sustainability

MDPI

Article

Interoperable Permissioned-Blockchain with Sustainable Performance

Swathi Punathumkandi [1,*], Venkatesan Meenakshi Sundaram [1] and Prabhavathy Panneer [2]

[1] Department of Computer Science and Engineering, National Institute of Technology Karnataka, Suratkal, Mangalore 575025, Karnataka, India; venkisakthi@nitk.edu.in
[2] School of Information Technology and Engineering, Vellore Institute of Technology, Katpadi, Vellore 632014, Tamilnadu, India; pprabhavathy@vit.ac.in
* Correspondence: swathip.187co005@nitk.edu.in

Abstract: Bitcoin is an innovative and path-breaking technology that has influenced numerous industries across the globe. It is a form of digital currency (cryptocurrency) that can be used for trading and has the potential to replace fiat money, where the underlying infrastructure is called Blockchain. The Blockchain is an open ledger that provides decentralization, transparency, immutability, and confidentiality. Blockchain can be used in enormous applications, such as healthcare, logistics, supply chain management, the IoT, and so forth. Most of the industrial applications rely on the permissioned blockchain. However, the permissioned blockchain fails in some aspects, such as interoperability among different platforms. This paper suggests a sustainable system to solve the interoperability issue of the permissioned blockchain by designing a new infrastructure. This work has been tested in ethereum and hyperledger frameworks, which obtained a success rate of 100 percent.

Keywords: Bitcoin; cryptocurrency; blockchain; permissioned-blockchain; sustainable; ethereum; hyperledger

check for updates

Citation: Punathumkandi, S.; Sundaram, V.M.; Panneer, P. Interoperable Permissioned-Blockchain with Sustainable Performance. *Sustainability* **2021**, *13*, 11132. https://doi.org/10.3390/su132011132

Academic Editors: Arun Kumar Sangaiah, Xi Zheng, Changqing Luo, Shuihua Wang, Ankit Chaudhary and Piera Centobelli

Received: 6 August 2021
Accepted: 4 October 2021
Published: 9 October 2021

1. Introduction

Blockchain is a decentralized computational and information-sharing platform enabling multiple authoritative domains which do not trust each other to cooperate, coordinate, and collaborate in rational decision-making processes [1]. It is an electronic, decentralized ledger that keeps a copy of all the transactions that take place within the network, which is peer-to-peer. It is a continuous list of transaction records stored in encrypted form, called a "block". Each block is uniquely connected with the previous block by digital signature, so that the record cannot be altered or tampered with without disturbing the records in the previous block of the chain, which makes the blockchain immutable. The unique feature of Blockchain is that there is no need for a third-party authentication mechanism. The transaction becomes valid if the entire peer-to-peer in the network agrees the transaction. One of the applications of Blockchain in crypto currency is bitcoin. Let us see the workings of Blockchain in terms of the Bitcoin transaction life cycle. Consider the scenario where Alice wants to send some coins to Bob—initially, Alice opens her Bitcoin wallet and provides the address of Bob and amount to transfer. Then she presses the send button, and the wallet constructs the transaction which is signed using Alice's private key. By applying digital signature techniques, the wallet signs the transaction made by Alice and broadcasts it over the network. Depending upon the network, all hubs in the system, or the majority of the hubs in the system receive that particular transaction. After receiving the transaction, the nodes in the network will validate the transaction based on the existing blockchain. Once this transaction is validated, then It is propagated to some particular nodes called miners [2]. The miner collects all the transactions for a duration of time, and they construct a new block and try to connect it with the existing blockchain through some cryptographic hash computation, and then they propagate the updated blockchain in the network.

Blockchain can be used in substantial valuable applications, such as medicinal services, academics, banking marketing, and much more. The cryptocurrencies mentioned previously come under the permissionless blockchain. Permissionless blockchains are also known as public or decentralized blockchains. Anyone can create and access the blockchain in which anyone can publish the self-executing contract (a smart contract, which will be explained in Section 2.1). Moreover, anyone can run the blockchain node with 100 percent transparency. However, organizations require an entirely different type of blockchain, which can safeguard their terms and policies. It should incorporate only pre-approved nodes. This type of blockchain is called permissioned blockchains.

The permissioned blockchain requires every peer to execute every transaction, maintain a ledger, and run a consensus (which will be explained in Section 2.2), a fault-tolerant mechanism. It cannot support the valid private transaction with confidential contracts. Hyperledger Fabric is one of the best blockchains which can deliver the modular and secure foundation for the industrial blockchain. The hyperledger fabric generally uses a practical Byzantine Fault Tolerance (PBFT) consensus algorithm. However, some factors pull back the industries from adapting the blockchain fully fledged. Interoperability among platforms is one among them. Interoperability indicates the possibility to freely share value across all blockchain networks without the need for intermediaries. Interoperability among enterprise systems is defined by Vernadat as "a measure of the capacity to execute interoperation between entities" (processes, software, systems, business units) [3]. The issue is to make it easier for various processes and units to "communicate, cooperate, and coordinate". Technical interoperability, legal interoperability, semantic interoperability, integrated public service governance, organizational interoperability, and interoperability governance are some of the interoperability levels. Technical interoperability, for example, is concerned with the technical processes that enable blockchain integration, whereas organizational interoperability is concerned with whether different organizations can work together across different blockchains a shown in Figure 1.

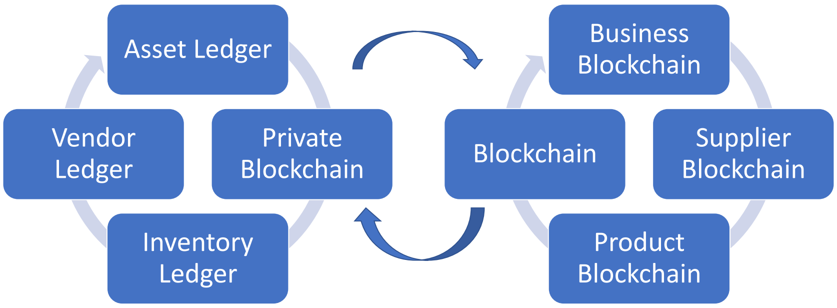

Figure 1. Interoperability among blockchain platforms.

Centobelli et al. [4] provided a broad qualitative and quantitative review of blockchain research using bibliometric analysis methodologies, and also bridged a research gap concerning the absence of a thorough overview of blockchain using a rigorous analytical method. It provides deep insights into current debates, advances the research area linked to the contextual and multilayered phenomena of Blockchain, and leads to future research paths. It is clear that Blockchain technology and its practical use require further scientific, helpful, and legal implementation [5]. The problem of establishing the status of cryptocurrencies and distributed registry technology is now on the table, and both good and bad instances may be used to show the necessity for a balanced and clearly defined economic policy in this field. Despite the fact that interoperability has a broad scope, we primarily focus on technical and organizational interoperability because this is where the majority of blockchain interoperability efforts is centred.

Interoperability will become an important aspect for any project to succeed as the blockchain industry continues to expand and advance. To have hundreds of blockchains totally isolated from each other makes no sense. The ability to exchange knowledge openly through blockchain networks is interoperability. In a completely interoperable world, you will be able to easily read, understand, and communicate with little effort if a user from another blockchain sends something on the blockchain [6]. To allow the above-mentioned use cases, there are three main techniques: notary systems, side-chains, and hash-locking notary schemes. Notary schemes use a trusted party between two blockchains as an intermediary. Therefore, the notary's job is to verify that a blockchain event took place and to feed this information to a second blockchain. Clarity is the key benefit of the notary scheme, as no changes in the underlying blockchains are needed. As one possible solution, a set of notaries they trust might be selected by all parties involved. By using consensus algorithms, such as BFT, the performance of notaries could then be generated. There would be no need to trust every single notary, but just two-thirds of the sidechain community [7]. A sidechain is a blockchain that has the potential to verify and collect data about the status of other blockchains. Although the data need to be fed from one blockchain to the other externally, due to the cryptographic properties of blockchains, this process does not involve trust. It would be easy to produce evidence that the headers were tampered with. By being able to enter the state from other blockchains, sidechains allow for a variety of use cases. To build a sidechain, however, smart contract capabilities are required. In addition, each blockchain will require a sidechain to attain maximum interoperability, which in turn needs to support every other blockchain. Interoperability between various blockchains, interoperability between dApps utilising the same blockchain, and interoperability between blockchain and other technologies were all highlighted by Besancon et al. [8]. According to Vitalik [9], the interoperability solutions sought to enable compatibility between cryptocurrency systems. This category catalogues and specifies several chain interoperability techniques used by public blockchains that allow cryptocurrencies, such as hash time hashlocks, sidechains, and notary schemes. Centobelli et al. [10] pointed out that in the field of circular supply chains, there is a growing corpus of blockchain literature. An increasing interest in the issue necessitates additional practical study on the design and execution of blockchain systems, in addition to the substantial theoretical contribution. The authors [10] analysed six key clusters of blockchain-related research contributions and divided research themes into motor themes, fundamental themes, emerging or fading themes, and specialized themes based on the centrality and density metrics. Even though the majority of contributions are in computer science, many papers on technology management provide valuable information to scholars. While many standards address various aspects of interoperability, there is still space for improvement.

Let us see these concepts in detail.

2. Materials and Methods

This section introduces some important terminologies related to this work.

2.1. Smart Contract

Smart contracts are just like contracts in the real world. The ultimate difference is that they are entirely digital. A smart contract is a compact computer program that is stored inside a blockchain. The smart contract will hold all the received funds until a particular goal is reached. For example, consider the execution of a project. The supporters of the project can transfer their money to the smart contract. If the project gets fully funded, the smart-contract passes all the money to the project's creator [11]. If the project fails to meet the sufficient fund within the time-frame, the money automatically goes to the supporters. Since the smart contract is inside the blockchain, everything is distributed and immutable; hence, the smart contract is completely trustable. Based on the business logic, several functions can be defined within a smart contract.

2.2. Consensus

Consensus mechanisms ensure the records are genuine and honest. Consensus is the basic building block of a distributed ledger [12]. The consensus mechanism ensures that all the transactions occurring on the network are genuine, and all participants agree on an agreement on the ledger's status. Public blockchains, such as bitcoin, use Proof of Work (PoW) as the consensus mechanism. There are vast variants of consensus mechanisms, such as Proof of Stake (PoS), Proof of Authority (PoA), Proof of capacity (PoC), Practical Byzantine Fault Tolerance (PBFT), and so forth. Hyperledger fabric uses PBFT [13].

2.3. Hyperledger Fabric v2.0

Hyperledger Fabric is a stage for distributed record arrangements supported by a private design conveying high levels of secrecy, strength, adaptability, and versatility. It is intended to help pluggable segment's usage and bind the unpredictability and complexities that exist across the financial ecosystem. Hyperledger Fabric has been updating for the last few years. Currently, it is on the v2.x version [14].

2.3.1. Nodes

A blockchain contains a few nodes which interact with each other for processing the transactions. Since hyperledger fabric is a permissioned network, the nodes have a unique identity provided by the membership service provider (MSP). A node can run in physical hardware, a container, or a virtual machine. According to hyperledger fabric, there are three types of nodes, namely, peers, orderers, and clients. The noticeable change in the hyperledger is its peers. The peers are decoupled into endorsers, committers, and consenters. Peers are the nodes that run the transactions and maintain them in the ledger. Peers will receive an ordered state update as a block from the ordering service and maintain it in the ledger, so by default, all peers are committers. Peers have an additional duty as an endorser. They will execute the smart contracts and simulate the transactions. The consenters verify whether the peers have exchanged some assets. Orderers order the transactions. The collection of orderers is termed as an ordering service. Finally, the end-users will be clients; they will send the transaction request to the peers. The clients will coordinate the orders and committers during the verification process.

2.3.2. Transaction Flow

The transaction flow of hyperledger fabric with three endorsing peers and one committing peer will occur as per the convenience of protecting data confidentiality in the transaction. The chaincode references the data collection. Figure 2 explains the transaction flow. The client application submits a proposal request to invoke a chaincode function to endorsing peers that are part of approved organizations. The endorsing peers simulate the dealing and store the non-public information in a temporary data store and send the proposal response back to the client [15]. The response consists of the supported read/write set and a hash of keys. The client application submits the transaction to the ordering service. The hashed transaction gets added to the block and is distributed among peers. The peers will validate the data by checking whether they can access the data during the commit time. If they have the authority to do so, the peers will check in the temporary data store whether their data have already been received [16]. If not, they will pull the data from their peers and validate the data. After validation, the data's copy is moved to private storage and deleted from the temporary storage.

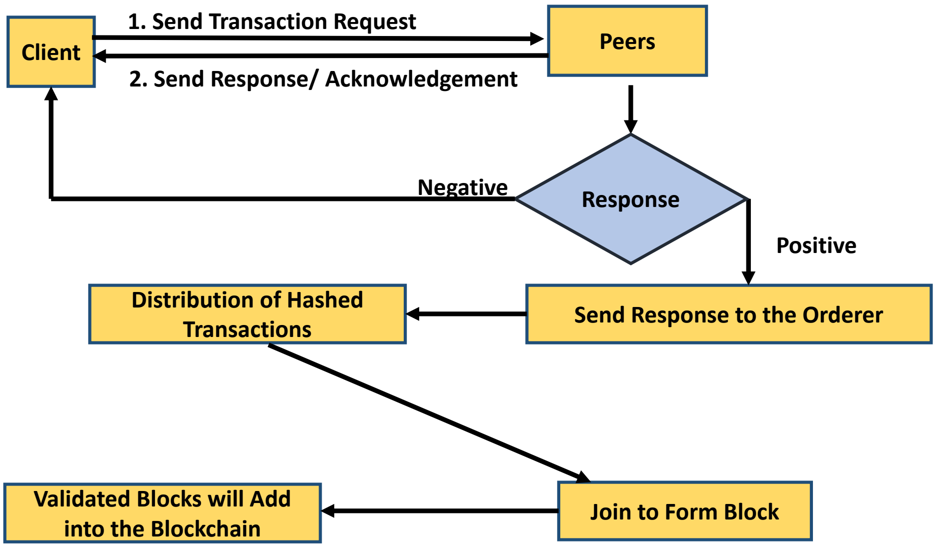
Client
1. Send Transaction Request
2. Send Response/ Acknowledgement
Peers
Response
Negative
Positive
Send Response to the Orderer
Distribution of Hashed Transactions
Join to Form Block
Validated Blocks will Add into the Blockchain

Figure 2. Transaction flow of hyperledger fabric.

2.3.3. Ethereum

In Buterin's article [9], Ethereum was presented to cover some shortcomings of Bitcoin. Ethereum then supports the status of the contract, as well as some other changes to the framework of the blockchain. Ethereum is made up of a network of cryptographic, or protected, public documents that are hard to alter because they are stamped with user data, time and date, and modifications that must be accepted by all users [17]. Anyone may establish a financial arrangement or hold debt or ownership registries on the ledger, removing the need for a third-party record-keeper or trust officer. They are called "trustless" transactions, and they do not include trusting the transaction's counter-party. Ethereum is a permissionless, non-hierarchical computer (node) network that builds and decides on an ever-growing sequence of "blocks" known as the blockchain. Whenever a node attaches a block to the chain, the transactions are always executed in their order and modify the Ethereum account storage values. A relatively small subset of the network, known as its peers, connects with each node. The transaction flow of Ethereum is depicted in Figure 3. Whenever a node tries to add a new transaction to the blockchain, it sends it to its peers, who send it to their peers, and so on. It travels across the network this way. Some nodes, known as miners, hold a list of all these recent transactions and use them to create new blocks, which are then sent to the rest of the network. Whenever a node receives a block, the validity of the block and of all its transactions is checked and, if correct, added to its blockchain, and all such transactions are executed. A node can obtain competing blocks, which may form competing chains, since the network is non-hierarchical. The network achieves unity on the blockchain by applying the "longest chain law", which specifies that the canonical chain is the one with the most blocks at any given time. Since miners do not want to spend their computing energy attempting to connect blocks to a chain that would be abandoned by the network, this rule achieves consensus.

By considering the above explanations of Bitcoin, Hyperledger Fabric and Ethereum, it is clear that all the platforms are entirely different and they work in their own way. There comes the role of interoperability.

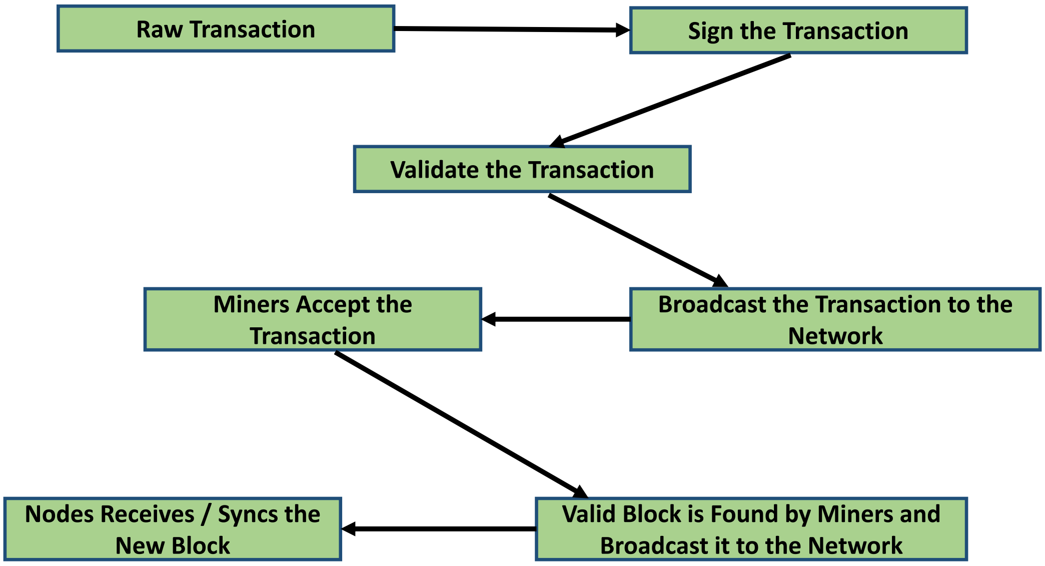
Raw Transaction
Sign the Transaction
Validate the Transaction
Miners Accept the Transaction
Broadcast the Transaction to the Network
Nodes Receives / Syncs the New Block
Valid Block is Found by Miners and Broadcast it to the Network

Figure 3. Transaction flow of Ethereum.

3. Proposed Methodology

Interoperability on the blockchain can be achieved using a variety of methods. According to the classes of strategies we defined, we divide blockchain interoperability into three categories: cryptocurrency-directed interoperability methods, Blockchain Engines, and Blockchain Connectors [18]. There are sub-categories within each division. It describes and establishes various chain interoperability techniques through public blockchains, the majority of which use cryptocurrencies. The next category focuses on general use-cases and heterogeneous systems, while the cryptocurrency-based interoperability approaches a category focused on cryptocurrency ecosystems, often homogeneous blockchain structures. Blockchain Engines are platforms that include reusable data, network, agreement, opportunity, and contract layers for building customizable blockchains that power decentralised apps. The use of tokens is included in this grouping, and is mostly used as an incentive tool for participants to adopt protocols and manage the network. The Blockchain Connector type includes non-cryptocurrency interoperability applications, as well as blockchain engines. Trusted Relays, Blockchain Agnostic Protocols, Blockchain of Blockchains, and Blockchain Migrators are some of the sub-categories we extracted from the studies [18]. Above all, this work has utilised a notary scheme for structuring a new framework, so that it may use a trusted party between two blockchains as an intermediary. Therefore, the notary's job is to verify that a blockchain event took place and to feed this information to a second blockchain.

This work has considered two blockchain platforms: Hyperledger Fabric, a permissioned blockchain, and Ethereum, a permissionless blockchain. The notary scheme of Fabric and Ethereum is depicted in Figure 4. The supporting layers (e.g., networking, storage, and encryption) are used to build the consensus engine, which organises transactions and appends them to the chain of blocks [19]. Hyperledger Fabric's consensus is modular and based on endorsement policies. A client (C) submits a transaction proposal to the peer nodes (P) and gets an endorsement (a signed transaction) in Fabric. The endorsements are checked by an orderer, who then produces a block of legal transactions that is added to the ledger. A node can suggest a block of transactions to be added to the ledger after discovering a PBFT solution. Because of the fundamental differences between the two types of blockchains, the interoperability challenge is unique. There are multiple layers for a blockchain [20]. The data layer defines how data on the blockchain are interpreted (e.g., transactions piled into blocks vs. transactions represented in a directed acyclic graph). The network layer defines the node category in a peer-to-peer network [1]. The consensus algorithm, as well as its security assumptions, are part of the consensus layer. The contract layer contains the smart contract execution environment, which provides the framework for the application layer, and includes blockchain-enabled corporate logic [21].

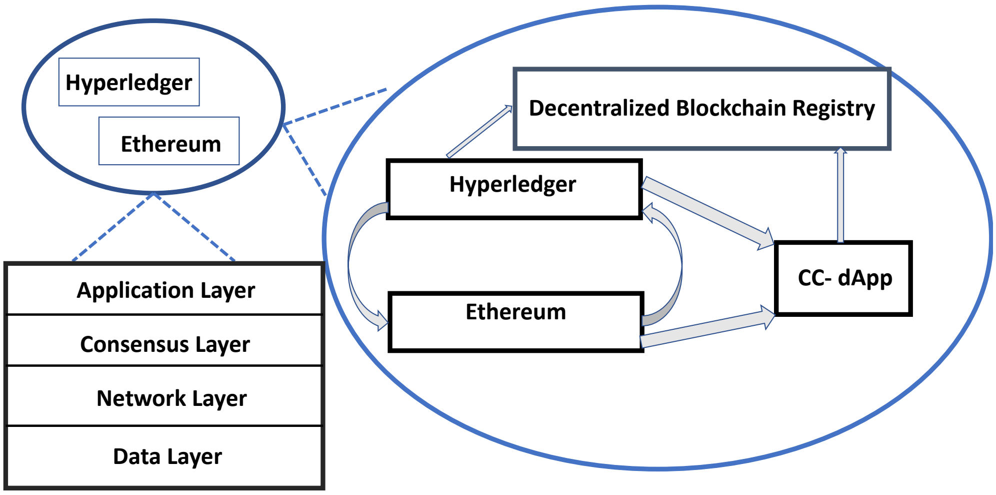

Figure 4. Network of Ethereum and Hyperledger Fabric in the notary scheme.

This methodology considered each blockchain as a self-contained system that connects with others via a cross-chain protocol that includes a notary mechanism. Interoperability gateways are created by nodes on both public and private blockchains. To encourage interaction across blockchains, decentralised blockchain registries that can recognise and address blockchains and their components (e.g., smart contracts and certificate authorities) can be utilised [22]. A repository for both public and private blockchains might be created on a public blockchain with solid security assumptions [23]. The registry's information is maintained in a customised shared database administered by significant blockchain players [24]. The decentralised repository would work similarly to a decentralised domain name structure.

Consolidating all of these data resulted in the creation of a new architecture for addressing the interoperability issue. Figure 5 shows a concept for the Hyperledger Fabric chaincode, which includes Ethereum integration. Ethereum is a permissionless and EVM-based blockchain, while Hyperledger Fabric is a permissioned blockchain. An Ethereum blockchain node's interaction endpoint (i.e., IP address) is registered with the blockchain registry. After that, it looks for the address of a Hyperledger fabric server with which it is supposed to communicate. The Cross-Chain Communication Protocol (CCCP) and the Cross-Blockchain Communication Protocol (CBCP) provide for unidirectional or bidirectional interoperability (CBCP). Because the Hyperledger node interprets Ethereum's block headers but not the other way around, a CBCP allows the Ethereum and Hyperledger nodes to communicate arbitrarily . A CC-dApp that is already linked to Ethereum and Hyperledger fabric utilises the private blockchain to create the required credentials after getting its address from the blockchain registry. A CC-dApp protocol lets an end-user achieve semantic interoperability by utilising Hyperledger fabric and Ethereum. These actions establish blockchain connection, culminating in the establishment of connection among Hyperledger Fabric and Ethereum. Existing blockchains would require changes to multiple levels, including the network, consensus, contract, and application layers. In essence, this work uses the Hyperledger Fabric permissioned blockchain infrastructure to allow users to interact with Ethereum smart contracts written in an EVM (Ethereum virtual machine) compatible language called Solidity. To complete the integration, the EVM chaincode (EVMCC) and the web provider are utilised. The EVMCC is a Go chaincode that encapsulates the Hyperledger fabric EVM bundle and maps out the various ways between the peer and the EVM. The EVMCC acts as a smart contract runtime, placing the implemented contract code on the ledger in the EVMCC namespace. Users may connect with smart contracts running in the Fabric EVM using tools like Web3.js. A proxy that provides a subset of Ethereum-compliant JSON RPC APIs. The Fabric GO SDK allows the proxy to connect to the Fabric network and communicate with the EVMCC. The Ethereum Smart Contract Runtime and the Hyperledger Fabric Runtime are being rebuilt by the EVMCC and proxy. Applications that employ the Ethereum JSON RPC API and EVM smart contracts should be able to interact seamlessly with Hyperledger Fabric. Fabreum is the name given to this innovative design since it functions as both Ethereum and Fabric as shown in Figure 6.

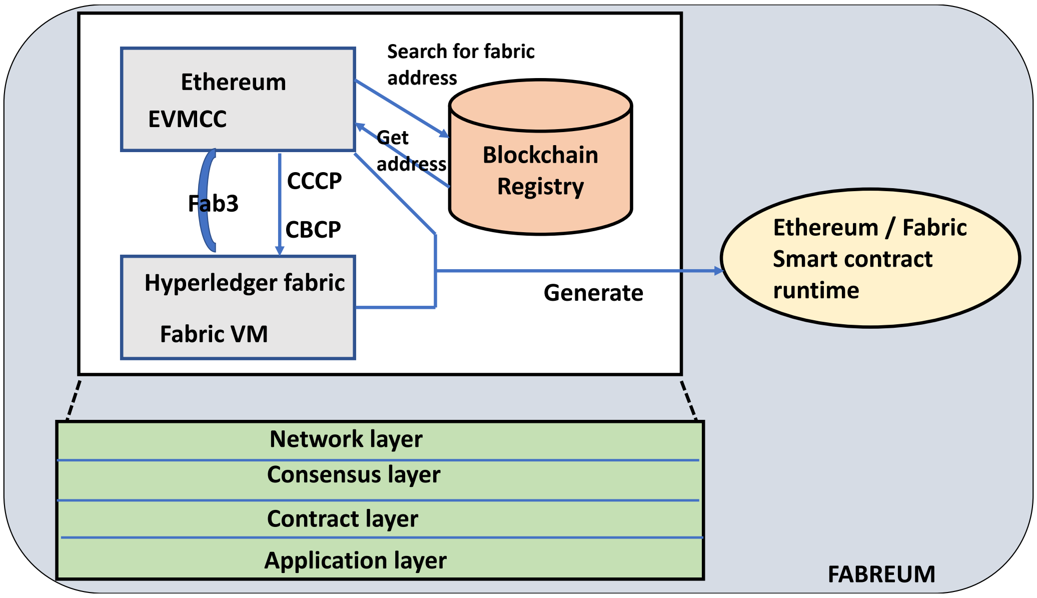

Figure 5. Interoperability framework of Ethereum and Fabric in detail.

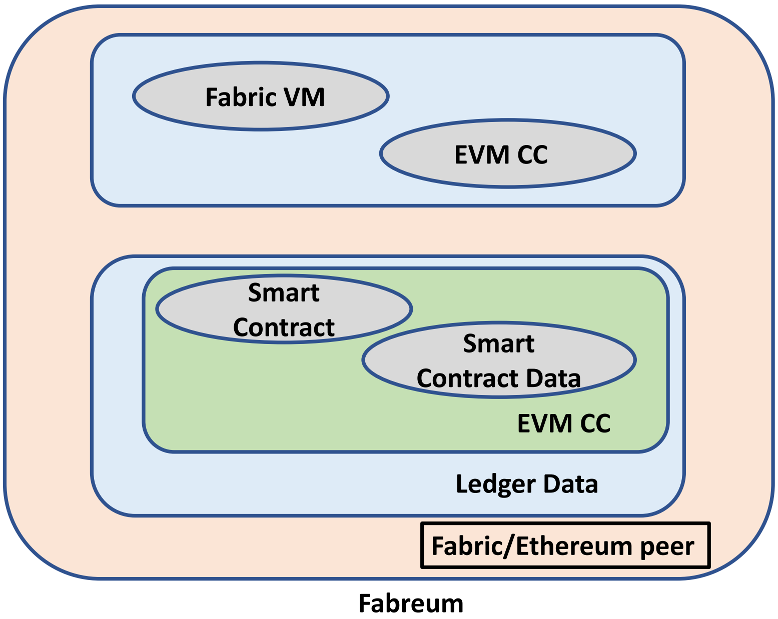

Figure 6. Consolidated interoperability framework of Ethereum and Fabric.

4. Result and Analysis

Two blockchain networks were established at NIT Karnataka utilising Hyperledger Fabric and Ethereum installed on a 64-bit Ubuntu operating system with an 8 GB RAM for performance assessment and validation of the suggested solution direction. The tests were conducted using a Hyperledger Fabric version 2.0, as well as Ethereum 1.10.8, which is the most recent version accessible at the time of our research. Three organisations were considered in this study. There are N organisations in a network of N peers, since each has one peer, one Certificate Authority client (CA), and one Membership Service Provider (MSP). A single channel connects all of the organisations. The experiments' chaincode is written in the Golang programming language. SOLO and Kafka-based ordering systems are two distinct ordering services developed in Fabric. Since SOLO is only intended for testing and not for usage in a production context, the suggested work employs Kafka in the tests. The test gathers performance metrics using caliper.

The performance matrix obtained from caliper is shown in Table 1. The experiments were done by taking Ethereum and Hyperledger Fabric as the source and destination, respectively, and vice versa for 500 transactions. The same experiment was even done for Ethereum as a source as well as destination, and Hyperledger as a source as well as destination. The comparison of an output of interoperability before and after applying the solution directive is illustrated in Figure 7. All 500 transactions in each of the cases became successful, and each case obtained a good level of latency and throughput. This experiment has resulted in an average of a 25.55 tps send rate. The send rate is nothing but the number of transactions sent per second.

$$Send\ rate = total\ number\ of\ transactions\ send/total\ time.$$

All of the cases showed a similar pattern in send rate, which means the transactions happen irrespectively of the sender or receiver. Transaction latency is the measure of time produced for an exchange's results to be usable over the system [25]. There is a similar pattern of transaction latency visible in the matrix. The transaction latency can be considered from two perspectives: the number of peers at which the exchange is seen to be settled, and the percentage of perceptions equivalent to or beneath where the estimation is substantial (percentile). Transaction latency is generally reported as the average latency, which is determined as follows:

$$Average\ Transaction\ Latency = sum\ of\ transaction\ latency/total\ committedtransactions,$$

and here we obtain an average of 8.96 s. In the same way, we collected an average transaction throughput of 13.25. The transaction throughput is the rate at which the blockchain arrangement submits legitimate exchanges in the defined timeframe. The throughput of transactions is the rate at which legitimate transactions are committed. Therefore,

$$Transaction\ Throughput = total\ committed\ transaction/total\ time.$$

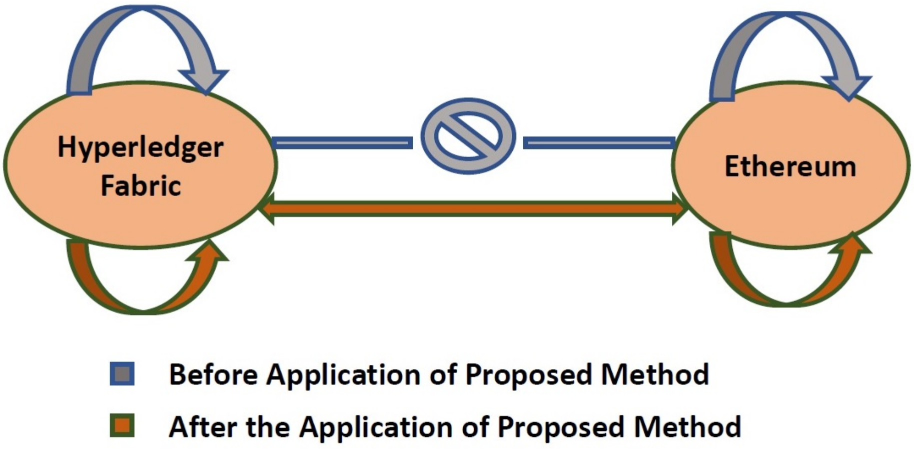
Hyperledger Fabric
Ethereum
Before Application of Proposed Method
After the Application of Proposed Method

Figure 7. An illustration as to the comparison of interoperabiliy before and after applying the proposed solution.

Table 1. Performance matrix of the system after applying the interoperability solution.

Src	Destn	Succ	Fail	Send Rate	Max Latency	Min Latency	Throughput
Fabric	Ethereum	500	0	25.2 tps	18.39 s	0.78 s	13.2 tps
Fabric	Ethereum	500	0	25.3 tps	18.56 s	0.78 s	13.4 tps
Ethereum	Fabric	500	0	25.9 tps	18.43 s	0.79 s	13.2 tps
Fabric	Ethereum	500	0	25.4 tps	18.24 s	0.78 s	13.2 tps
Ethereum	Fabric	500	0	25.6 tps	18.37 s	0.77 s	13.2 tps
Ethereum	Fabric	500	0	25.7 tps	18.46 s	0.78 s	13.5 tps
Fabric	Ethereum	500	0	25.4 tps	18.95 s	0.79 s	13.1 tps
Ethereum	Ethereum	500	0	25.6 tps	18.21 s	0.78 s	13.3 tps
Ethereum	Ethereum	500	0	25.8 tps	18.32 s	0.79 s	13.1 tps
Fabric	Fabric	500	0	25.4 tps	18.54 s	0.77 s	13.4 tps
Ethereum	Fabric	500	0	25.7 tps	18.47 s	0.78 s	13.3 tps
Fabric	Fabric	500	0	25.7 tps	18.56 s	0.78 s	13.2 tps

Vo et al. [22] focused on interoperability architecture, providing some Blockchain of Blockchain and contract solutions. Buterin et al. [9] gave an overview of public connectors, including notary methods, sidechains, and hash-time locking mechanisms. Conversely, other studies concentrated on public connections, with a particular focus on sidechains and hash lock time contracts [26,27]. Meanwhile, Qasse et al. [28] arranged solutions across sidechains, blockchain routers, smart contracts, and industrial solutions. Siris et al. [29] and, Kannengiesser et al. [30] did a survey on interoperability problems, and Johnson et al. [31] and Koens et al. [32] concentrated on Ethereum as the framework that allows for interoperability across various types of applications. When looking at these literatures, it is apparent that they mainly focused on public blockchains, particularly cryptocurrencies. This study, on the other hand, is primarily focused on the interoperability of permissioned and permission-less blockchain blockchain networks. The following is a list of the observations made during this work.

Observations

- Hyperledger fabric has been designed to interact with Ethereum smart contract.
- Interaction is achieved through EVM Chaincode and fabric vm.
- Ethereum vm chaincode wraps the Hyperledger Fabric in a GO chaincode, together named Fabreum.
- The Ethereum chaincode acts as the smart contract runtime and stores the deployed contract on the ledger.
- Combining both Fabric and Ethereum will act as twins so that features can be incorporated.
- Obtained 100 percent success rate in 500 transactions with better latency and throughput.

5. Conclusions

Blockchain innovations have developed quickly in the current decade. The involvement of Blockchain in lifestyles is not so far off. With the expanding reception of blockchain innovation, the quantity of clients has consistently expanded. However, its performance still needs much improvement compared with the mainstream processors. The system blockage of the existing framework is a common issue, and experts are cautiously considering how to settle down the interoperability issue. The proposed solution directive put forward a new framework for solving the interoperability issue by incorporating EVM Chaincode and fabric vm. This paper included applying the solution in the Hyperledger Fabric and Ethereum framework and analyzing the resultant system in terms of throughput, latency, and successful rates of transaction. The proposed method exhibits better throughput and latency for all cases of source and destination combos for all 500 transactions. In future works, we can aim at generalising the framework for all platforms. Many current products aspire for this status, and a universal interoperable network will be an ideal option for Blockchain interoperability concerns. No-one knows for sure that this will ever be the case. Finally, almost all of these platforms are in competition with one another. In future, the research can extend to various application levels.

Author Contributions: S.P.: Conceptualization and design of the work, methodology, data collection, data analysis and interpretation and drafting the article; V.M.S.: Supervision, critical revision of the article and final approval of the version to be published; P.P.: Revision of the article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System, Cryptography Mailing List. 2009. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 14 August 2018).
2. Swathi, P.; Modi, C.; Patel, D. Preventing Sybil Attack in Blockchain using Distributed Behavior Monitoring of Miners. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6. [CrossRef]
3. Vernadat, F.B. Interoperable enterprise systems: Architectures and methods. *IFAC Proc. Vol.* **2006**, *39*, 13–20. [CrossRef]
4. Centobelli, P.; Cerchione, R.; Esposito, E.; Oropallo, E. Surfing blockchain wave, or drowning? Shaping the future of distributed ledgers and decentralized technologies. *Technol. Forecast. Soc. Chang.* **2021**, *165*, 120463. [CrossRef]
5. Karpenko, L.; Akhlamov, A.; Onyshko, S.; Chunytska, I.; Starodub, D. Blockchain as an Innovative Technology in the Strategic Management of Companies. *Acad. Strateg. Manag. J.* **2019**, *18*, 1–6.
6. Jolma, A.; Rizzoli, A.-E. A Review of Interoperability Techniques for Models, Data, and Knowledge in Environmental Software. 2003. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.553.1526&rep=rep1&type=pdf (accessed on 21 October 2018).
7. Mockapetris, P.; Dunlap, K.J. Development of the domain name system. In Proceedings of the Symposium Proceedings on Communications Architectures and Protocols, SIGCOMM, Stanford, CA, USA, 16–18 August 1988; Association for Computing Machinery: New York, NY, USA, 1988; pp. 123–133. [CrossRef]
8. Besancon, L.; Ferreira da Silva, C.; Ghodous, P. Towards Blockchain Interoperability: Improving Video Games Data Exchange. 2019. pp. 81–85. Available online: https://ieeexplore.ieee.org/document/8751347 (accessed on 5 June 2020).
9. Buterin, V. A Next-Generation Smart Contract and Decentralized Application Platform, White Paper. 2015. Available Online:https://people.cs.georgetown.edu/~clay/classes/fall2017/835/papers/Etherium.pdf (accessed on 28 December 2019).
10. Centobelli, P.; Cerchione, R.; Vecchio, P.D.; Oropallo, E.; Secundo, G. Blockchain Technology for Bridging Trust, Traceability and Transparency in Circular Supply Chain, Information and Management. 2021. Available online: https://www.sciencedirect.com/science/article/pii/S0378720621000823 (accessed on 27 August 2021).
11. Szabo, N. Smart Contracts : Building Blocks for Digital Markets. 2018. Available online: https://www.researchgate.net/publication/340376424_Smart_Contracts_Building_Blocks_for_Digital_Transformation (accessed on 18 January 2020).
12. Ongaro, D.; Ousterhout, J. In search of an understandable consensus algorithm. In Proceedings of the USENIX Annual Technical Conference, Philadelphia, PA, USA, 19–20 June 2014.
13. Sousa, J.; Bessani, A.; Vukolic, M. A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In Proceedings of the 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Luxembourg, 25–28 June 2018; pp. 51–58.
14. IBM Corporation. Hyperledger-Fabricdocs Documentation. Technical Report. 2019. Available online: https://hyperledger-fabric.readthedocs.io/_/downloads/en/release-1.4/pdf/ (accessed on 19 September 2020).
15. Androulaki, E. Barger, Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. 2018. Available online: https://arxiv.org/pdf/1801.10228.pdf (accessed on 20 November 2020).
16. Krsti, M.; Krsti, L. Hyperledger frameworks with a special focus on hyperledger fabric. *Vojnoteh. Glas.* **2020**, *68*, 639–663. [CrossRef]
17. Sompolinsky, Y.; Zohar, A. Secure High-Rate Transaction Processing in Bitcoin. 2015. Available online: https://link.springer.com/chapter/10.1007/978-3-662-47854-7_32 (accessed on 2 March 2019).

18. Belchior, R.; Vasconcelos, A.; Guerreiro, S.; Correia, M. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. 2020. Available online: https://arxiv.org/pdf/2005.14282.pdf (accessed on 20 July 2021).
19. 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.
20. Zhu, Q.; Loke, S.W.; Trujillo-Rasua, R.; Jiang, F.; Xiang, Y. Applications of distributed ledger technologies to the internet of things: A survey. *ACM Comput. Surv.* **2019**, *52*. [CrossRef]
21. Dinh, T.T.A.; Wang, J.; Chen, G.; Liu, R.; Ooi, B.C.; Tan, K.-L. Blockbench: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD '17, Chicago, IL, USA, 14–19 May 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 1085–1100. [CrossRef]
22. Vo, D.K.H.T.; Wang, Z. Internet of blockchains: Techniques and challenges ahead. In Proceedings of the 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; pp. 1574–1581.
23. Zheng, Z.; Xie, S.; Dai, H.-N.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the IEEE International Conference on Services Economics (SE), Honolulu, HI, USA, 25–30 June 2017.
24. Hardjono, T.; Lipton, A.; Pentland, A. Toward an interoperability architecture for blockchain autonomous systems. *IEEE Trans. Eng. Manag.* **2020**, *67*, 1298–1309. TEM.2019.2920154. [CrossRef]
25. Zhou, Q.; Huang, H.; Zheng, Z. Solutions to scalability of blockchain: A survey. *IEEE Access* **2020**, *8*, 16440–16455.
26. Abebe, E.; Behl, D.; Govindarajan, C.; Hu, Y.; Karunamoorthy, D.; Novotny, P.; Pandit, V.; Ramakrishna, V.; Vecchiola, C. Enabling enterprise blockchain interoperability with trusted data transfer (industry track). In Proceedings of the 20th International Middleware Conference, Davis, CA, USA, 9–13 December 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 29–35.
27. Barber, S.; Boyen, X.; Shi, E.; Uzun, E. Bitter to Better—How to MakeBitcoin a Better Currency. In *International Conference on Financial Cryptography and Data Security*; Technical Report; Springer: Berlin/Heidelberg, Germany, 2012.
28. Qasse, I.A.; Talib, M.; Nasir, Q. Inter blockchain communication: Asurvey. In Proceedings of the ArabWIC 2019, Rabat, Morocco, 7–9 March 2019.
29. Siris, V.A.; Nik, P.; Voulgaris, S.; Fotiou, N.; Lagutin, D.; Polyzos, G.C. Interledger approaches. *IEEE Access* **2019**, *7*, 89948–89966. [CrossRef]
30. Kannengiesser, N.; Pster, M.; Greulich, M.; Lins, S.; Sunyaev, A. Bridges between Islands: Cross-Chain Technology for Distributed Ledger Technology. 2020. Available online: https://www.researchgate.net/publication/335867834_Bridges_Between_Islands_Cross-Chain_Technology_for_Distributed_Ledger_Technology (accessed on 7 August 2021).
31. Johnson, S.; Robinson, P.; Brainard, J. Sidechains and Interoperability. 2019. Available online: https://arxiv.org/pdf/1903.04077.pdf (accessed on 20 August 2021).
32. Koens, T.; Poll, E. Assessing interoperability solutions for distributed ledgers. *Pervasive Mob. Comput.* **2019**, *59*, 101079. [CrossRef]