. B2 query processing.

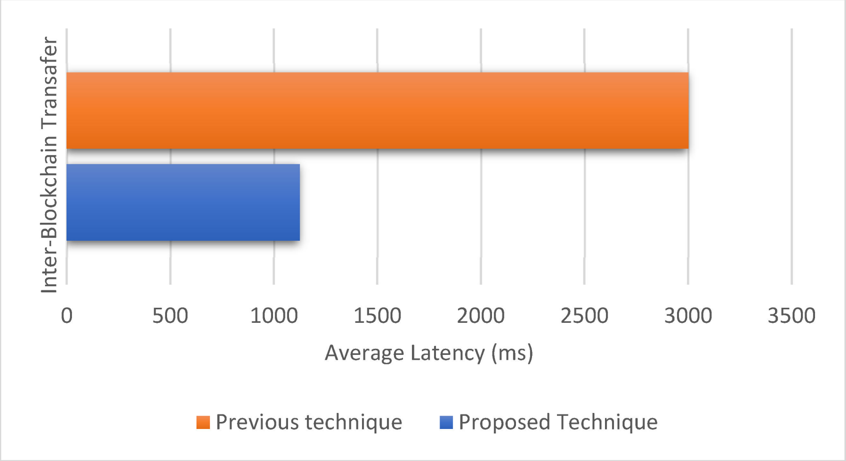


Figure 10. Comparison of average latency in inter-blockchain communication.

6. Conclusions

Blockchain technology has been extensively adopted in many applications, with its use in sharing patient EHRs being a prominent example in the case of healthcare. With such high demand and adoption of blockchains in a healthcare federation, the research community has recently investigated interoperability among the independent blockchains and proposed various kinds of solutions. However, these solutions are unable to fully resolve blockchain integration. In this study, we present a smart contract-based blockchain integration solution in a federation of independent blockchain networks for application in healthcare. We used local and global smart contracts for intra-blockchain and inter-blockchain communication, respectively. The results show that our proposed technique has improved performance compared with the recent work reported in [56]. We significantly minimized the average latency of inter-blockchain transfer and provided an efficient mechanism of blockchain integration for EHR sharing. However, further research is still needed to achieve optimal security and scalability performance in the blockchain communication process.

In our future work, we aim to develop heterogeneous blockchain integration solutions for EHR sharing in a healthcare federation with an enhanced security level and compare them with available relevant developed solutions within the same domain. Since inter-blockchain communication research is continuously progressing, we hope to find relevant experimental results for comparison in the near future.

Author Contributions: This work was designed and implemented by F.H. under the supervision and guidance of K.S. and F.S, who reviewed and corrected the technical aspects, algorithms, and the original writeup. As the principal investigator, K.S. provided funding. F.H. wrote the original draft of the paper with the help of K.S. and F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by Zayed Center for Health Sciences, the United Arab Emirates University, Grant number 31R180.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Jamoom, E.W.; Yang, N.; Hin, E. Adoption of Certified Electronic Health Record Systems and Electronic Information Sharing in Physician Offices: United States, 2013 and 2014. *NCHS Data Brief* **2016**, *236*, 1–8.
2. Uddin, M.; Memon, M.S.; Memon, I.; Ali, I.; Memon, J.; Abdelhaq, M.; Alsaqour, R. Hyperledger Fabric Blockchain: Secure and Efficient Solution for Electronic Health Records. *Comput. Mater. Contin.* **2021**, *68*, 2377–2397. [CrossRef]
3. Rouhani, S. MediChain TM: A Secure Decentralized Medical Data Asset Management System. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; No. Section II. pp. 1533–1538.
4. Tanwar, S.; Parekh, K.; Evans, R. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *J. Inf. Secur. Appl.* **2019**, *50*, 102407. [CrossRef]
5. Al-Karaki, J.N.; Gawanmeh, A.; Ayache, M.; Mashaleh, A. DASS-CARE: A decentralized, accessible, scalable, and secure healthcare framework using blockchain. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 330–335. [CrossRef]
6. McGhin, T.; Choo, K.-K.R.; Liu, C.Z.; He, D. Blockchain in healthcare applications: Research challenges and opportunities. *J. Netw. Comput. Appl.* **2019**, *135*, 62–75. [CrossRef]
7. Shuaib, K.; Abdella, J.; Sallabi, F.; Serhani, M.A. Secure decentralized electronic health records sharing system based on blockchains. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 5045–5058. [CrossRef]
8. Fan, K.; Wang, S.; Ren, Y.; Li, H.; Yang, Y. MedBlock: Efficient and Secure Medical Data Sharing Via Blockchain. *J. Med. Syst.* **2018**, *42*, 136. [CrossRef]
9. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. MedRec: Using Blockchain for Medical Data Access and Permission Management. In Proceedings of the 2016 2nd International Conference on Open and Big Data (OBD), Vienna, Austria, 22–24 August 2016; pp. 25–30. [CrossRef]

10. Li, H.; Zhu, L.; Shen, M.; Gao, F.; Tao, X.; Liu, S. Blockchain-Based Data Preservation System for Medical Data. *J. Med. Syst.* **2018**, *42*, 141. [CrossRef]
11. Zghaibeh, M.; Farooq, U.; Hassan, N.U.; Baig, I. SHealth: A Blockchain-Based Health System With Smart Contracts Capabilities. *IEEE Access* **2020**, *8*, 70030–70043. [CrossRef]
12. Hashim, F.; Shuaib, K.; Sallabi, F. MedShard: Electronic Health Record Sharing Using Blockchain Sharding. *Sustainability* **2021**, *13*, 5889. [CrossRef]
13. Milojkovic, M. Privacy-Preserving Framework for Access Control and Interoperability of Electronic Health Records Using Blockchain Technology. Showcase of Undergraduate Research and Creative Endeavors, April 2018, United States, [Online]. Available online: https://digitalcommons.winthrop.edu/source/SOURCE_2018/posterpresentations/64 (accessed on 15 June 2022).
14. Fatokun, T.; Nag, A.; Sharma, S. Towards a Blockchain Assisted Patient Owned System for Electronic Health Records. *Electronics* **2021**, *10*, 580. [CrossRef]
15. Fallis, A. Rootstock Platform: Bitcoin Powered Smart Contracts—White Paper. *J. Chem. Inf. Model.* **2015**, *53*, 1689–1699.
16. Back, A.; Corallo, M.; Dashjr, L.; Friedenbach, M.; Maxwell, G.; Miller, A.; Poelstra, A.; Timón, J.; Wuille, P. Enabling Blockchain Innovations with Pegged Sidechains. 2014. Volume 72, pp. 201–224. Available online: http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains (accessed on 21 May 2022).
17. Poon, J.; Buterin, V. Plasma: Scalable Autonomous Smart Contracts. White Paper, 2017, pp. 1–47. [Online]. Available online: https://plasma.io/ (accessed on 15 June 2022).
18. Wang, H.; Cen, Y.; Li, X. Blockchain router: A cross-chain communication protocol. In Proceedings of the 6th International Conference on Informatics, Environment, Energy and Applications, Jeju Island, Korea, 29–31 March 2017; pp. 94–97. [CrossRef]
19. Chen, Z.D.; Zhuo, Y.; Duan, Z.B.; Kai, H. Inter-Blockchain Communication. *DEStech Trans. Comput. Sci. Eng.* **2017**, 448–454. [CrossRef]
20. Kan, L.; Wei, Y.; Muhammad, A.H.; Siyuan, W.; Linchao, G.; Kai, H. A Multiple Blockchains Architecture on Inter-Blockchain Communication. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 16–20 July 2018; pp. 139–145. [CrossRef]
21. Fynn, E.; Bessani, A.; Pedone, F. Smart Contracts on the Move. In Proceedings of the 50th Annual IEEE/IFIP International Conference on Dependable Syst Networks, Valencia, Spain, 1 June 2020; pp. 233–244.
22. Dagher, G.G.; Adhikari, C.L.; Enderson, T. Towards Secure Interoperability between Heterogeneous Blockchains using Smart Contracts. In Proceedings of the Future Technologies Conference (FTC), Vancouver, BC, Canada, 29–30 November 2017; pp. 73–81.
23. Bennink, P.; Gijtenbeek, L.V.; Deventer, O.V.; Everts, M. An Analysis of Atomic Swaps on and between Ethereum Blockchains Using Smart Contracts; Technical Report; 11 Feb 2018. Available online: https://rp.os3.nl/2017-2018/p42/report.pdf (accessed on 19 April 2022).
24. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System, Decentralized Bussines Review 2009, p. 21260. [Online]. Available online: www.bitcoin.org (accessed on 6 July 2022).
25. Sumathi, M.; Sangeetha, S. Blockchain Based Sensitive Attribute Storage and Access Monitoring in Banking System. *Int. J. Cloud Appl. Comput.* **2020**, *10*, 77–92. [CrossRef]
26. Osmani, M.; El-Haddadeh, R.; Hindi, N.; Janssen, M.; Weerakkody, V. Blockchain for next generation services in banking and finance: Cost, benefit, risk and opportunity analysis. *J. Enterp. Inf. Manag* **2021**, *34*, 884–899. [CrossRef]
27. Ravi, D.; Ramachandran, S.; Vignesh, R.; Falmari, V.R.; Brindha, M. Privacy preserving transparent supply chain management through Hyperledger Fabric. *Blockchain Res. Appl.* **2022**, *3*, 100072. [CrossRef]
28. Queiroz, M.M.; Telles, R.; Bonilla, S.H. Blockchain and supply chain management integration: A systematic review of the literature. *Supply Chain Manag. Int. J.* **2019**, *25*, 241–254. [CrossRef]
29. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef]
30. Yetis, R.; Sahingoz, O.K. Blockchain based secure communication for IoT devices in smart cities. In Proceedings of the 7th International Istanbul Smart Grids and Cities Cong. and Fair (ICSG), Istanbul, Turkey, 25–26 April 2019; Available online: https://ieeexplore.ieee.org/abstract/document/8782285/ (accessed on 6 July 2022).
31. Hashim, F.; Shuaib, K.; Sallabi, F. Performance Evaluation of Blockchain Consensus Algorithms for Electronic Health Record Sharing. In Proceedings of the 2021 Global Congress on Electrical Engineering (GC-ElecEng), Valencia, Spain, 10–12 December 2021; pp. 136–143. [CrossRef]
32. Hölbl, M.; Kompara, M.; Kamišalić, A.; Nemec Zlatolas, L. A Systematic Review of the Use of Blockchain in Healthcare. *Symmetry* **2018**, *10*, 470. [CrossRef]
33. Hasselgren, A.; Kralevska, K.; Gligoroski, D.; Pedersen, S.A.; Faxvaag, A. Blockchain in healthcare and health sciences—A scoping review. *Int. J. Med. Inform.* **2019**, *134*, 104040. [CrossRef]
34. Mohan, A.P.; Gladston, A. Merkle Tree and Blockchain-Based Cloud Data Auditing. *Int. J. Cloud Appl. Comput.* **2020**, *10*, 54–66. [CrossRef]
35. Gai, K.; Guo, J.; Zhu, L.; Yu, S. Blockchain Meets Cloud Computing: A Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2009–2030. [CrossRef]

36. Qasse, I.A.; Talib, M.A.; Nasir, Q. Inter blockchain communication: A survey. In Proceedings of the ArabiWIC 6th Annual International Conference Research Track, Rabat, Morocco, 7–9 March 2019. [CrossRef]
37. Singh, A.; Click, K.; Parizi, R.M.; Zhang, Q.; Dehghantanha, A.; Choo, K.-K.R. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *J. Netw. Comput. Appl.* **2019**, *149*, 102471. [CrossRef]
38. Li, W.; Sforzin, A.; Fedorov, S.; Karame, G.O. Towards Scalable and Private Industrial Blockchains. In Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, Abu Dhabi, United Arab Emirates, 2–6 April 2017; ACM: New York, NY, USA, 2017; pp. 9–14. [CrossRef]
39. Deng, L.; Chen, H.; Zeng, J.; Zhang, L.J. Research on cross-chain technology based on sidechain and hash-locking. *Lect. Notes Comput. Sci.* **2018**, *10973*, 144–151. [CrossRef]
40. Donawa, A.; Orukari, I.; Baker, C.E. Scaling Blockchains to Support Electronic Health Records for Hospital Systems. In Proceedings of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 10–12 October 2019; pp. 0550–0556. [CrossRef]
41. Belchior, R.; Vasconcelos, A.; Guerreiro, S.; Correia, M. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. *ACM Comput. Surv.* **2021**, *54*, 168. [CrossRef]
42. Ding, D.; Duan, T.; Jia, L.; Li, K.; Li, Z.; Sun, Y. InterChain: A Framework to Support Blockchain Interoperability. Second Asia Pacific Work. Netw. 2018, [Online]. Available online: https://icowhitepapers.co/wp-content/uploads/ (accessed on 15 June 2022).
43. Borkowski, M.; Sigwart, M.; Frauenthaler, P.; Hukkinen, T.; Schulte, S. Dextt: Deterministic Cross-Blockchain Token Transfers. *IEEE Access* **2019**, *7*, 111030–111042. [CrossRef]
44. Polk, W.T.; Hastings, N.E.; Polk, W.T. Bridge Certification Authorities: Connecting B2B Public Key Infrastructures. In PKI Forum Meeting Proceedings; 2000; pp. 27–79. Available online: https://csrc.nist.rip/groups/ST/crypto_apps_infra/documents/B2B-article.pdf (accessed on 25 April 2022).
45. De Angelis, S.; Aniello, L.; Baldoni, R.; Lombardi, F.; Margheri, A.; Sassone, V. PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain. In Proceedings of the 2018 Italian Conference on Cyber Security, Milan, Italy, 6–9 February 2018.
46. Tseng, L. Recent results on fault-tolerant consensus in message-passing networks. *Lect. Notes Comput. Sci.* **2016**, *9988*, 92–108. [CrossRef]
47. Castro, M.; Liskov, B. Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.* **2002**, *20*, 398–461. [CrossRef]
48. Al Asad, N.; Elahi, M.T.; al Hasan, A.; Yousuf, M.A. Permission-based blockchain with proof of authority for secured healthcare data sharing. In Proceedings of the 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), Dhaka, Bangladesh, 28–29 November 2020; pp. 35–40. [CrossRef]
49. A Blockchain Platform for the Enterprise—Hyperledger-Fabricdocs Main Documentation. Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/ (accessed on 15 January 2022).
50. Antwi, M.; Adnane, A.; Ahmad, F.; Hussain, R.; Rehman, M.H.U.; Kerrache, C.A. The case of HyperLedger Fabric as a blockchain solution for healthcare applications. *Blockchain Res. Appl.* **2021**, *2*, 100012. [CrossRef]
51. Pajooh, H.H.; Rashid, M.; Alam, F.; Demidenko, S. Hyperledger Fabric Blockchain for Securing the Edge Internet of Things. *Sensors* **2021**, *21*, 359. [CrossRef]
52. Figueroa-Lorenzo, S.; Benito, J.A.; Arrizabalaga, S. Modbus Access Control System Based on SSI over Hyperledger Fabric Blockchain. *Sensors* **2021**, *21*, 5438. [CrossRef]
53. Podda, A.S.; Pompianu, L. An overview of blockchain-based systems and smart contracts for digital coupons. In Proceedings of the 2020 IEEE/ACM 42nd International Conference on Software Engineering Work ICSEW, Seoul, Korea, 27 June–19 July 2020; Volume 20, pp. 770–778. [CrossRef]
54. Getting Started | Hyperledger Caliper. Available online: https://hyperledger.github.io/caliper/v0.4.2/getting-started/ (accessed on 20 February 2022).
55. Hyperledger Cactus: On the Road to General Blockchain Integration—Hyperledger Foundation. Available online: https://www.hyperledger.org/blog/2021/03/31/hyperledger-cactus-on-the-road-to-general-blockchain-integration (accessed on 15 March 2022).
56. Bellavista, P.; Esposito, C.; Foschini, L.; Giannelli, C.; Mazzocca, N.; Montanari, R. Interoperable Blockchains for Highly-Integrated Supply Chains in Collaborative Manufacturing. *Sensors* **2021**, *21*, 4955. [CrossRef]