The Knowledge Engineering Review, Vol. 35, e23, 1 of 17.
doi:10.1017/S0269888920000314

Cross-chain interoperability among blockchain-based systems using transactions

BABU PILLAI[1], KAMANASHIS BISWAS[1,2] and VALLIPURAM MUTHUKKUMARASAMY[1]

[1]*Griffith University, Gold Coast, Australia*
[2]*Australian Catholic University, Sydney, Australia*
e-mails: babu.pillai@griffithuni.edu.au, kamanashis.biswas@acu.edu.au, v.muthu@griffith.edu.au

Abstract

The blockchain is an emerging technology which has the potential to improve many information systems. In this regard, the applications and the platform they are built on must be able to connect and communicate with each other. However, the current blockchain platforms have several limitations, such as lack of interoperability among different systems. The existing platforms of blockchain applications operate only within their own networks. Even though the underlying technology is similar, it relies on centralized third-party mediators to exchange or retrieve information from other blockchain networks. The current third-party intermediaries establish trust and security by preserving a centralized ledger to track 'account balances' and vouch for a transaction's authenticity. The inability for independent blockchains to communicate with one another is an inherent problem in the decentralized systems. Lack of appropriate inter-blockchain communication puts a strain on the mainstream adoption of blockchain. It is evident that blockchain technology has the potential to become a suitable solution for some systems if it can scale and is able to cross communicate with other systems. For the multisystem blockchain concept to become a reality, a mechanism is required that would connect and communicate with multiple entities' blockchain systems in a distributed fashion (without any intermediary), while maintaining the property of trust and integrity built by individual blockchains. In this article, we propose a mechanism that provides cross-chain interoperability using transactions.

1 Introduction

Blockchain technology has revealed another dimension of the power of decentralization (Käll, 2018). In future, many information infrastructures will be built on decentralized networks, and blockchain technology will play a vital role in this area (Bahri & Girdzijauskas, 2019; Zheng *et al.*, 2017). In principle, blockchains may be considered as new global systems that operate like the Internet (Tasca & Tessone, 2017). However, instead of transmitting packets of information, blockchains move values or digital assets. It should be noted that their interaction is currently limited within the network they are operating on. Moreover, the models of blockchain systems are fragmented with progress being achieved in silos, whereas today's digital economy demands multiple systems to communicate with each other. However, there is no unified standard in blockchain designs yet, and this leads to the need for research regarding interoperability between chains (Buterin, 2016; Hardjono et al., 2018). The main purpose of cross communication among blockchain systems is to enable exchange or to retrieve information between different networks. Although interoperability of records between systems is a must, the current architecture of blockchain including other limitations such as scalability and latency (Zheng et al., 2017) does not support cross communication between two systems (Tasca & Tessone, 2017).

Introducing interoperability without violating the underlying structure and assumptions is a key challenge. It is expected that cross communication between systems should not alter any fundamental

behavior of the blockchain system. Therefore, the proposed mechanism is required to protect the unique characteristics of each blockchain by effectively managing the 'state change' (data append and validation) process. Moreover, the cross-communication process must follow the protocol employed by the corresponding blockchain system such as a consensus mechanism. To address these issues, this article proposes a cross-communication model that works on the application layer.

In the context of this article, we assume that the blockchain system operates on a consensus mechanism with N number of nodes. At any given time t, a subset of nodes $B(t) \subset N$ are Byzantine fault and their mining power $m(b)$ is less than 50% of the total compute power.

$$\forall t : \sum_{b \in B(t)} m(b) < 50\% \sum_{n \in N} m(n)$$

This article proposes a simplified solution to address cross communication between blockchain-based systems without an intermediary. Being user driven and transaction based, this process ensures the authenticity of the information generated and will not alter the heterogeneous nature of the blockchain system.

We are looking into a future where, in a network, many types of blockchains will be operated by different enterprises (both public and private blockchains) that need to interact with each other. It is evident that blockchain will change the way we interact with governments, banks, institutions, and each other (Zheng *et al.*, 2016). Many services including government will run blockchain systems solely to have an advantage of the decentralized immutable database, but not necessarily run and hold the data by some arbiter nodes instead, in the custody of multiple nodes within the organization. So having multiple blockchains that can interoperate may possibly unlock the full potential of the blockchain technology. In future, we may be able to communicate to a blockchain system just like how we send an e-mail between different networks.

The rest of the article is organized as follows. Section 2 describes the related works on blockchain interoperability. Section 3 provides a brief description about interoperability. Section 4 presents the proposed cross-communication model for blockchain-based systems. Section 5 analyzes the proposed model and finally Section 6 concludes the article.

2 Related works

Both industry and academia are actively performing research on blockchain architecture and protocols that would allow blockchains to cross communicate between different networks and facilitate the exchange of transactions. Especially, many start-up companies are working to build an interoperable blockchain platform to enable instant transfer and recording of data and value between two systems.

Li *et al.* (2017) have proposed a model of multi-blockchain architecture forming interconnected blockchain that is fit for industrial requirements. The model consists of a number of satellite chains which are individual blockchains having their own consensus protocols and access mechanisms connected together to form a network of blockchains. The assets are transferred between satellite chains through a transaction process based on the policies. These chains cross communicate with each other through special nodes that facilitate the transactions for the connected satellite chains. Thus, the system addresses the scalability problem by interconnecting multiple blockchains where each blockchain is secured by its own consensus mechanism. Furthermore, the system achieves 'governance' by employing 'regulators' connected to each satellite chain to enforce policies that are deployed in the form of *smart contract* and *auditors* linked to the satellite chains to monitor the transaction activities.

Wang *et al.* (2017) have proposed an application known as a blockchain router to connect different blockchains through a router mechanism that is similar to the basic concepts of the Internet router. Independent blockchain systems are connected to a 'sub-chain', and the sub-chain holds a copy of the connected blockchain data. These sub-chains then connect to the 'router' through a connector. The 'connector' is the link between sub-chain and the router that operates as a blockchain using the practical Byzantine fault tolerance algorithm and powered by a token called 'zac'.

A HyperServices (Liu et al., 2019) project proposes interoperability platform for building and executing decentralized applications (dApps) across heterogeneous network of blockchains. The platform consists of many dApps interacting with a verifiable execution systems (VESes) that process and execute the request from dApp to the corresponding network of blockchain. Both the dApps and VESec operate on a cryptography protocol to securely execute the transactions across different blockchains. Here the VESes operate as the mother blockchain. Similarly, Greenspan (2015) proposed an off-the-shelf configurable platform called multichain but can only work with homogeneous network.

Kan et al. (2018) proposed a multiple blockchain architectures that can transact across a heterogeneous network. In their model, cross-chain communication happens through an inter-blockchain connecter of a routing management system. The router system maintains routing information of the involved blockchain system. A network must choose a router node to communicate with other networks. These router nodes establish the network and obtain neighbors network information. Once the router information is updated, all router nodes consent the newest routing table. In this way, the router blockchain system records the validated address of each participating blockchain. When a transaction between chain N_1 and chain N_2 is generated, chain N_1 can establish a connection with chain N_2, transferring the data according to the routing information written in router blockchain. The article also introduced a unified packet for the transaction and routing.

Weber *et al.* (2019) propose platform architecture of a multi-tenant systems. Each tenant will have its own private network and all those networks are anchored into a main public blockchain. The system achieves data integrity by anchoring the tenant chains data to the public chain. Such an architecture is more useful for a long-living and a short-lived blockchain for long- and short-running business needs or a separate blockchain per year.

Ding *et al.* propose 'InterChain' a blockchain framework that supports interoperability between blockchain networks. The architecture consists of a number of networks called sub-chains (the chain that need to be connected) and a mother blockchain called InterChain that connects all sub-chains together. The InterChain consists of validators (that participate in the connected interchain) and gateway nodes (that participate in both the chain). The gateway node relays the cross-chain transactions to the InterChain, thus the InterChain validate cross-chain transactions. This proposal is lacking details on the consensus algorithm of that InterChain.

The Cosmos[1] project aims at creating an Internet of blockchain that connects different chains through the inter-blockchain communication protocol. Interoperability is achieved through the shared Cosmos Hub powered by the Tendermint (Kwon, 2014) consensus protocol which keeps track of the number of tokens in each connected chain and manages transfers between them. Similar to the Cosmos project, the Polkadot project aims to address cross-chain communication and transfer of values. In addition, Polkadot aims to address the transfer of data between blockchains. In a Polkadot[2] network, the main blockchain is called a 'relay chain', and the connected chains are called 'parachain'. In this network, all the parachains have to adopt a pool consensus operated by the main relay chain. This allows each parachain and the relay chain to utilize the entire network's validators to secure the overall network. If a parachain is compatible with Polkadot, it can connect and leverage the security of Polkadot's consensus mechanism. Otherwise, they must use a bridge to connect to the Polkadot network.

Overledger[3] is another proposal for a multi-blockchain application. Rather than making another blockchain to support interconnection among many blockchains, a blockchain operating system is introduced to operate with other connected ledgers. The overledger is based on the concept of Multi-Chain Applications. They propose an application layer information exchange protocol, where the application can communicate, migrate, and exchange information and value regardless of the ledger on which they have been deployed.

The current research on interoperable chains, where value can be transferred from one chain to another, is only a concept and not yet tested in practice. In addition, these proposed models need to either adopt

[1] https://cosmos.network/ (last accessed 12 March 2020)
[2] https://polkadot.network (last accessed 12 March 2020)
[3] https://quality.coinpaper.io/files/whitepapers/qnt-quant_whitepaper.pdf (last accessed 12 March 2020)

a private blockchain or customize the design to interconnect through third-party bridges. In order to evaluate, these systems should be put into practice. However, there is no such system in the space of private blockchain that has been adopted on a vast scale in order to test these concepts in practice.

3 Interoperability

Interoperability refers to the ability of two or more systems to provide or accept service from the other system and to utilize the service of a common exchange effectively together (Vernadat, 2006). The linkage should allow these connected systems to exchange data accurately, effectively, and consistently (Geraci *et al.*, 1991). The purpose of cross communication among blockchain systems is to enable 'exchange' or to 'retrieve' information or value between different networks. This deals with information obtained from another system and makes a change in the state of that system based on the received information. However, inherently the blockchain is an 'append-only' model, and the state can only be appended through transactions, by nodes within its own network using their consensus mechanism (Alqassem & Svetinovic, 2014; de Kruijff & Weigand, 2017; Tasca & Tessone, 2017; Xu *et al.*, 2017; Zheng *et al.*, 2017). Therefore, here the underlying assumption is that "cross-communication is not intended to make direct state changes to another blockchain system. Instead, a cross-communication should trigger some set of functionalities on the other system that is expected to perform an operation within its own network", as an example, verifying the authenticity of information requested within its own network.

Transactions are the functional property (Xu *et al.*, 2017) of blockchain-based system that serves as an input request to the network in an attempt to update the state (de Kruijff & Weigand, 2017). A transaction-based cross-communication process will ensure the authenticity of the information generated through the request. Therefore, a smart contract-based cross-communication transaction mechanism between different networks of blockchain systems will not alter its heterogeneous nature. Considering the decentralized architecture, where multiple nodes participate in the process to reach finality, nodes must retain the same result. For that, nodes must have or be given the information in order to process the transaction. However, if the nodes are set to fetch data from other blockchain systems, the dynamic nature of values would interfere with the consensus. Therefore, a user-driven process for cross communication using transactions is proposed.

4 The proposed cross-communication model

To provide cross communication for blockchain-based systems, we propose an application-level cross-communication model. The proposed cross-communication model has two stages. The first stage retrieves information from a blockchain system through a process called 'information query'. In this stage, the recipient can request blocks from a client and verify the blocks. However, in a distributed system, there is no guarantee that every node is reliable. Therefore, getting information through full consensus is required. In the second stage, the state of the system is updated by adding data to the blockchain and this process is called state changes. This is achieved through the transaction, verification, and validation process of the blockchain system.

In this article, the cross-communication transaction refers to the transaction that calls a smart contract specifically designed for executing the functional requirements (Xu *et al.*, 2017) of the cross-communication process. Our approach is to treat each blockchain as individual network and such network may deal with records of financial transactions or ownership of assets. These individual blockchains can be customized and designed to suit their own specific requirements, which include aspects such as network participants' access permission and consensus protocol. The cross-communication process between different blockchain is established from an application through the user's account.

4.1 Assumptions

Our work aims to provide a simple mechanism for cross-chain asset verification and transfer. The process should ensure transfers are performed in a decentralized and trustworthy manner. Assets can be

represented on blockchains in various ways. Apart from native currencies such as BTC on Bitcoin and ETH on Ethereum network, there are other types of crypto assets created on top of a blockchain that represent a wide range of assets beyond crypto currencies. In the recent past, various asset types with different properties have been discussed, such as crypto coins, asset tokens, and utility tokens. We refer to our previous work for a thorough analysis (Pillai *et al.*, 2019).

The following assumptions are made for the proposed model.

- ✓ Both the networks involved in the cross-communication process recognize and form a common understanding of the crypto assets they are holding.
- ✓ Each participating blockchain networks' security will entirely depend on their system's design.
- ✓ The users involved in the cross-communication process 'trust' each other to a required level and are willing to process the transaction if valid proof is presented by the other party.
- ✓ Correspond blockchain systems are being able to run a smart contracts.

4.2 Method of cross communication

In our model, the cross communication among blockchains is established through the user's account. First, a cross-communication transaction is triggered in the source blockchain; later, with the confirmed block from the source blockchain, a transaction is triggered in the destination blockchain in order to complete the cross-communication process. The consensus mechanisms of the corresponding blockchains verify each such transaction, so that the protocols can maintain the integrity and security aspects of the system. This consensus process includes the economic incentive model, the verification process, and access control for the participants on the network. Although these blockchains operate on different networks, the assumption is that they accept transfer from one to another by having a fully verified transaction in a block. Unlike other methods of inter-communication using an intermediary, this approach provides inbuilt authenticity for the exchanged information.

Figure 1 demonstrates an overall high-level view of the proposed model. The diagram represents a process of exchanging information from one blockchain system to another. The vertical lines represent the actors and the horizontal arrows represent communications with the network. The source blockchain system where the information is exchanged or retrieved is referred to as the 'sender' system, and the second blockchain system that updates its state based on received information is referred to as the 'recipient' system. The horizontal arrow represents the direction of communication between the entities.[4]

The cross-communication process begins with an information query transaction in sender blockchain initiated by the user. The transaction triggers a cross-communication function within the sender blockchain and reaches finality with full consensus of the nodes within that network. The user then communicates through a wallet software or application with the recipient blockchain's user and provides the confirmed block as a proof of the transaction. The next step is the state change transaction process, which takes place in the recipient blockchain. In this step, a cross-communication transaction will be triggered within the recipient blockchain by its user on the basis of the information provided by the sender. Once this transaction goes through and reaches finality, the cross-communication process is marked as completed.

In our design, the user initiates both the cross-communication process and information exchange as part of cross-communication process. We considered a user as a light client, running a smart app or wallet software capable of sending the transaction to the blockchain system to which they are connected. We presume a wallet software or smart app can be developed to create these cross-communication processes using web3.js to communicate with the Ethereum network, similar to a MetaMask[5] extension. We have chosen the Ethereum platform for our experiments since it is currently the most widely used smart contract platform.

[4] https://www.itu.int/rec/T-REC-Z.120 (last accessed 12 March 2020)
[5] https://metamask.io/ (last accessed 12 March 2020)

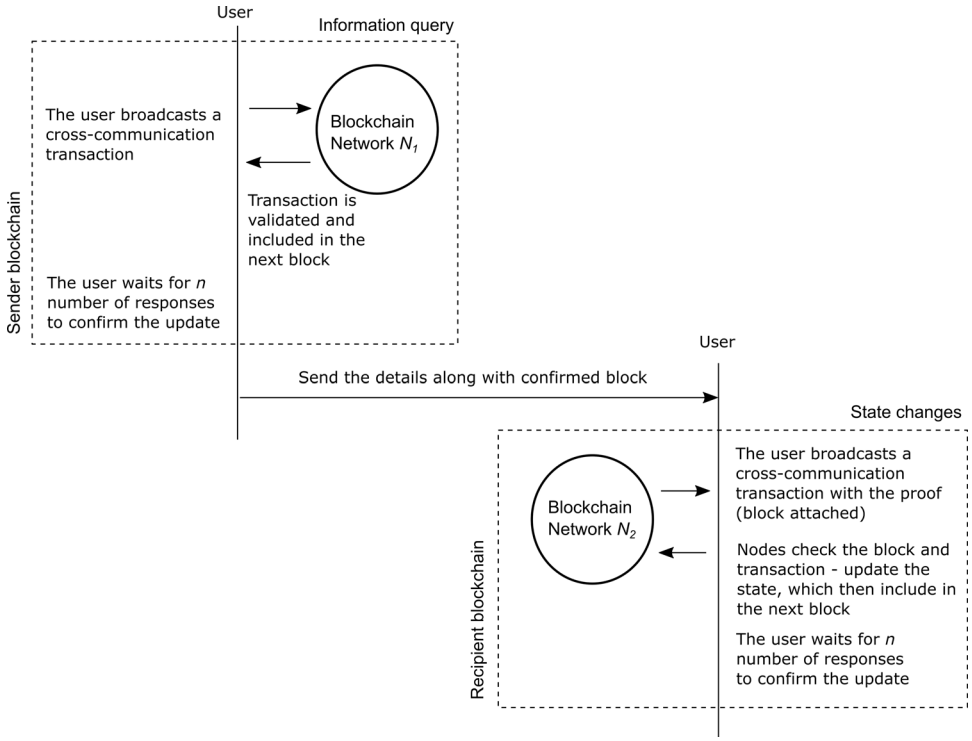

Figure 1 Overview of the proposed model.

4.3 Information query—transaction

The operation of the proposed cross-communication model begins with an information query when retrieving information from a blockchain system. For example, a blockchain system holding asset records facilitates users from another blockchain system that is able to query the ownership (details) of the assets. In the information query phase, to facilitate the information query, the sender blockchain system should deploy a cross-communication smart contract that is capable of verifying the asset records belonging to that blockchain. The information query process is performed on the sender blockchain in three steps:

a) **Set up the query:** The user will create a transaction T_x requesting information I and broadcast it to the network N_1. This transaction query M is addressed to the cross-communication smart contract deployed on the corresponding blockchain.

$$M = T_x\,(I)$$

b) **Verify the query:** Any receiving node belonging to the corresponding blockchain treats this transaction as standard transaction; then the node processes and propagates to other nodes within the network to include this transaction in the next block. Thereby the transaction T_x will be verified by n number of nodes in network N_1 and included in the next block B_n along with other verified transactions.

$$B_n = \sum_{i=1}^{m} M_i$$

c) **Verify the result:** Once the transaction is verified and included in a block, the sender can verify the result and use it as proof of a valid transaction. The user receives the B_n as proof of T_x which has the information I.

$$R_H = H\,(H\,(M_1)\,, H\,(M_2)\,, \ldots..., H\,(M_m))$$

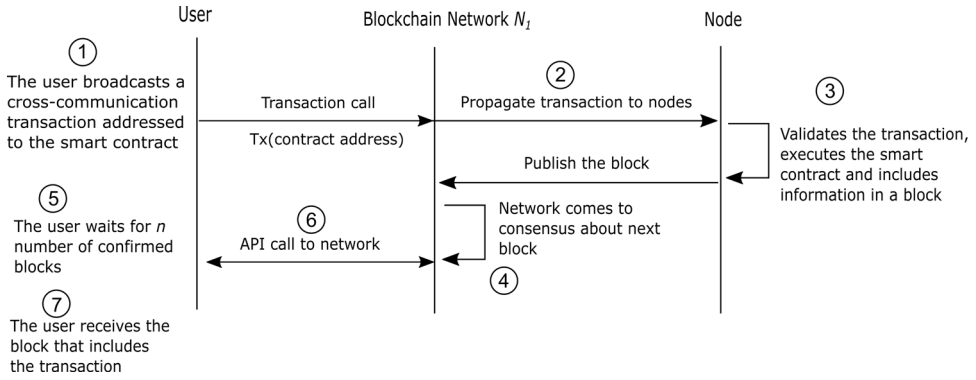

Figure 2 Information query—message sequence chart.

R_H denotes the root hash of the transaction obtained from the hash of all transaction hashes included in that block.

Figure 2 demonstrates a concept model of a sender blockchain system that records asset ownership. The vertical line represents the actors and the horizontal arrow with header represents the direction of communication between the entities. The numbers in circles indicate the time order of actions. This blockchain system has deployed the cross-communication smart contract and published the smart contract address to the relevant parties.

A brief high-level description of how the communication process interacts with the system is described below.

1. The user broadcasts a cross-communication transaction that is addressed to the smart contract deployed on the blockchain network N_1.
2. Nodes belonging to this blockchain treat this as a standard transaction that calls the smart contract. Each node then processes the transaction and propagates to other nodes in the network to be included in the next block.
3. Nodes verify the transaction by executing the smart contract, which carries out the asset verification process. After that, the transaction result will be included on the next block they form.
4. The network comes to a consensus and selects the next block on the chain.
5. The user waits for '*n*' number of confirmed blocks on top of the block.
6. The user's wallet will make an API[6] call to monitor the progress.
7. The wallet will notify the user once it reaches the desired block height.

The underlying assumption here is that this blockchain has a smart contract that is capable of running the information query transaction. This transaction does not make any state changes. Instead, it verifies the status of the state and records the result in the next block, so that the sender can use it as a proof. This is a simple verification service that can have several use cases. Since a confirmed block contains the state of the transactions hashed into its block header, for a given block, one can compute and prove that a transaction was included in that block's Merkle tree.

4.4 State change—transaction

The state change process is responsible for updating the state of the blockchain system based on the retrieved information. Referring to the above assumption, this blockchain system and its users accept a transaction if it is included in the block. The communication process is very similar to the information query model, except that the user who is invoking the state change process must belong to that blockchain and authorize the state change by signing the transaction.

[6] API refer to application programming interfaces that help the wallets to subscribe events on a blockchain network.

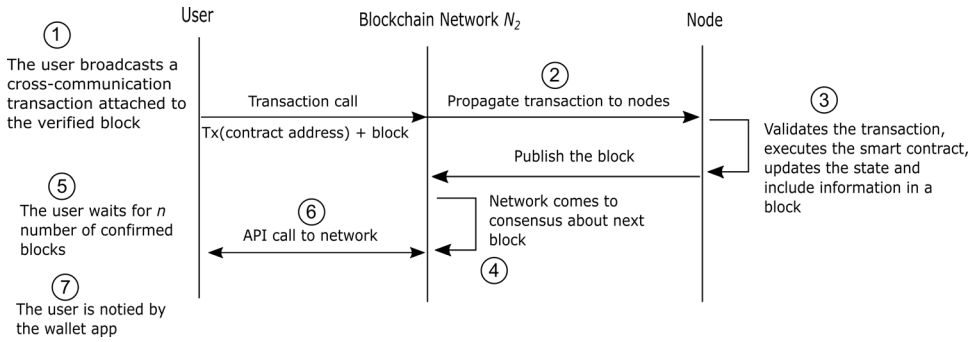

Figure 3 State change—message sequence chart.

The following activities take place in the recipient blockchain.

a. **Set up the state change request:** The user creates a transaction T_x attaching information I obtained through the information query transaction. The transaction is encrypted by the private key K_r of the user and broadcasts to the network N_2. The message generated in this phase is given by

$$M' = [T_x\ (I)]_{K_r}$$

b. **Verify the request:** The n number of nodes on the network N_2 verifies the received encrypted message with the public key K_p and validates the transaction. Once validated, the transaction T_x will be included in the next block B'_{n} along with other verified transactions as follows.

$$B'_n = \sum_{i=1}^{m} M'_i$$

c. **Verify the result:** The user gets the B'_{n} as proof of T_x which has information about the state change.

$$R'_H = H\left(H\left(M'_1\right), H\left(M'_2\right), \ldots\ldots, H\left(M'_m\right)\right)$$

Once the transaction is verified and added in a block, the state will be updated and the sender can verify the result and use it as proof of a valid transaction. Figure 3 demonstrates a concept model of a recipient blockchain system that processes the state change. The vertical line represents the actors and the horizontal arrow with header represents the direction of communication between the entities. The numbers in circles indicate the time order of actions.

The following is a high-level description of how the communication process interacts with the system:

1. The user broadcasts a cross-communication transaction addressing the smart contract deployed on the blockchain network N_2. Here the user must include the proof (block) of information query transaction received from the sender blockchain and the user has to sign the transaction with his private key.
2. Nodes process the transaction and propagate to other nodes (as with the information query).
3. Nodes will check the signature and validate the transaction, which executes the smart contract to carry out the asset verification process. The transaction result will be included on the next block of the node form. The nodes verifying the transaction will verify the transaction root hash of the given block, and they will not store the block itself; instead they only store the hash of the block header.
4. The network comes to a consensus about the next block (as with the information query).
5. The user can wait for n number of confirmed blocks on top of this block.
6. The user's wallet software will make an API call to monitor the progress.
7. The wallet software will notify the user once it reaches the desired block height.

The model assumes that the system will trust and accept a verified transaction in a block as proof, provided it meets the predefined requirements such as block height and majority consensus. Moreover, each

transaction log is recorded in the corresponding blockchain systems and are thereby verifiable by the users.

5 Testing and analysis

For testing and analysis purposes, we demonstrate a high-level conceptual model of a multi-blockchain cross-communication scenario. The goal is to measure the total processing time T for a cross-communication process carried out between two distinct blockchain systems. These are independent networks of blockchains, running their own consensus protocols. However, it is assumed that the value of information processing is the same and known by both the systems. The analysis is performed at a high-level abstraction: we consider interactions existing between the system and its environment and between its components without taking data security into consideration.

5.1 Application testing

Potentially, this technology can have applications in areas such as finance, economics, and digital assets management. In general, from an application perspective, one of the main performance measures corresponds to how fast a blockchain system performs for a given operation, for example, confirmation of a transaction. However, due to the distributed nature of the system, the performance measures resemble the collective nodes' response time. On top of such design constraints, there also exist other factors belonging to the distributed systems, especially the network propagation time which is dependent on the network topology and hardware configuration of the participating nodes. Therefore, evaluating the performance based on a single parameter is not ideal (Hileman & Rauchs, 2017).

In order to verify the feasibility of the proposed model, we have used an asset enquiry application and conducted the experiment. The application is part of the cross-communication process, where a user verifies the ownership of an asset registered on the blockchain and updates the state. For this process, an application layer comparison is used, such as the performance of the transaction process. A couple of performance metrics, 'transaction deployment latency' and 'block confirmation latency' are chosen to evaluate the model.

Currently, most of the tests performed to evaluate blockchain systems are on a simulated network that consists of a few user nodes, mining nodes, and one or more network topologies (Kan et al., 2018). However, the real network is not as steady as the simulated network. Therefore, we decided to experiment with three shared public Ethereum blockchain test networks that are configured to simulate blockchain systems, such as Ropsten[7], Rinkeby,[8] and Kovan[9]. The tests were conducted over the period of 2 days on each network, and the data were collected for the following activities: smart contract deployment—the time required to deploy the smart contract on the network; and asset verification and block confirmation process—the time it took to commit the asset verification transaction and include it in a block. Smart contracts were deployed using the Truffle framework through Infura[10], a gateway which provides a connection to a full node. Testing of asset verification and state change transaction was done through a JavaScript browser[11] interface with web3[12] API and Metamask[13], a chrome extension that interacts with the Ethereum network.

The test was performed as a user requesting a piece of information through a transaction about an asset to a targeted blockchain network N. Each transaction was sent in a synchronous manner, that is, one after the other one was confirmed. This is because the intention was only to measure the time latency t for the given transactions. Moreover, the network used was as close as possible to the real network; therefore,

[7] https://ropsten.etherscan.io. (last accessed 12 March 2020)
[8] https://www.rinkeby.io (last accessed 12 March 2020)
[9] https://authorities.kovan.network. (last accessed 12 March 2020)
[10] https://infura.io/ (last accessed 12 March 2020)
[11] https://github.com/b-pillai/KER2019.git (last accessed 12 March 2020)
[12] https://web3js.readthedocs.io/en/1.0/index.html (last accessed 12 March 2020)
[13] https://metamask.io/ (last accessed 12 March 2020)

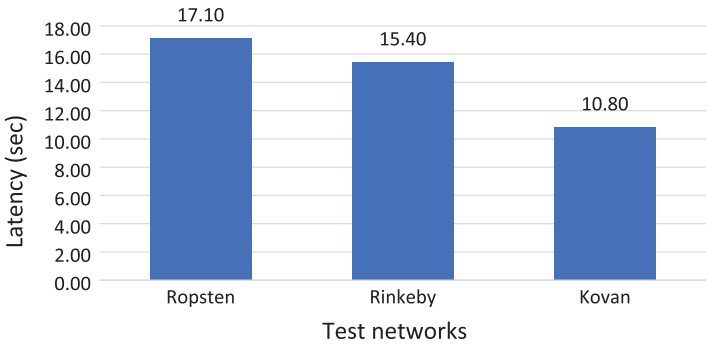

Figure 4 Smart contract deployment latency across Ethereum-based testnets.

the only requirement was to measure how this application performs in different networks. At the time of our test, these networks had 100 nodes participating in the verification process, using Proof of Work consensus for Ropsten network, Proof of Authority consensus for Rinkeby network, and Proof Authority consensus for Kovan network.

5.1.1 Smart contract deployment

Smart contracts are programmed logics, that are deployed on the blockchain, and have to be executed by every consensus node. As in the context of blockchain, smart contracts are agreements in the form of computer code stored on the blockchain (Sklaroff, 2017). These smart contracts not only define the rules, they also enforce those rules automatically. This enables the developers to write their own contracts in a logical code that includes contractual terms of each party and enforces to self-execute. Given results show the latency in deployment of smart contracts, while the other parameters such as gas limit and mining difficulty remain constant.

Figure 4 has been plotted from smart contract deployment latency across Ropsten, Rinkeby, and Kovan testnets. The tests were conducted using a Dell Inspiron 15 7000 series laptop with a 16-GB RAM running on an Intel i7 processor. The resultant mean latency is shown in seconds for different test networks. Ropsten network recorded the highest latency of 17.10 s, and Kovan recorded the lowest of 10.80 s. Even though the parameters used, such as gas variant and limit related to the smart contract, are the same, the latency variation may be due to the consensus model employed by the network and network propagation delay.

5.1.2 Transaction process

After deploying the smart contract, we interacted with it through a transaction or call function. The 'transaction' goes through the consensus process of mining and, if valid, will be included on the next block, whereas a 'call' is a local request of a contract function that does not broadcast or publish anything on the blockchain. Any interaction that makes a state change has to be a transaction. For our experiment, we have used transaction, because we wanted the result to be validated by the network. Given results show the latency of information query and state change transactions across Ropsten, Rinkeby, and Kovan testnets using the same system configuration as above. The other parameters such as gas limit and mining difficulty remain constant for each test.

Table 1 shows the mean and standard deviation measurements obtained from the test data. For the information query process, the results indicate a slightly higher standard deviation for Ropsten network than for other networks. This indicates that some information query transactions experience higher network propagation delay, which may have been due to network congestion. The overall average (across three networks) transaction deployment time of 13.26 s and block creation time of 18.88 s were obtained from the information query. For the state change process, the results indicate a somewhat consistent latency across the network. For this, the overall average transaction deployment time of 21.46 s and block creation time of 51.46 s were obtained.

Table 1 Mean (M) latency (s) and standard deviation (SD) of the transaction process

	Information query				State change			
	Transaction deployment		Block creation		Transaction deployment		Block creation	
Network	**M**	**SD**	**M**	**SD**	**M**	**SD**	**M**	**SD**
Ropsten	16.46	3.52	22.00	5.01	21.20	3.91	48.06	5.41
Rinkeby	12.53	3.04	19.60	4.59	20.66	4.04	50.26	5.83
Kovan	10.80	3.07	15.06	3.15	22.53	4.08	56.06	5.16

The state change process had a higher latency than the information query process. This was as expected, mainly because the state change process has a different smart contract which performs a state change operation. The state change process involves more computation; therefore, it takes more time and uses more gas. It was also noticed that when different networks were compared, these transactions had different latency. Even though these transactions referred to the same smart contract code and format, their response time was different based on the actual network. These results indicate the general nature of a blockchain-based system, where every transaction response time varies depending on the factors such as network latency, consensus protocol, gas limit, number of mining nodes, and transaction size.

From the above analysis, for this given situation, a mean block creation time was of 18.88 s for information query and 51.46 s for state change transaction were obtained. These results were used as typical values to estimate the cost of a cross-communication process. However, the cost assumption will vary according to the operational task of the smart contract, and the time constraint will depend on the network. Our aim was to study the cost of cross communication, with a specific configuration under the information query process. Full implementation and testing of the proposed model will be for our future work.

5.2 Theoretical analysis

We modeled two networks of systems, namely N_1 and N_2. The following variables were defined.

- ✓ t_1 is the transaction function call time.
- ✓ t_2 is the average transaction propagation time on a network.

The time t_2 is dependent on a number of factors and can be expressed as

$$t_2 = f_1\,(s, c, e, b)$$

where s corresponds to the contribution factor due to the execution of a smart contract function, c corresponds to constrains contribution factor due to the blockchain's consensus protocol, e corresponds to the economics incentive model, and b corresponds to the block size.

- ✓ t_3 is the transaction validation time.
- ✓ t_4 is the block confirmation time.

The block confirmation time is dependent on a number of factors and can be expressed as

$$t_4 = f_2\,(s, c, b)$$

- ✓ t_5 is the API callback time to get the confirmed results.
- ✓ T is the total time for the whole process to be completed and is therefore given by

$$T = t_1 + t_2 + t_3 + t_4 + t_5.$$

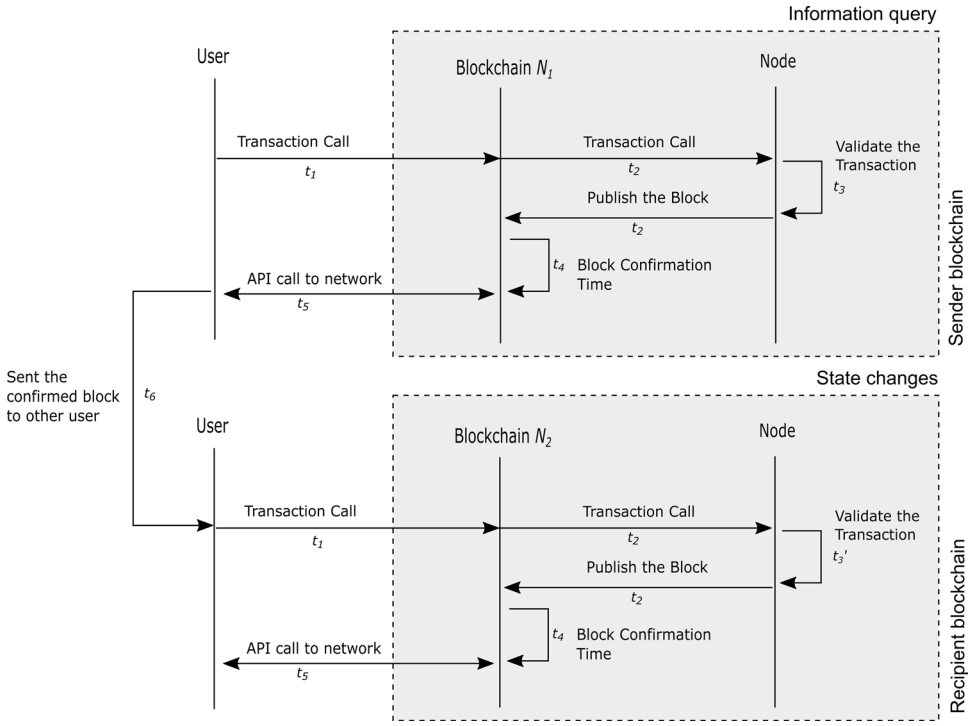

Figure 5 Message sequence chart of proposed model.

5.3 Activity sequence of the proposed model

Figure 5 represents an activity sequence diagram of our proposed model for a multi blockchain cross-communication scenario. The diagram represents the process of information exchange between two blockchain systems initiated through a user's account. The horizontal arrow with header represents the direction of communication and vertical line represents the entities. We considered the users to be running a light client such as a wallet software to initiate the transaction call.

The first half of the cross-communication process began with the user initiating a transaction function call to the sender's blockchain network, and we measured the time it took in each iteration of the process for the transaction call to reach finality and to be included in a block. As briefly stated above, t_1 is the time taken to broadcast a transaction function call; t_2 is the time taken to propagate the transaction call to the network, t_3 is the time it takes to verify the transaction by a miner, t_4 is the block confirmation time, and t_5 is the time to call the API and receive the results from the network. Once the network reaches finality, the user can send the confirmed block to the user belonging to the recipient's blockchain. Then the second half of the process of cross communication begins. At this stage, the user initiates a transaction function call to the recipient's blockchain network, attaching the confirmed block from the sender blockchain as a proof to process the request. Similar to the first half, t_1 is the time taken to broadcast a transaction function call, t_2 is the time a transaction call takes to propagate to the network, and t_3' is the time a transaction takes to be verified by a mining node. Here the timing will be different because both the processes run different smart contracts. The time complexity of processing a single transaction on a single node is a function of the code whose execution is triggered by the given transaction. t_4 is the time taken to confirm a block. t_5 is the time an API call to the network takes to get the results.

$T =$ information query time + information exchange time + state change time.

$$\begin{aligned} T &= (t_1 + t_2 + t_3 + t_2 + t_4 + t_5) + t_6 + (t_1 + t_2 + t_3' + t_2 + t_4 + t_5) \\ &= 2t_1 + 4t_2 + t_3 + t_3' + 2t_4 + 2t_5. \end{aligned}$$

In order to get an overall understanding, we use the average latency obtained from our testing, then the total time would be as shown in Table 2.

Table 2 Proposed model—application latency

Networks	Information query (N_1)	State change (N_2)	Total time $T(N_1 + N_2)$
Ropsten (N_1) to Rinkeby (N_2)	22.00	50.26	72.26
Rinkeby (N_1) to Kovan (N_2)	19.60	56.06	75.66
Kovan (N_1) to Ropsten (N_2)	15.06	48.06	63.12

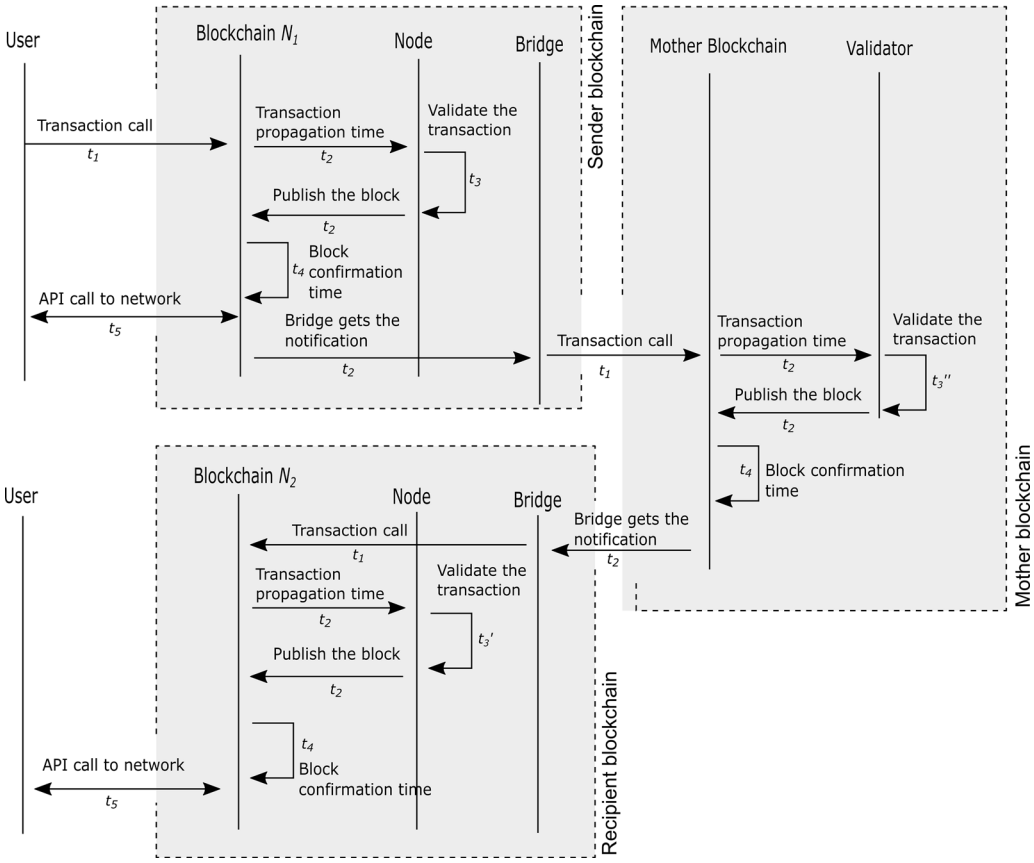

Figure 6 Message sequence chart of a mother blockchain model.

N_1 and N_2 denoted the corresponding networks, and the time T indicates an approximate time to complete the cross-communication process for the application. However, this time may be reduced or increased depending on the actual time it would take for each operation.

5.4 Activity sequence of the existing models

As proposed by Li et al. (2017), Wang et al. (2017), and other fintech start-up companies (Cosmos, Polkadot, Comit,[14] and Aion[15]), we developed a generic high-level conceptual model of the multi blockchain cross-communication model using a mother blockchain. Figure 6 represents the process of information exchange between two blockchain systems using this model. The horizontal arrow with header represents the direction of communication, and vertical line represents the entities. In this model,

[14] http://www.comit.network/ (last accessed 12 March 2020)
[15] https://aion.network/ (last accessed 12 March 2020)

Table 3 Mother blockchain model—application latency

Networks	Source blockchain (N_1)	Mother blockchain	Designation blockchain (N_2)	Total time $T(N_1 + N_2)$
Ropsten (N_1) to Rinkeby (N_2)	22.00	50.26	50.26	122.52
Rinkeby (N_1) to Kovan (N_2)	16.60	56.06	56.06	128.72
Kovan (N_1) to Ropsten (N_2)	15.06	48.06	48.06	111.18

the participating blockchains are connected to a mother blockchain through a bridge, and the mother blockchain validates the transfer process.

The cross-communication process begins with the user initiating a transaction function call to the sender's blockchain network. The bridge connected to the sender's blockchain network monitors the status of this transaction. As the transaction gets validated (and included in a block), the bridge initiates a transfer transaction request to the mother blockchain. The mother blockchain then validates the transfer request from the bridge and includes the result in the next block (here the assumption is that the mother blockchain also functions as a blockchain system). Similar to the sender blockchain network, the bridge connected to the recipient's blockchain monitors the transfer request transaction. Once that transaction gets approved in the mother blockchain, the bridge initiates a transaction to the recipient blockchain network, where it goes through, updates the state, and completes the cross-communication process.

Here we were using the same scenario of a cross-communication process and determined the timing for each process to complete. We aimed to use the same time factors and process as above, beginning with t_1 as cross-communication transaction broadcast time, t_2 as the time a transaction call takes to propagate to the network, t_3 as the time a transaction takes to be verified by a mining node, and t_4 as the time it takes to confirm a block.

A user from the sender blockchain processes a transaction function call, and it takes t_1 time, which is the time taken for the transaction to propagate through the network, t_3 is the time it takes for a transaction to be verified by a mining node, and t_4 is the time it takes to confirm a block. Here begins the second stage of the cross-communication process, based on the above transaction. The bridge node, connected to the sender and mother blockchain, creates and broadcasts a transfer transaction request to the mother blockchain. Here we assume the process of propagation, verification, and confirmation of the transactions are the same as t_1, t_2, t_3, and t_4. Once the transaction is verified and included in a block, the third stage begins. Based on this transaction, the bridge connection between the mother blockchain and the receiving blockchain creates and broadcasts a cross-communication transaction to the recipient blockchain network. Similar to the state change process, this transaction carries out a state change that makes the cross-communication processes complete. Here we assume the processes involved are t_1, t_2, t_3, and t_4.

T = (source blockchain + mother blockchain + designation blockchain)

$$\begin{aligned} T &= (\mathrm{t}_1 + 3\mathrm{t}_2 + \mathrm{t}_3 + \mathrm{t}_4) + (\mathrm{t}_1 + 3\mathrm{t}_2 + t_3'' + \mathrm{t}_4) + (\mathrm{t}_1 + 2\mathrm{t}_2 + t_3' + \mathrm{t}_4 + \mathrm{t}_5) \\ &= 3\mathrm{t}_2 + 8\mathrm{t}_2 + \mathrm{t}_3 + t_3' + t_3'' + 3\mathrm{t}_4 + \mathrm{t}_5 \end{aligned}$$

In order to keep things consistent, we applied the same timing used in our model, that is, the latency obtained from our test. We assumed the mother blockchain would take the same time as the state change transaction, because both processes involve state change. Therefore, the total time would be, as shown in Table 3. N_1 and N_2 denote the corresponding networks, and the time T indicates an approximate time to complete the cross-communication process.

Based on the above analysis, our proposed model takes an average of 70 s to complete the cross-communication process compared to the mother blockchain model, which takes an average of 120 s. Therefore, for this given condition, our model can perform more efficiently than other mother blockchain-based models. Further, our proposed methodology of cross communication for blockchain systems can be extensively used for scenarios such as asset ownership and identity verification. In our model, both

the participating blockchains need not be interconnected. Instead, the participating blockchain systems have to accept and execute a transaction that is capable of verifying the state of the asset (this will depend on what type of asset the corresponding blockchain is holding) and report response. The complexity and cost of the proposed system is expected to be reasonable. This is because our model proposes cross-communication function through a smart contract, which is easy to deploy on any blockchain system supporting smart contracts. Thus, the cross-communication process eliminates the need for modifying the blockchain client. Updating or modifying a native blockchain is quite expensive. Instead, forking an existing blockchain by writing a smart contract to support the design requirement, such as cross-communication function, is one of the best ways for creating simple blockchain applications.

5.5 Summary

The proposed model presents a generalized form of cross communication with a blockchain system that has the potential for extensive use in scenarios such as asset ownership and identity verification across multiple platforms. In the scenarios where the user can be an agency, they are able to verify asset ownership or identity details from a blockchain system. In this model, both the cross-communication processes are executed through transactions. Each such transaction is verified based on the corresponding blockchain system's consensus protocol. Thus, this model is not altering any fundamental operation of the blockchain system. The user initiates these cross-communication transactions and the exchange of information (block) between the blockchain systems. Thus, we are not introducing any intermediary.

First, we run the cross-communication transaction on the sender blockchain to obtain the information. Here the purpose of obtaining information through cross-communication transaction function is to protect the correctness of the information. Formal verification of correctness of the obtained information can be modeled, assuming that the system enforces its own consensus mechanism. To be clear, a thorough understanding of potential security threats is crucial. This is particularly important when dealing with a blockchain-based system, where the database is replicated, and any single node can be corrupted. We address this by enforcing a full consensus to obtain the information. Thus, unlike other methods of inter-communication using an intermediary, this approach provides the authenticity for the exchanged information. Then, we pass this fully confirmed block which includes the information to the recipient blockchain's users.

Second, we run the cross-communication transaction on the recipient blockchain system. This transaction includes an update request and carries the proof as to the transaction from the sender's blockchain. Here the user who creates and broadcasts the cross-communication transaction must authorize the state change by signing the transaction with his/her private key. This will allow only the account owner to operate state changes to the account. The rest of the verification and state change process follows in line with the blockchain's system.

As this is an early stage of the proposed model, it has some limitations. At this stage, we are not addressing any security concerns of the exchanged information. We assume the existing digital signatures or encryption mechanisms may address this issue. It is also noted that our proposed model employs cross-communication transactions using smart contracts. However, smart contracts are currently heavily researched; there may be vulnerabilities found in the code or even in the structure of the code itself (Delmolino *et al.*, 2016). Therefore, before deploying a contract, the code has to be thoroughly examined for any such issues.

6. Conclusion

We are looking into a future where, in a network, many types of blockchains will be operated by different enterprises (both public and private blockchains) that need to interact with each other. It is evident that blockchain technology will change the way we interact with governments, businesses, and institutions. Many organizations will run blockchain systems to harvest the advantage of the decentralized immutable database. Therefore, multiple blockchains that can interoperate would be able to unlock the full potential of the blockchain technology. In the future, it is likely that we should be able to exchange crypto assets

between blockchain system in a seamless manner, just as we now transfer an e-mail or message from one system to another.

We have theoretically analyzed the performance of the proposed model and briefly compared with other proposed systems that employ a mother blockchain as an intermediary. The goal is to measure and compare the total processing time T for a cross-communication process carried out between two distinct blockchain systems. In our proposed model, two major steps of operations have to be performed to complete the cross-communication process; first, on the sender blockchain, and second on the recipient blockchain. In contrast, in the mother blockchain model, for a cross-communication process to be completed, it has to go through three major steps of operations; first, on the sender blockchain, second on the mother blockchain, and finally on the recipient blockchain. Our proposed model is relatively simple to implement in any blockchain system that supports smart contract. The experimental results show that our model achieves relatively better performance than the mother blockchain-based models.

The future works aim to implement the proposed model in a single application environment connecting two different networks of blockchains. Since blockchain technology, more broadly distributed ledger technology, is continually evolving, both in the private and public sectors, we believe this research can serve as a foundation for further studies on interoperability issues in blockchain.

References

Alqassem, I. & Svetinovic, D. 2014. *Towards reference architecture for cryptocurrencies: Bitcoin architectural analysis*. Paper presented at the Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE.

Bahri, L. & Girdzijauskas, S. 2019. *Blockchain Technology: Practical P2P Computing (Tutorial)*. Paper presented at the 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W).

Buterin, V. 2016. Chain interoperability. R3 Research Paper, 2018(12 June).

de Kruijff, J. & Weigand, H. 2017. *Understanding the blockchain using enterprise ontology*. Paper presented at the International Conference on Advanced Information Systems Engineering.

Delmolino, K., Arnett, M., Kosba, A., Miller, A. & Shi, E. 2016. *Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab*. Paper presented at the International Conference on Financial Cryptography and Data Security.

Ding, D., Duan, T., Jia, L., Li, K., Li, Z. & Sun, Y. InterChain: A Framework to Support Blockchain Interoperability, accessed 12 march 2020. URL: https://conferences.sigcomm.org/events/apnet2018/posters/6.pdf.

Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., . . . Springsteel, F. 1991. *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*: IEEE Press.

Greenspan, G. 2015. Multichain private blockchain-white paper. URl: http://www.multichain.com/download/MultiChain-White-Paper.pdf

Hardjono, T., Lipton, A. & Pentland, A. J. A. P. A. 2018. Towards a Design Philosophy for Interoperable Blockchain Systems.

Hileman, G. & Rauchs, M. 2017. *2017 Global Blockchain Benchmarking Study*. Rochester, NY: Social Science Research Network

Käll, J. 2018. Blockchain control. *Law and Critique*, **29**(2), 133–140. doi:10.1007/s10978-018-9227-x

Kan, L., Wei, Y., Muhammad, A. H., Siyuan, W., Linchao, G. & Kai, H. 2018. *A Multiple Blockchains Architecture on Inter-Blockchain Communication*. Paper presented at the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C).

Kwon, J. J. D. v., fall. 2014. Tendermint: Consensus without mining.

Li, W., Sforzin, A., Fedorov, S. & Karame, G. O. 2017. *Towards scalable and private industrial blockchains*. Paper presented at the Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts.

Liu, Z., Xiang, Y., Shi, J., Gao, P., Wang, H., Xiao, X., . . . Hu, Y.-C. 2019. *Hyperservice: Interoperability and programmability across heterogeneous blockchains*. Paper presented at the Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security.

Pillai, B., Biswas, K. & Muthukkumarasamy, V. 2019. *Blockchain Interoperable Digital Objects*. Paper presented at the International Conference on Blockchain.

Sklaroff, J. M. 2017. Smart contracts and the cost of inflexibility. *University of Pennsylvania Law Review*, **166**(1), 263.

Tasca, P. & Tessone, C. J. 2017. Taxonomy of blockchain technologies. Principles of identification and classification. arXiv preprint arXiv:1708.04872.

Vernadat, F. 2006. Interoperable enterprise systems: architectures and methods. *IFAC Proceedings Volumes,* **39**(3), 13–20.

Wang, H., Cen, Y. & Li, X. 2017. *Blockchain router: A cross-chain communication protocol*. Paper presented at the Proceedings of the 6th International Conference on Informatics, Environment, Energy and Applications.

Weber, I., Lu, Q., Tran, A. B., Deshmukh, A., Gorski, M. & Strazds, M. 2019. *A platform architecture for multi-tenant blockchain-based systems*. Paper presented at the 2019 IEEE International Conference on Software Architecture (ICSA).

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., . . . Rimba, P. 2017. *A taxonomy of blockchain-based systems for architecture design*. Paper presented at the 2017 IEEE International Conference on Software Architecture (ICSA).

Zheng, Z., Xie, S., Dai, H.-N. & Wang, H. 2016. Blockchain challenges and opportunities: A survey. Work Pap.–2016.

Zheng, Z., Xie, S., Dai, H., Chen, X. & Wang, H. 2017. *An overview of blockchain technology: Architecture, consensus, and future trends*. Paper presented at the Big Data 2017 IEEE International Congress on (BigData Congress).