IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 10, 2021, accepted August 17, 2021, date of publication August 20, 2021, date of current version August 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106384

Towards Interoperable Blockchains: A Survey on the Role of Smart Contracts in Blockchain Interoperability

SAJJAD KHAN[1], MUHAMMAD BILAL AMIN[2], AHMAD TAHER AZAR[3,4], (Senior Member, IEEE), AND SHERAZ ASLAM[5], (Member, IEEE)

[1]COMSATS University Islamabad, Islamabad 45550, Pakistan
[2]College of Sciences and Engineering, University of Tasmania, Hobart, TAS 7005, Australia
[3]College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia
[4]Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt
[5]Department of Electrical Engineering, Computer Engineering, and Informatics, Cyprus University of Technology, 3036 Limassol, Cyprus

Corresponding authors: Sajjad Khan (sajjadkhancui@gmail.com) and Ahmad Taher Azar (aazar@psu.edu.sa)

This work was supported by Prince Sultan University.

ABSTRACT The slower than expected adoption rate of blockchain technology has highlighted that there are barriers due to the diversity of its applications and its users. To overcome this limitation and take full advantage of the novel technology, researchers from academia as well as industry are dedicated to find different solutions, where two or more blockchains can interact with each other. As a result, several interoperability solutions have presented themselves. To investigate the functionalities and underlying mechanisms of interoperable blockchain solutions, researchers have conducted several surveys by discussing the features and innovations of these methods. However, the existing surveys tend to focus on the architectural description of the interoperability solutions and completely overlook the most promising aspect of blockchain adaptability, namely the smart contract. This paper fills the gap by exploring the role of smart contracts in blockchain interoperability solutions. Our research has classified the existing interoperability solutions into three main categories: heterogeneous blockchains and homogeneous smart contracts, homogeneous blockchains and homogeneous smart contracts, heterogeneous blockchains, and heterogeneous smart contracts. To provide a systematic overview of the smart contracts used in blockchain interoperability, each category is further divided into subcategories by identifying the functionalities of the smart contract used. Based on our survey, a taxonomy is proposed to help classify the blockchain interoperability solutions. The interoperability solutions in each category are analyzed in-depth, and the results are presented in tabular format to illustrate the characteristics of the interoperability solutions in a meaningful way. Finally, a number of open issues and research directions are discussed to overcome the limitations and improve the performance of blockchain interoperability.

INDEX TERMS Blockchain interoperability, smart contracts, chain code, cross chain transactions, sidechains, decentralized applications, cryptocurrency, digital assets.

I. INTRODUCTION

Blockchain has revolutionized the finance industry by introducing cryptocurrencies to the digital world [1]. It is the core technology that enabled the transfer of cryptocurrencies by ensuring consistency in a decentralized environment without mutual trust. However, the applications and use of blockchains were limited only to the storage and transfer of value of Bitcoin. With the development of competing digital ledger technologies such as Ethereum [2], smart contracts were introduced into the blockchain architecture. Smart contracts transformed blockchains from a mere ledger to a programmable machine. This opens the doors for many individuals and organizations from academia and industry

The associate editor coordinating the review of this manuscript and approving it for publication was Mehedi Masud.

to work on Blockchain. As a result, a number of platforms with smart contract capabilities have been developed (e.g., Corda [3], Quorum [4] and Hyperledger Fabric [5], etc.). Today, the blockchain domain is very rich. Moreover, due to its temper-resistance ledger with zero-knowledge proof and smart contract capabilities in the decentralized environment, it has gained attraction from various fields such as cloud computing [6], Internet of Things (IoT) [7], [8], energy domain [9], food supply chain [10] and education [11], etc. The application of blockchain varies from cross domain data sharing for industrial IoT [12], privacy preserving data aggregation model in smart gird [13] and electronic health records [14], etc.

According to Gartner, the blockchain industry has poured billions of dollars into research [15]. Both government and private sector organizations are betting on the use of blockchain technology. However, the growing interest in this area is focusing on creating new types of Blockchains. Although these new blockchains are capable of meeting the changing needs of users, they lack the ability to interact or communicate with each other. Thus, if a user on one blockchain wants to interact with another user on a different blockchain, they must either have an account on that blockchain or switch to the target blockchains. The advantage of choosing a new blockchain over an existing blockchain allows a user to take advantage of the advanced features of modern technology. The disadvantage of choosing a novel blockchain, on the other hand, involves security risks due to the immaturity and security concerns of novel blockchains [16]. In short, the ways in which two blockchains can interact with each other remain unexplored.

To solve the problem of blockchains' inability to interact and communicate with each other, Adam *et al.* proposed [17] sidechains. ''A sidechain refers to a secondary blockchain that validates data from other blockchains''. A pegged sidechain refers to a blockchain that has the ability to import and export digital property (i.e., coins, assets, etc.) from other blockchains at an agreed-upon price or exchange rate. It allows the transfer of digital properties between the two blockchains using a Simplified Payment Verification (SPV).[1] Sidechains are independent blockchains. Each sidechain has its own consensus mechanism and security protocols. Similarly, each blockchain is sovereign to implement its own identity management and cryptographic algorithms [18]. Therefore, malicious attempts on one sidechain cannot affect the performance of the main or parent blockchain in the event of interoperation between the interacting chains.

Although sidechains are emerging as promising solutions, they are used for a specific use case, namely connecting the parent blockchain to a secondary blockchain to exchange tokens by locking a certain set of tokens on one chain and releasing the corresponding set of tokens on the secondary chain using trusted or semi-trusted intermediaries [19]. Since the introduction of smart contracts, the applications of blockchains are no longer limited to token creation and management, a number of platforms with smart contract capabilities have emerged to connect blockchains. Some of the well-known solutions are Cosmos [20], Polkadot [21], AION [22] that connects independent blockchains using an intermediate chain, Tokrex [23], Blocknet [24], Agent chain [25], Komodo [26] using decentralized exchanges, Scheid *et al.* [27] by developing a policy-based framework, Testimonium [28] a validation relay on demand, Zendoo [29] a sidechain capable of creating, communicating and integrating new sidechains with the main chain, Ghaemi *et al.* [30] by developing a publisher/subscriber architecture and Frauenthaler *et al.* [31] using a dynamic framework to switch users from one blockchain to another blockchain. Although all of these solutions aim to achieve blockchain interoperability. There are significant trade-offs between all existing solutions. Therefore, our goal in this paper is to provide an overview of interoperability solutions. Although a number of authors have studied interoperability approaches, their work focused on studying interoperability from an architectural perspective, i.e., none of the existing work studied the role of smart contracts in interconnecting blockchains. Therefore, by studying the role of smart contracts in interconnecting blockchains, interoperability can be greatly improved. The main reason why smart-contract-based interoperability will outperform architecture-based interoperability solutions is its ability to enforce contractual terms in decentralized environments. When implemented properly, smart-contract-based interoperability solutions or cross-chain smart contracts can minimize the excess cost and time required to develop an architecture-based interoperability solution to a greater extent. The main contributions of this paper are as follows.

1) This study defined several areas through which interoperability between blockchains can be achieved.
2) This study provides an overview of solutions for interoperable blockchains by discussing the role of smart contracts in blockchain interoperability. To the best of our knowledge, this is the first study on the role of smart contracts in blockchain interoperability.
3) A classification of interoperable blockchain solutions is provided on the basis of blockchain type and programming languages used to develop smart contracts used in the interoperable blockchain solutions. A hierarchical taxonomy diagram is created in order to help readers to explore the field of blockchain interoperability.
4) For each category and subcategories, the corresponding interoperability solution is analyzed. The in-depth analysis of each solution is summarized in tables to comprehensively present the results.
5) This study examined the main issues of each interoperability solution and analyzed their implications with a detailed discussion.

[1]It consists of block headers and cryptographic proof to show that a particular output has been created.

TABLE 1. Summary of the existing surveys.

References	Permission type (Permissioned/Public)	Blockchain type (homogeneous/heterogeneous)	Smart contract scripts	Arbitrary data	Comparison	Role of smart contracts
Buterin [40]	Public	Both	✗	✗	✗	✗
Borkowski et al. [41]	Public	Both	✓	✗	✗	✗
Borkowski et al. [42]	Public	Homogeneous	✗	✗	✗	✗
Koens et al. [43]	Public	Both	✗	✗	✓	✗
Singh et al. [44]	Public	Both	✗	✗	✓	✗
Siris et al. [45]	Public	Both	✗	✗	✓	✗
Kannengießer et al. [46]	Public	Both	✗	✗	✓	✗
Mahdi et al. [47]	Public	Both	✗	✗	✗	✗
Sandra et al. [48]	Permissioned	Homogeneous	✗	✗	✓	✗
Qasse et al. [49]	Public	Heterogeneous	✗	✗	✗	✗
Vo et al. [50]	Both	Both	✗	✓	✗	✗
Schulte et al. [51]	Public	Both	✗	✗	✗	✗
Belchior et al. [52]	Both	Both	✗	✓	✓	✗
This survey	**Both**	**Both**	✓	✓	✓	✓

6) Finally, open issues, challenges, and future directions are discussed to enhance the performance of interoperability solutions in the blockchain.

The rest of this paper is organized as follows. A literature review of the existing surveys is discussed in section II. The preliminary on smart contracts and programming languages are discussed in section III. Interoperability and the various approaches through which interoperability between blockchains can be achieved is discussed in section IV. Section V discusses the role of smart contracts in interoperability solutions. The future directions and challenges are discussed in section VI. Lastly, section VII concludes the paper.

II. EXISTING SURVEYS

This section discusses existing surveys on blockchain interoperability. Due to the novelty of this area, the number of peer-reviewed articles is limited. Therefore, to give the reader a clear idea of all the efforts done in this area, self publications are also included in this survey. A summary of the existing surveys is given in table 1. In table 1, permission type represents blockchain permissions such as public or private blockchains. Blockchain type represents blockchain type such as homogeneous or heterogeneous blockchains. Smart contract scripts represent if (or not) the existing surveys discussed the smart contracts scripts or programming languages of the interacting blockchains. Arbitrary data shows if (or not) the existing study review interoperability techniques by considering data transfer across blockchains. Comparison represents if (or not) the existing survey has performed any comparison/analysis of the interoperability solutions. The role of smart contracts shows if (or not) the authors discussed the role of smart contracts in interoperability.

Buterin [40] classified interoperability solutions into three categories: notary schemes, sidechains/relays, and hash locks. The author discussed that interoperability can be achieved for portable assets, payment vs payment or payment vs delivery methods, cross-chain assets, asset encumbrance, and general cross-chain smart contracts. Borkowski *et al.* [41] discussed atomic cross-chain transfer for token exchange and user-issued assets. This work is extended in [42] by studying the work done by the Token Atomic Swap Technology (TAST) project. Moreover, the authors highlighted a number of open issues, challenges, and possible research directions. Schulte *et al.* [51] discussed token transfer and smart contract interaction in cross-blockchain transactions.

Koens *et al.* [43] assessed two of the renowned blockchain interoperability solutions provided by Cosmos and Polkadot using twelve key properties. Singh *et al.* [44] and Sandra [48] surveyed interoperability solutions in sidechain technologies. Siris *et al.* [45] classified inter ledger approaches into six categories: atomic cross-chain transaction, bridging, sidechains, the inter ledger protocols, transactions across a network, and ledger of ledgers. In this work, the authors focused on how the ledgers are interconnected. Moreover, an analysis is performed on the exchange of values or assets, transaction cost, complexity, scalability, and inter ledger trust mechanisms. Kannengießer *et al.* [46] categorized inter ledger approaches into four categories: manual asset exchange, notary schemes, relays, and hybrid solutions. Miraz *et al.* [47] classified atomic cross-chain swaps into on-chain atomic swaps and off-chain atomic swaps. In this work, the authors analyzed the pros, cons, and key challenges associated with atomic cross-chain swaps in cryptocurrencies.

Qasse *et al.* [49] classified interoperability into four categories namely sidechains, industrial solutions, smart contracts, and blockchain routers. This work also discussed inter-blockchain communication approaches. Vo *et al.* [50] discussed interoperability solutions that support communication and interconnection between multi-chain architecture. This work focused on the Internet of blockchains and inter-blockchain communication between the interacting blockchains. Belchior *et al.* [52] classified interoperability solutions into three categories: cryptocurrency directed approach, blockchain engines, and blockchain connectors. The cryptocurrency directed approach is further categorized into sidechains, notary schemes, hashed time-locks, and hybrid solutions. Similarly, the blockchain connectors are further categorized into trusted relays, blockchain agnostic protocols, blockchain of blockchains, and blockchain migrators.

Table 1 summarizes the closely related surveys/reviews on blockchain interoperability and reveals our survey's novelty. The aforementioned surveys and review work either focus on discussing the architectural aspects [40]–[52] or failed to present a broad image of smart contracts in blockchain interoperability. For instance, some surveys only discussed interoperability in permission-less blockchains and some are discussing permissioned blockchains; on the other side, some are reviewing only homogeneous blockchains and others are discussing heterogeneous blockchains. Furthermore, none of the existing surveys/reviews focused on the role of smart contracts in blockchain interoperability. Our survey work is therefore intrinsically different due to its broad view, along with several features that are considered simultaneously. This study presents a detailed overview of solutions for interoperable blockchains by discussing the role of smart contracts in blockchain interoperability. To the best of our knowledge, this is the first study on the role of smart contracts in blockchain interoperability. Moreover, a classification of interoperable blockchain solutions is provided on the basis of blockchain type and programming languages used to develop smart contracts used in the interoperable blockchain solutions. A hierarchical taxonomy diagram is created in order to help readers to explore the field of blockchain interoperability. Eventually, based on our comprehensive survey, this study outlines various issues that still remain to be tackled and research opportunities for the future.

III. SMART CONTRACTS AND PROGRAMMING LANGUAGES

A smart contract is defined as a computer program that enforces the promises agreed by the interacting parties in the absence of trusted intermediaries. Though the concept of smart contracts is relatively new, the idea was first introduced by Szabo [32] in 1990. Back then, due to the unavailability of technological requirements and decentralized network protocols, the concept was only limited to theory. With the development of the Ethereum ecosystem, the smart contract becomes the key player to shape blockchains from a distributed ledgers to programmable state machines by introducing the execution of Decentralized Applications (dApps). The distinguishing features that make smart contracts pertinent to many applications are built-in transparency and immutability. Like all other transactions, smart contracts are stored in blockchains. Moreover, novel user requirements can easily be implemented by deploying smart contracts. However, smart contracts maintenance is different as compared to traditional computer software because they cannot be altered once deployed, even by the creator of the smart contracts [33].

Smart contracts can be developed in a number of programming languages. The most popular language for developing smart contracts is Solidity [34]. Solidity is an object-oriented, Turing complete language developed by the Ethereum platform to execute smart contracts on the Ethereum Virtual Machine (EVM). Solidity smart contracts can also be executed on Hyperledger Fabric [5] chains. Similarly, Rootstock (RSK) [19] is a sidechain to Bitcoin and compatible with Ethereum. General programming languages can also be used to design smart contracts for some blockchains e.g., Hyperledger Fabric, Neo [35], Eos [36] and Tendermint [37] supports smart contracts designed in Go, Java, NodeJS, Python, and C++. Steller uses Javascript, Golang, PHP, and python, however, the smart contracts of Steller are not Turing complete. An overview of the programming languages and smart contracts application development platform is given in [38].

IV. INTEROPERABILITY

Generally, interoperability in computer science refers to ''the ability of computer systems or software to exchange and make use of information''. National Institute of Standards and Technology (NIST) defined interoperability between blockchains as: ''an interoperable blockchain architecture is a composition of distinguishable blockchain systems, each representing a unique distributed data ledger, where atomic transaction execution may span multiple heterogeneous blockchain systems, and where data recorded in one blockchain is reachable, verifiable and referable by another possibly foreign transaction in a semantically compatible manner '' [39]. As blockchain is essentially a data structure, in which transactions/records from various entities are stored using the cryptographic mechanism with decentralized consensus mechanism by means of a smart contract. This study defined blockchain interoperability as follows: ''The ability of a distributed ledger to process transactions originated in another distributed ledger with homogeneous/ heterogeneous identity management, cryptographic management, consensus mechanism, and smart contracts capabilities''. Based on the aforementioned definition, blockchain interoperability is classified into four main areas.

A. IDENTITY MANAGEMENT

It is likely that each blockchain has its own identity management. However, when two or more blockchains are interconnected. It is necessary to uniquely identify each user, transaction or process in a cross-chain transaction while reading and writing transactions or data from one blockchain to another in order to ensure accountability.

B. CRYPTOGRAPHIC MANAGEMENT

In the case of cross-chain transactions or communication, each blockchain will have different cryptographic/hash methods. Though intermediary chains facilitating cross-chain transactions are tasked to resolve the cartographic hash of each blockchain. For true interoperability, a cryptographic management system must be developed to enable blockchains to add users/transactions from other blockchains dynamically. Such a cryptographic management system can grant access to specific users temporarily.

C. CONSENSUS MECHANISMS

Every blockchain has its own consensus algorithm. For interoperability, a case needs to be determined with an efficient

algorithmic scheme such that either one consensus algorithm can be scaled upon all the participating networks or a compatible mechanism for all the interacting blockchains can be applied.

D. CODE LEVEL INTEROPERABILITY

Chain code/smart contracts are written in different languages. For true interoperability, a contract written in one language can be scaled to other blockchain networks by referencing their particular contract code. For this purpose, a virtualization-based approach can be adopted. The virtualization-based approach enables the execution of a smart contract on multiple and heterogeneous blockchain platforms by creating an abstraction layer over the underlying blockchain. Furthermore, a user interface-based engine can be developed to create smart contract workflows between blockchains. This study focuses on code-level interoperability between blockchains.

V. ROLE OF SMART CONTRACTS IN BLOCKCHAIN INTEROPERABILITY

Today, the blockchain interoperability domain is very rich. However, the number of peer-reviewed articles in this area is limited. Therefore, self-published articles available on public platforms such as arXiv and ResearchGate are also included in this study. Moreover, the publicly available whitepapers of the renowned interoperability solutions such as Polkadot, Cosmos, Block Collider, and ICON republic, etc. are also surveyed in this study. The search process for the articles was conducted using google scholar. It is worthy to mention that this survey does not cover all the research articles in the blockchain interoperability area. This study is focused to investigate the role of smart contracts in blockchain interoperability solutions. Therefore, in this study only those research articles were surveyed in which smart contract is the main facilitator or key player in blockchain interoperability.

Although smart contracts have the potential to enhance blockchain interoperability, no attention has been paid to investigate the role of smart contracts in the interoperability domain. One of the key advantages of achieving interoperability in blockchains using smart contracts is that any peer in the network can deploy a smart contract. Similarly, any peer that agrees to the contractual clause of an existing smart contract is also capable to execute it. As a result, a smart contract can be reused. Smart contracts can call or invoke other smart contracts using calls to fulfill a certain task, or inquire about a past event. Typically, a smart contract can only be invoked from the local chain. However, recent advancements show that smart contracts on remote blockchains can also be invoked by passing arbitrary data or machine level byte code in the form of a transaction or Remote Procedure Call (RPC).

In order to give the readers an intuitive view of our classification of the interoperability solutions in accordance with the functionalities of the smart contracts, a high-level taxonomy of the interoperability solutions is provided in Figure 1. Creating a taxonomy implies grouping and classifying existing interoperability solutions into a compact representation, allowing the exploration and comparison of different solution designs. Similarly, it is difficult to evaluate if a taxonomy is good, especially if the domain is emerging rapidly. It is worth mentioning that there is not a hard rule in creating the proposed taxonomy, as the interoperability solutions vary on the basis of requirements and applications. The proposed taxonomy is based on the existing work found in the literature. It simply aims to set the foundation of achieving interoperability using smart contracts.

An overview of all the interoperability solutions is given in table 2. In table 2 privacy and security shows if (or not) the authors considered any mechanism to ensure privacy and security. Scalability shows if (or not) the proposed interoperability solution is scalable. Scalability in interoperable blockchains can be vertical scalability or horizontal scalability. We considered both the types and a detailed overview of each type is given in the discussion. The degree of confidence shows if (or not) the proposed solution considered a mechanism to ensure that the transaction is added to the block by following the longest chain. It refers to feedback mechanism that a target blockchain used to inform the source blockchain. Feedback is usually given by waiting till a number of blocks are confirmed on the main chain. It is used to avoid forks in blockchain. Bidirectional transactions represent if (or not) the interacting chains are able to send and receive transactions from each other. The interacting blockchain shows the details of blockchains for which the interoperability solution is achieved. Deployment mode shows the key player (or mechanism) through which interoperability is achieved between the interacting chains. Lastly, applications show the scenarios in which the solution can be applied. Throughout this paper, chain code (in Hyperledger blockchains) and smart contracts are used interchangeably. Source chain represents a blockchain where the transaction or smart contract is initiated whereas target chain represents the blockchain where the transaction terminates.

A. HETEROGENEOUS BLOCKCHAINS AND HOMOGENEOUS SMART CONTRACTS

This subsection discusses interoperability solutions between heterogeneous blockchains. Heterogeneous blockchains refer to different blockchains e.g., Bitcoin, Ethereum and Hyperledger, etc.). Whereas, homogeneous smart contracts refer to smart contracts that are developed in the same programming language e.g., Solidity, etc.

1) TIME LOCK CONTRACTS

Herlihy [53] proposed an atomic swap protocol for assets exchange across multiple blockchains. This protocol guarantees that if all interacting parties agree to the protocol, then the exchange of assets occurs. If any of the parties deviates from the protocol, then none of the parties incurs losses. Furthermore, there is no incentive for deviation from the protocol. The protocol can be seen as a directed graph

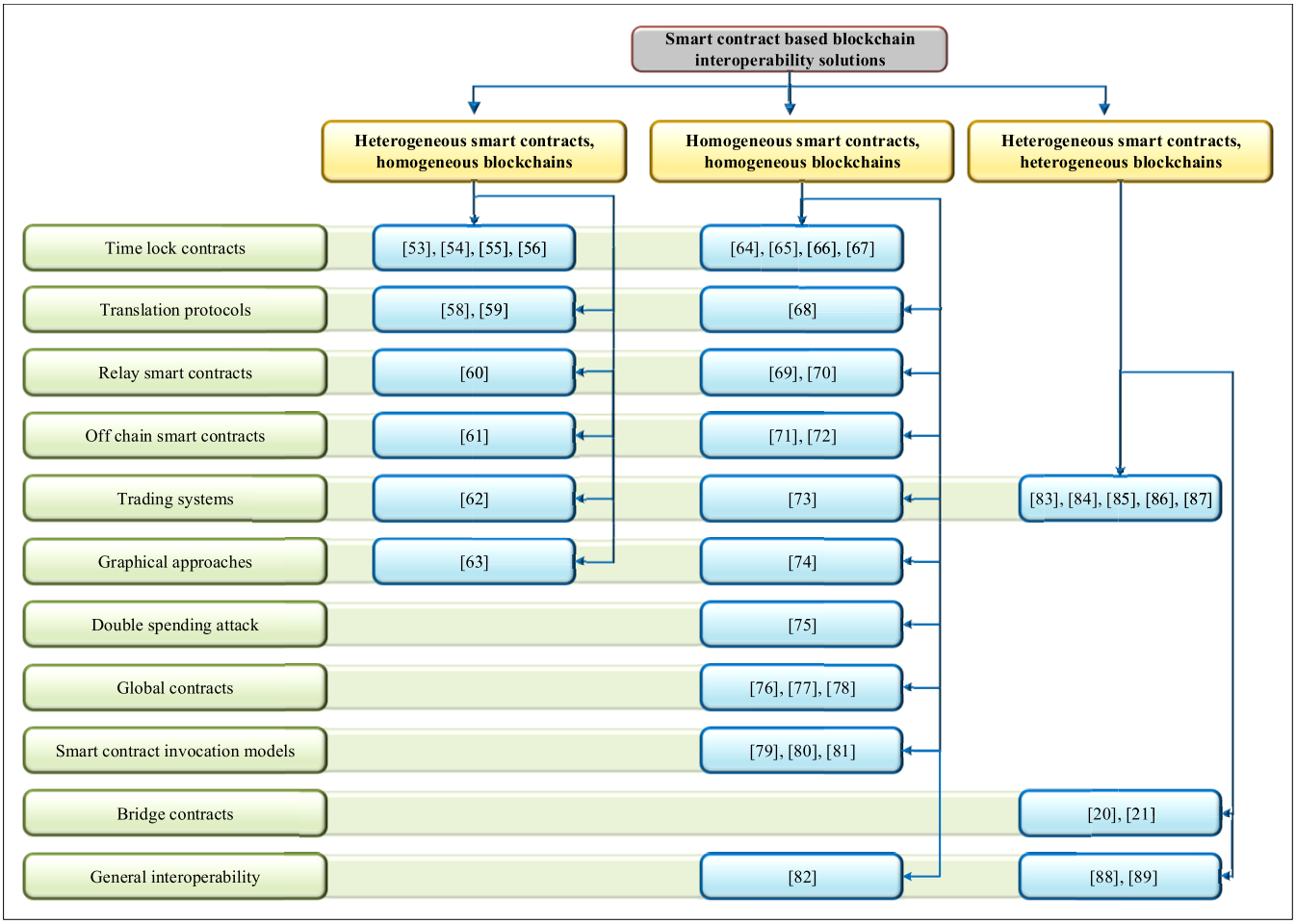


FIGURE 1. Taxonomy of smart contract based interoperability solutions in blockchains.

with a finite set of vertices and an ordered pair of arcs. The vertices represent the interacting parties, and the arcs represent the exchange of assets. Atomic swap protocol uses a Hashed Time Lock Contract (HTLC) to exchange assets in order to assume the control of the other party assets temporarily. A typical HTLC comprises hashed locks and keys. The ownership of the assets changes upon receiving a matching key before a certain time limit. However, if the matching key is not received under the time limits, then the assets are retained by the original owner. In this work, the presence of a market-clearing service is assumed to enable communication between the interacting parties to ensure consistency. The protocol occurs in two phases. In the first phase, the protocol is propagated in a leader-follower manner. In the second phase, the interacting parties propagate the secrets via hashkeys. The propagation terminates if the time limit expires or all the secrets are unlocked.

Fynn *et al.* [54] proposed atomic move. In this work, the smart contract from the source blockchain migrates to the target blockchain. Migration occurs in two steps. In the first step, the state of the smart contract is locked on the source chain. In the second step, the smart contract is recreated in the target blockchain to execute transactions. When the smart contract is locked on the source or target chain, only read operations can be performed on that chain. It is assumed that the source and target blockchains have the same virtual machine to execute the smart contracts. Moreover, there exists a procedure by which the source and target chain can verify the state variables (such as Merkle-tree) of each other.

Black *et al.* [55] proposed atomic loans. Atomic loans can be implemented as an extension of atomic swaps. The process is divided into four periods: loan, bidding, seizure, and refund protocols. In this work, it is assumed that two users residing on different chains communicate by means of a communication protocol. When the users agree to the terms of the loan such as the interest rate, repayment of collateral, and liquidate the collateral (in the case of default), the loan is issued and the terms are incorporated in a smart contract. The authors highlighted a number of use cases to ensure neither lender nor the borrower incurs losses. The loan process requires blockchains with smart contract capabilities, therefore this work is not compatible with some blockchains such as bitcoin due to the limited functionality of scripting language. Atomic loan is open-source.[3]

[3]https://github.com/AtomicLoans;

TABLE 2. Overview of interoperability solutions. Note: 1) P&S: Privacy and security, SC: Scalability, DoC: Degree of confidence, and BT: Bidirectional transactions.

Model & Ref.	P&S	SC	DoC	BT	Interacting blockchains	Deployment mode	Applications
Atomic swap [53]	✗	✗	✗	✓	Applicable to all smart contract supported chains	Hashed locks and Keys are used to exchange tokens or assets	Cross blockchain assets token or services exchange, Network sharding.
Atomic move [54]	✗	✗	✗	✗	Ethereum and Hyperledger Burrow	Same virtual machine is used on the interacting blockchains	Network sharding
Atomic loan [55]	✗	✗	✗	✓	Bitcoin and Ethereum	Terms are negotiated between the interacting users whereas hashed keys and locks are used to issue loan	Landing and borrowing of tokens or assets
XCLAIM [56]	✗	✗	✗	✗	Ethereum ecosystem	Intermediary vaults are incorporated to back funds	Multi party atomic swaps, temporary transaction offloading from main chain to sidechains
SCIP [58] [59]	✗	✗	✓	✗	Ethereum and Hyperledger Fabric	Implemented as a gateway on the interacting blockchains	It can be implemented as a layer on top of any blockchain
BTC Relay [60]	✗	✗	✗	✗	Bitcoin and Ethereum	As a smart contract	Enable users to pay Bitcoins in order to use Ethereum
COMIT [61]	✗	✗	✗	✓	Bitcoin and Ethereum	Liquidity provider nodes have accounts on interacting blockchains	Prevent double spending attacks
Fusion [62]	✓	✗	✗	✓	Bitcoin and Ethereum	Act as intermediary between the interacting blockchains	Key sharding and distributed storage
XChain [63]	✗	✗	✗	✗	Ethereum ecosystem	Hashed locks and keys to exchange tokens or assets	Can be used as market clearing service
Dai et al. [64]	✗	✗	✗	✗	Ethereum ecosystem	Act as intermediary between the interacting blockchains	Cross blockchain payment systems.
Sigwart et al. [65]	✗	✗	✓	✓	Ethereum ecosystem	Blockchains are interconnected with each other	SPV, Cross blockchain payment systems
Abebe et al. [68]	✗	✗	✗	✓	Hyperledger Fabric ecosystem	A relay service on each blockchain enable data sharing	To invoke smart contract on remote chain
Peace Relay [69]	✗	✗	✗	✗	Ethereum ecosystem	As a smart contract	Enables cross blockchain trading and communication

TABLE 2. ***(Continued.)*** **Overview of interoperability solutions. Note: 1) P&S: Privacy and security, SC: Scalability, DoC: Degree of confidence, and BT: Bidirectional transactions.**

POA [70]	✗	✗	✓	✓	Ethereum ecosystem	A number of fore-chosen validators	Community currency exchange
NOCUST [71]	✓	✓	✓	✓	Ethereum ecosystem	Operators act as intermediaries on the interacting blockchains	Off chain instant payments between the interacting parties
Garou [72]	✓	✓	✓	✓	Ethereum ecosystem	Leader act as intermediary between the interacting chains	Off chain instant payments between the interacting parties.
Wang et al. [73]	✗	✗	✓	✗	Ethereum ecosystem	Direct trading of assets or tokens	Multi chain trading between interacting users
CAPER [74]	✗	✓	✗	✓	Hyperledger Fabric ecosystem	A set of special nodes are used for cross blockchain transactions	Cross blockchain payment systems
Sai et al. [75]	✓	✗	✗	✗	Ethereum ecosystem	Trustee act as intermediaries between the interacting blockchains	Discouraging double spending attacks only
Robinson et al. [76]	✗	✓	✗	✓	Ethereum sidechains	The coordination blockchain coordinates all the transactions	Assets or token transfer between the interacting chains
STEM [77]	✗	✗	✗	✗	Ethereum (Rinkeby test net)	Exchange contract act as intermediary	Assets or token exchange
GPACT [78]	✗	✗	✗	✓	Ethereum ecosystem	Instances of control contract deployed on the interacting chains	General interoperability for Ethereum ecosystem
Nissl et al. [79]	✗	✗	✗	✓	Ethereum (Rinkeby test nets)	Intermediaries are used to forward calls between the interacting chains	Data and token transfer between the interacting blockchains
Pillai et al. [80]	✗	✗	✗	✓	Ethereum (Rinkeby,Ropsten and Kovan)	Cross chain transactions calls a smart contract directly	Messaging, smart contract invocation on remote blockchains.
RPCs [81]	✗	✗	✗	✓	Ethereum ecosystem	Off chain clients act as intermediaries between the interacting blockchains	Cross blockchain payment systems
Hyperservice [82]	✓	✗	✗	✓	Ethereum ecosystem	Programming framework	General interoperability for Ethereum ecosystem
Block Collider [83]	✗	✗	✗	✓	Bitcoin, Ethereum, Waves, Neo, Lisk and an anonymous [2] blockchain	Act as intermediary bridge between the blockchains	Cross blockchain payment systems
Hangyu et al. [85]	✗	✗	✗	✓	Bitcoin, Litecoin and Ethereum	Act as an intermediary between the interacting blockchains	Cross blockchain token transfer only

TABLE 2. *(Continued.)* **Overview of interoperability solutions. Note: 1) P&S: Privacy and security, SC: Scalability, DoC: Degree of confidence, and BT: Bidirectional transactions.**

ARK [86]	✗	✓	✗	✗	ARK, Bitcoin, Ethereum and Litecoin	Act as an intermediary between the interacting blockchains	Cross chain payments, data transfer
Wanchain [87]	✓	✗	✗	✓	Ethereum and Bitcoin	Act as intermediary bank between the interacting blockchains	Landing, borrowing, settlements and investment
ICON [88]	✗	✗	✗	✓	ICON and Ethereum	Act as an intermediary between the interacting blockchains	Cross chain payment systems
Overledger [89]	✓	✗	✓	✓	Ethereum, JPM Quorum, Hyperledger Fabric, XRP ledger, R3 Corda, Bitcoin, EOS	Operating system and layered approach	General interoperability solution
Cosmos [20]	✓	✓	✓	✓	Zones and proprietary chains	Hub act as intermediary between zones, TokenBridge act as intermediary between hub and proprietary chains	A framework for developing interoperable blockchain applications
PolkaDot [21]	✓	✓	✓	✓	Parachains and proprietary chains	Relay chain act as intermediary between parachains, TokenBridge act as intermediary between hub and proprietary chains	General interoperability framework that enables interaction between chains regardless of the underlying governance mechanism

Zamyatin *et al.* [56] developed XCLAIM (cross-claim). XCLAIM is a cryptocurrency-backed approach for issuing tokens on Ethereum. In such approaches, assets or tokens on one blockchain are backed by cryptocurrencies on another blockchain. This work used Bitcoin-backed tokens on Ethereum. The primary objective of this work is to enforce the correct issue, transfer/swap, and redeem operations on the backing and issuing blockchains. In this work, all the operations are performed on a number of actors: requester, sender, receiver, redeemer, vault, and smart contract. Requester locks the cryptocurrencies on the backing blockchain. The sender is used to shift ownership from one user to another on the issuing blockchain. The receiver is assigned ownership of the tokens. Redeemer destroys the corresponding tokens on the backing blockchain. Vault is used to redeem requests of the issuing blockchains on the backing blockchain. The smart contract is used to issue as well as manage tokens on the issuing blockchains. This work incorporates proof of punishment approach to enforce the correct behavior of all the actors. Furthermore, collateral and publicly verifiable audit logs are maintained on the backing as well as issuing blockchains. In this work, P2PHK [57] transactions are used on the backing blockchain. XCLAIM currently supports Bitcoin-backed tokens on Ethereum. However, the authors claim that the backing and issuing of blockchains can be extended to other cryptocurrencies as well. The performance of this work is evaluated in terms of execution cost for issuing, transfer, and redeem protocols. This work claims to have outperformed HTLCs by minimizing the execution cost and the overall time. XCLAIM is open-source.[4]

2) TRANSLATION PROTOCOL

Falazi *et al.* [58] developed Smart Contract Invocation Protocol (SCIP). SCIP is an interface that enables the interaction

[4]https://github.com/crossclaim/xclaim-sol;

of smart contracts residing on different chains. The interface acts as an intermediary and contains methods, roles, messages, and data formats of heterogeneous blockchains. It is responsible to convert the data types and blockchain-specific formats of one chain to the data and specific formats of other chains. In this work, JSON schema is used to describe data types and formats into text-based JSON. For conversion, first, the abstract specification of data types and formats of the interacting blockchains are generated. Afterward, an encoding function generates the underlying inputs for the target blockchain. For this purpose, one-to-one mapping rules are defined for all the interacting blockchains and the interface is responsible to invoke the desired method. Moreover, special nodes (i.e., market nodes) are employed for decentralized order matching. This work also supports offline trading. SCIP is open-source[5] and a prototype is implemented in [59] for seafood supply chain using gateways.

3) RELAY SMART CONTRACT

BTC Relay [60] is a trustless Ethereum smart contract. It is mainly used to verify or arbitrarily pass on Bitcoin transactions. It can also be used to store or inspect Bitcoin block headers stored in a smart contract. Nodes that submit block headers to the BTC relay are known as Relayers. Relayers are rewarded for submitting block headers. BTC relay is open-source.[6]

4) OFF CHAIN SMART CONTRACTS

Hosp *et al.* [61] proposed a Cryptographically secure Off-chain Multi-asset Instant Transaction network (COMIT). COMIT allows instant transactions between blockchains using off-chain smart contracts. It aims to develop a communication protocol so that nodes can communicate with each other. COMIT network use payment channels and HTLCs for assets exchange. The network comprises three entities: users, businesses, and liquidity provider nodes. Users are entities who acquire COMIT network services. Businesses are the entities that upgrade their infrastructure to the COMIT network. Liquidity provider nodes are the entities that have accounts in multiple blockchains. They are market makers. Their job is to convert one blockchain asset to another by providing liquidity. The privacy-preserving protocols of COMIT are under development at the time of writing. COMIT is open-source.[7]

5) TRADING SYSTEMS

Fusion [62] is a crypto-financial platform where tokens from various blockchains as well as centralized exchanges can be traded using smart contracts. It aims to develop a bridge where tokens from various chains are mapped in multi-token smart contracts. Fusion use Distributed Control Rights Management (DCRM) to store the private keys of digital assets owned by individuals from the interacting chains or centralized exchanges. DCRM ensures that no individual or node can take control of all the private keys or digital assets via crypto-assets mapping. Its primary objective is to have a management layer that is capable of interacting with all the blockchains to overcome the inability of interaction or transfer of digital assets or tokens. Fusion is open-source.[8]

6) GRAPHICAL APPROACH

XChain [63] is a three-phase protocol designed for general cross-chain transactions. The protocol is implemented in a leader-follower manner. Leaders (i.e., feedback vertex set) are a special set of nodes in the system. They are responsible for the initiation of smart contracts. All other nodes are followers. In the first phase, smart contracts are created. In the second phase, leaders release their secrets for propagation. In the third step, the secret is relayed to the representative sources. The representative sources forward the collected secret to the nodes who are responsible for propagating the secrets throughout the network. This protocol ensures that all the parties in a trade do not deviate. For this purpose, the leaders wait to receive an incoming smart contract before releasing the secrets for propagation. It is assumed that transactions are assembled by market clearing services. This work uses HTLCs for each party in the cross-chain transactions.

7) DISCUSSION

Interoperability solution for a limited number of chains or designated chains will only increase the number of isolated chains rather than interconnecting chains. Interoperability between independent chains can be accomplished using a plug-in manner with generalized multi-chain communication and cross-chain transaction protocols. A key advantage of [53] is that it can be extended to any model for assets or service exchange. However, the model is vulnerable to distributed denial of service attacks in the event if an adversary repeatedly fails to complete the protocol. This results in locking the assets of the interacting party and making him/her unavailable to trade assets with other parties.

Ever since its emergence, the widespread adoption of blockchain is confronted with scalability. Interoperability has a huge impact on blockchain scalability. Scalability in blockchains refers to two scenarios: i) higher transaction processing capabilities and ii) interconnecting more and more blockchain networks. The former is usually referred to as vertical scalability whereas the latter is called horizontal scalability. BTC relay [60] is not a scalable solution due to the underlying consensus mechanism in Bitcoin blockchain. Moreover, the solution is only applicable to Ethereum. In [54], smart contracts move according to client's request. As a result, smart contracts may end up in a repeated back-and-forth cycle. To prevent this, once a smart contract is moved from one blockchain to another blockchain. It is not allowed to move to another blockchain for three days. In such

[5]https://github.com/ghareeb-falazi;
[6]https://github.com/ethereum/btcrelay;
[7]https://github.com/comit-netwok

[8]https://github.com/FUSIONFoundation

a case, the clients from other blockchains repeatedly try their transactions, and if the transaction is not successful, then the client has to wait randomly.

Blockchains were created to eliminate centralized third parties with the distributed consensus mechanism such as proof of work or proof of stake, etc. It is noteworthy that decentralization is the key to success in the case of interoperable blockchain networks. However, the current interoperable solutions are moving towards centrality e.g., a fixed number of validators in modified consensus mechanisms, etc. Other interoperability solutions that use vaults or any other attestation services are also vulnerable to a single point of failure. In [58], the gateway is responsible to formulate and sign transactions on behalf of the client applications. As a result, a bottleneck can occur at the gateway if the number of transactions increases. Moreover, the increase in the number of interacting blockchains may also degrade the performance of the system. As the gateway is responsible for translation using pre-defined rules.

In interoperable blockchain solutions, cross-chain transactions are charged from the users' accounts in order to reward the entities who partake in the process. In some cases, when the cross-chain value is low, the transaction may end up costly. Keeping in view the transaction fee and interest rates and market fluctuations, the loan process in [55] may end up very costly to the borrower.

Some existing interoperable solutions use distributed nodes in cross-chain transactions. To date, most of the existing work assumes honest behavior or ceasing collateral in the event of dishonest behavior. What will happen if a node or set of distributed nodes acts maliciously for a cross-chain transaction with a cross-chain value more than the collateral they submit? work in [56], relies on the performance of vaults. Similarly, [61] use liquidity provider nodes, whereas, [62] use distributed nodes to hold private keys.

In [63], if the graph is strongly connected, then the secrets can be easily propagated throughout the network. However, in the case of a weakly connected graph where every vertex is not reachable by all other vertices, a follow-up mechanism is required. Moreover, the timeout mechanism in an HTLC must be carefully designed, as leaders may not be directly reachable to the parties.

B. HOMOGENEOUS BLOCKCHAINS AND HOMOGENEOUS SMART CONTRACTS

This subsection discusses interoperability solutions between homogeneous blockchains for which smart contracts are either developed in the same programming language or a similar execution/virtual environment is considered to execute smart contracts. Homogeneous blockchains refer to similar blockchains e.g., Ethereum and Ethereum classic, etc.

1) TIME LOCK CONTRACTS

Dai *et al.* [64] proposed a cross-chain transaction model. This model comprises three roles: individual blockchains, users, and the cross-chain system. It is assumed that users on the interacting blockchains have their own addresses for cross-chain transactions. Moreover, the cross-chain system is responsible for coordinating transactions, locking or unlocking transactions, and providing notary nodes to monitor transactions of the interacting system. A cross-chain transaction between the interacting chains occurs in three phases. First, assets on the interacting chains are locked using a multi-signature address provided by the cross-chain system. Secondly, keys are negotiated between the interacting parties on each chain using a notary from the cross-chain system. Keys are negotiated using the Diffie Hellman algorithm. Lastly, the transactions are processed and coordinated by the notaries. Assets are swapped using time-out smart contracts to ensure atomicity. Same smart contracts are used on the interacting chains.

Sigwart *et al.* [65] discussed the verification process of cross-chain transactions. In this work, clients constantly pass block headers to the destination chain. To keep the clients motivated for their participation in the submission process, an incentive structure is proposed in [66]. When a new header is submitted for verification, this work store the information to track the branches of the main chain and discard the remaining information. To verify a specific transaction, clients request the destination chain about a specific transaction. A smart contract is used to extract the logged information of the block headers from the publicly available transnational history. Furthermore, the hash of the aforementioned information is calculated to verify a specific transaction as valid or disputed. The steps used in cross-chain transactions verification process for [65] are outlined in [67]. These steps are as follows: first, a specific number of tokens on a chain can only be created if the source chain guarantees that the same amount of token has been burned. Secondly, the burning process cannot be faked. Third, for every burned token on a source chain, the corresponding tokens can only be created once on the destination chain. Lastly, it is not possible to burn tokens on one chain unless the same amount of tokens are recreated on the target chain. In this work, the authors also discussed an optimization technique to minimize data storage for cross-chain transactions.

2) TRANSLATION PROTOCOL

Abebe *et al.* [68] proposed architecture for data sharing between interacting blockchains. The proposed architecture comprises two autonomous chains. Relay service is incorporated on each chain to enable data sharing along with verifiable proof. It is assumed that each blockchain is capable to accept as well as verify data from the other chain. For verification of the data, a policy is provided by the source chain. Moreover, the target chain is capable to provide verifiable proof in accordance with the source chain demands. The relay service on each blockchain communicates with each other using a neutral protocol. Furthermore, the relay service is capable of translating the network protocol messages into the underlying network implementation using a set of plug-in network drivers. A set of special system contracts are

incorporated for data sharing as well as enforcing network rules. The governing bodies of the interacting chains are responsible to initialize the metadata of the system contracts to enable interoperability.

3) RELAY SMART CONTRACTS

Peace relay [69] is a smart contract that enables Ethereum blockchain to interact and communicate with other Ethereum chains such as Ethereum classic. Using the Peace relay, transactions can be read as well as verified on Ethereum classic and vice versa. Furthermore, it can also be used to verify account balances. Peace relay is open-source.[9]

Proof of Authority (PoA) [70] is an autonomous network that incorporates EVM-based blockchains. This project aims to enhance the interoperability and transparency of the blockchain ecosystem with the PoA consensus. It enables cross-chain data as well as assets transfer using the PoA TokenBridge. The main components of a TokenBridge are Bridge Monitor (BM), user interface, Bridge Deployment Playbooks BDP smart contracts (PDPSCs), and Arbitrary Message Bridge (AMB). BM is used to check balance and unprocessed events on the TokenBridge. BDPSCs keep a record of the configurations as well as deployment of the remote chains. The user interface enables cross-chain token transfer. AMB enables data exchange between EVM-based chains. The data exchanged using POA can be used to transfer tokens, invoking a cross-chain smart contract, disseminate token exchange rate to the target blockchain, and synchronize smart contract states of the interacting chains. This work depends on the correct behavior of validators. Therefore, a number of fore-chosen validators are incorporated to partake in the consensus mechanism. POA is open-source.[10]

4) OFF CHAIN SMART CONTRACTS

Khalil *et al.* [71] developed NOCUST. It uses a challenge-response mechanism for off-chain payments. The primary objective of NOCUST is to improve throughput by processing transactions on the sidechain without publishing on the main chain. To process transactions on the sidechain, users must on account on the main chain as well as a sidechain. Exchange occurs through an operator who acts as an intermediary between the main and sidechain. Users communicate with the off-chain operator to register and send tokens to the recipient on the sidechain. In this work, after a pre-defined time, checkpoints are incorporated to update the states of all users on the main chain. Moreover, users can evaluate the behavior of the operator by publishing challenges using smart contracts. If the behavior of the operator is malicious, users can penalize the operator and triggers the recovery mechanism on the chain. This enables the user to recover payment from the last validated checkpoint. The performance of this work is evaluated in terms of gas consumption and latency. NOCUST is open-source.[11]

Ye and Wu [72] presented Garou. Garou is an off-chain token transfer protocol. It uses an election process to elect a leader among the participants to execute off-chain transactions. A new leader is elected at the beginning of each epoch. The leader is elected on the basis of an arbitrary hash function and the initial balance of the participants at the beginning of each epoch. The leader is responsible to keep a track of the initial balance and the total balance sent or requested by the participants during an epoch. Moreover, the leader is also responsible ensure that all participants partake in a consensus mechanism regarding the initial state of the next epoch and answer challenges posted by the participants in the event of a dispute. To ensure that participants didn't lose their funds, all disputes are resolved using the on-chain contract. The performance of this work is evaluated in terms of throughput and latency.

5) TRADING SYSTEM

Wang *et al.* [73] proposed a cross-chain trading model for joint operations of multi micro-grids. In this work, two blockchains are incorporated with built-in P2P trading networks. These blockchains represent independent micro-grids that trade power to an external network in the event of unbalanced electric power. Cross-chain trading occurs in six steps. First, the source chain identifies the requirements for trading by writing the description and deadline in a smart contract. In the second step, the cross-chain trading requirements are verified in the source chain using the local consensus mechanism. If the verification process succeeds, a smart contract is deployed, and a request is forwarded to the target chain. Third, the request is verified by the target chain. In case of successful verification, a smart contract is built and broadcast to all the nodes. If the verification at the target chain fails, the request is ignored. Afterward, the target chain has to conclude the trading deal and send a response to the source chain. In the next step, the source chain extracts the response from the received message of the target chain and returns the transaction keys upon the execution of the smart contracts. Lastly, the source and target chain broadcast their respective certificate to the multi microgrid system. A key management interoperable protocol is used for communication between the interacting chains. This protocol uses the RSA algorithm based on the Chinese reminder theorem. Moreover, a special set of nodes in the system validates cross-chain communication.

6) GRAPHICAL APPROACH

Amiri *et al.* [74] proposed an asynchronous blockchain system to support a set of distributed applications in a trustless manner. The applications on the interacting blockchain maintain two sets of records, namely public and private records. These records are stored in a single data store. Private records

[9] https://github.com/KyberNetwork/peace-relay;
[10] https://github.com/poanetwork/;
[11] https://github.com/liquidity-network/nocust-contracts-solidity;

can be accessed and edited by the application itself whereas, public records are visible as well as maintained by all the applications. Similarly, each application maintains two types of smart contracts, namely private and public smart contracts. Private smart contracts are used to implement internal transactions whereas public smart contracts are used to implement cross-chain transactions using a service layer agreement. Moreover, the public smart contract can be executed on all the applications to enforce the terms of cross-chain transactions. The system comprises a special set of nodes called agents and orderer nodes. Agent nodes are used to execute applications, whereas orderer nodes are devoted to order cross-chain transactions globally. In this work, the ledger is neither maintained by the nodes nor applications. Instead, each application is responsible to maintain a local view of the ledger in the form of a directed acyclic graph. The blockchain system is implemented in an execute-order-validate manner.

7) DOUBLE SPENDING ATTACK

Sai *et al.* [75] proposed a scheme to discourage attackers who try to double-spend a transaction in cross-chains transactions. It is assumed that a number of trustees exist in the network. These trustees are responsible for cross-chain transactions and hold sufficient funds on the interoperable chains. Observers are employed in the network to endorse transactions. A trustee only processes the transactions that are endorsed by the observers. To discourage a double-spend attack, observers and trustees initiate a smart contract. All the observers submit their responses to the trustee. The decision is made on the basis of the majority of the responses submitted to the trustee. If the outcome of an observer does not match the majority of the responses, the observer is considered malicious and penalized for his dishonest response. Whereas all the other observers are rewarded. A game-theoretical analysis is performed to demonstrate that the observers do not collude in a double-spending attack.

8) GLOBAL CONTRACTS

Robinson *et al.* [76] developed a protocol for cross-chain transactions in Ethereum based sidechains. In this work, a cross-chain transaction comprises an originating transaction and/or a subordinate transaction and subordinate view. The originating transaction is Ethereum transactions that originate in Ethereum blockchains, whereas subordinate transactions or subordinate views are the transactions that result due to originating transactions on sidechains. Every originating transaction may or may not have a subordinate transaction or subordinate views. The cross-chain transaction protocol is based on multi-chain nodes, coordination blockchain, and cross-chain coordination contracts. Multi-chain nodes represent a group of more than one node residing on different chains. These nodes collaborate to enable cross-chain transactions. Moreover, these nodes have validator nodes on each sidechain to validate transactions. A coordination blockchain can be any Ethereum blockchain such as Ethereum Mainnet, etc. It has access to all the sidechains and coordinates the cross-chain transactions. A cross-chain coordination contract is deployed on the coordination blockchain. It enables the sidechain to commit or discard the updates related to a cross-chain transaction. A key feature of the coordination contract is the Transaction Timeout Block Number (TTBN). TTBN is assigned by the coordination contract for all the cross-chain transactions. If a cross-chain transaction does not publish a commit message to the coordination blockchain before TTBN is assigned. The corresponding transactions are time-out. When the coordinating chain receives a cross-chain transaction request, it has to check the status of the coordination contract. The contract can either be in a locked or unlocked state. If the contract is unlocked, the transaction request is processed. If the contract is locked, then the transaction is ignored. A cross-chain transaction state represents the state update of a transaction between commit or ignores state in the coordination contract.

Darisi *et al.* [77] proposed a token exchange mechanism using a global exchange contract between the interacting chains. The mechanism comprises two actors: token traders and token owners. Token traders are user accounts who trade tokens with the counterparty, whereas token owners are user accounts who supply new tokens by deploying smart contracts. In this work, the authors used two ways to exchange tokens. In the first method, the interacting parties register their tokens on the central exchange contract. After registration, any of the interacting parties can initiate the exchange. Once the exchange is completed, the interacting parties un-register their tokens from the contract. In the second method, the authors used oracles and atomic swaps for token exchange. In this method, the central exchange contract is also used to exchange trade price and hash secrets.

Robinson *et al.* [78] proposed General Purpose Atomic cross Chain Transactions (GPACT) for Ethereum based blockchains. It is primarily used in the scenario when an application needs to invoke a smart contract residing on multiple blockchains. In this work, a cross-blockchain control contract with instances deployed on all the interacting chains is used to manage function calls. All the events emerging from the control blockchain are trusted on the interacting chains. First, an application fetches the state of the smart contract from a blockchain to determine the parameter value of the remote functions. Afterward, a simulation of the contract code is executed. The protocol starts with an application calling the root blockchain. The root blockchain has a start function that contains the entry points to the call graph. It registers the account that performs cross-chain transactions of the interacting blockchains. Moreover, it registers other parameters such as expected function values and cross-chain transaction identifiers, etc. These parameters are included in a start event and passed to the segment function. The segment function is used to invoke a smart contract as part of the cross-chain transaction. If the segment function executes successfully, it returns a segment event containing a list of lock contracts. The start event and signed segment event are passed as parameters to the root functions. A root function indicates

that all updates on the blockchain should either be committed or discarded. Updates are discarded if an error message is received in the segment event. A root function publishes a root event upon successful execution. The root event along with the segment event are passed as input parameters to the signaling function. The signaling function is called if an update needs to be committed or ignored on any of the interacting blockchains. A signaling event is published upon the successful execution of the signaling function, indicating that smart contracts are unlocked. The performance of this work is evaluated in terms of gas consumption and latency. GPACT is open-source.[12]

9) SMART CONTRACT INVOCATION MODELS

Nissl *et al.* [79] proposed a framework to invoke smart contracts on one blockchain from other blockchains. It is assumed that users or smart contracts deployed on the source as well as target blockchain are capable of sending and receiving a response from each other. A distributor and invocation smart contract is deployed on the source and target chains, respectively. Intermediaries (who are not part of the interacting blockchains) are incorporated to forward calls between the interacting chains. The proposed framework comprises six phases: register, offer, execution, forwarding, verification, and finalization phase. In the registration phase, the metadata of the aforementioned entities is saved by the caller of the distribution contract. In the offer phase, an event is announced on the source chain about the beginning of the offer phase. Afterward, intermediaries with accounts on the interacting chains are used to broke offers to the distribution contract. In the execution phase, the intermediaries execute the process of smart contract invocation on the target chain. To invoke a remote contract from a local chain, the desired method along with its parameters to activate the smart contract (e.g., the start gas, gas price, and the number of tokens) are transferred to the target blockchain. In the forwarding phase, the intermediaries selected are used to forward calls between the interacting chains. Whereas, in the verification phase, validators compare the data stored on the source chain and target chain to ensure that the intermediaries have not published incorrect data. Lastly, in the finalization phase, the call to invoke a remote contract terminates by distributing the reward among validators. A voting mechanism is used to distribute the transaction reward or reimburse the transaction cost. The performance of this work is evaluated in terms of gas consumption. This framework is open source.[13]

Pillai *et al.* [80] proposed a cross-chain communication model using transactions. In this work, a transaction refers to a call that invokes a remote smart contract to perform a certain task. The proposed model comprises two stages. In the first stage, information is retrieved by requesting blocks from clients. In the second stage, the state of the blockchain is updated using transactions, validation, and verification process. It is assumed that the interacting blockchains trust each other to a certain level. Moreover, the participating chains are capable of processing cross-chain transactions if the counterpart presents valid proof. The performance of this work is evaluated in terms of latency.

Sigwart *et al.* [81] proposed RPCs invoke smart contracts in target blockchain. RPCs are implemented in a request-response paradigm. RPCProxy and RPCServer smart contracts are deployed to initiate and verify the call requests between the source and target chain. Off-chain clients residing on a different chain are used to forward the call requests between the interacting chains. This model is open source.[14]

10) GENERAL INTEROPERABILITY

Liu *et al.* [82] developed a Hyperservice that is a programming platform. It enables the development and execution of cross-chain applications. The core components of Hyperservice are dApp clients, Verifiable Execution System (VESys), Network Status blockchain (NSB), and Insurance Smart Contracts (ISCs). DApp clients act as a gateway for the dApps to interact with the Hyperservice platform. VESys is used to compile and execute high-level programs in the dApps. NSB provides the execution status of the dApps whereas, ISCs ensure accountability. ISCs provide atomicity in a financial transaction. Moreover, in the event of misbehavior from the interacting parties, ISC is capable to revert all the financial transactions. To enable the execution of dApps across different chains, Hyperservice uses a Unified State Model (USM). USM unifies the interacting chains by providing a virtualization layer. This enables the dApps to execute on any chain regardless of the underlying blockchains implementations such as smart contracts, execution environment, and consensus mechanism, etc. The proposed USM is developed using Hyperservice programming Language (HSL). HSL is a proprietary programming language developed by Hyperservice platforms for writing cross-chain dApps. It also supports smart contracts written in different languages. For this purpose, a special program is written in HSL to abstract smart contracts as interoperable entities. This work is evaluated in terms of end-to-end execution latency and throughput. Hyperservice is open-source.[15]

11) DISCUSSION

There is a need for a reward and punishment management system for the validators or entities who assist cross-chain transactions. With a fixed number of validators, if a validator is punished by ceasing the funds only, the malicious validator is still present in the system. A proper mechanism is required to block such entities from participation in future transactions. The work in [75] relies on the responses submitted by observers. It is assumed that most of the responses submitted are true. In the event, if two out of three observers collude and initiate a secret smart contract with each other against

[12]https://github.com/ConsenSys/LTACFC;
[13]https://github.com/markusnissl/cross-chain-smartcontracts
[14]https://github.com/pantos-io/x-chain-smartcontracts
[15]https://github.com/HyperService-Consortium;

the third honest observer, then the honest observer can be penalized. Though the game theoretical analysis shows that it is in the best interest of the observer to remain honest to the network. Observers may deviate if the transferring funds are more than their deposited stakes. Similarly, [71] and [72] relies on the performance of operators and leaders. Though the behavior of the leaders and operators can be analyzed and checkpoints are incorporated to recover the system from a validated point. The malicious entities can still partake in the validation process. Similarly, [73] and POA [70] use a special set of validators. The work in [81] did not discuss the behavior of off-chain clients to invoke smart contracts.

The authors in [74] and [80] achieved promising results to minimize latency. In [74], the same set of nodes are used for local as well as cross-chain transactions. Therefore, increasing the number of cross-chain transactions does not improve performance. It is due to the reason that Fabric blockchain is only capable to process a certain number of transactions with a latency of 40ms. Similarly, [80] analyzed the performance of the proposed work with a mother blockchain acting as an intermediary to evaluate application latency. In practical situations, the latency may vary for each application.

The work in [76] and [77] use global smart contracts to enable interoperability. Similarly, in [64], the cross-chain system is responsible for coordinating transactions. A single chain or smart contract handling numerous chains or transactions are susceptible to a single point of failure. The work in [78] is evaluated to read and write integer values from one blockchain to another. Moreover, the performance of this work may be confronted with latency as the successful execution of the events depends on locking and unlocking smart contracts of the chains.

The work in [79] is evaluated for permissioned blockchains. This limits the applicability of the interoperability solution to public chains. Moreover, intermediaries and validators from various publish chains are unable to partake in the validation process. The current implementation of [82] is not fully atomic. Moreover, at this stage, the inter blockchain communication protocol does not inform the smart contract if an event terminates prematurely. The authors hope to ensure atomicity by using stateless smart contracts. However, this requires additional requirements such as decoupling the consensus layer and using a trusted execution environment.

C. HETEROGENEOUS BLOCKCHAINS AND HETEROGENEOUS SMART CONTRACTS

This subsection discusses interoperability solutions between heterogeneous blockchains (e.g., Ethereum, hyper-ledger, and Bitcoin, etc.). Heterogeneous smart contracts refer to the smart contracts that are developed in different programming languages e.g., interoperability between different blockchains using smart contracts developed in Solidity and Java, etc.

1) TRADING SYSTEMS

Block Collider [83] aims to develop a multi-chain trading platform by bridging heterogeneous blockchains. The promising feature of Block Collider is to enable the invocation of smart contracts residing on one chain from a remote chain and secure transactions in the absence of validators or third-party intermediaries. Block Collider unifies the recent most state of blocks from intermediary bridges into the Block Collider multi-chain. Multi-chain transactions are executed using incentive-based atomic swaps. A modified version of Nakamoto consensus, i.e., proof of distance is used to determine the next block. It is noteworthy that in Block Collider, the number of transactions in the blocks is not fixed. Borderless [84] is a user interface that interacts with Block Collider and enables the users to place as well as retrieve trade orders in a human-readable format. Block Collider is open-source.[16]

Tian *et al.* [85] used intermediaries on the Ethereum test net to exchange Bitcoins and Litcoins for Ethers. It is assumed that the intermediaries are capable to support as well as verify transactions on the interacting chains. Moreover, they are also part of the validation committee and forwards transactions between users of the interacting chains using smart contracts.

ARK [86] is a blockchain capable of generating novel blockchains on user demands. The ARK ecosystem comprises decentralized blockchains that are interoperable with each other using ARK SmartBridge. The SmartBridge enables the interacting blockchains to send data as well as assets to each other. ARK uses two types of communication protocols, namely Protocol-specific SmartBridge and Protocol-Agonostic SmartBridge. The former enables the communication between ARK main net and the blockchains that originate from the ARK core technology, whereas the latter enables communication and token transfer between heterogeneous blockchains such as Bitcoin and Ethereum, etc. To enable communication between the interacting chains, a special data section known as ''Vender filed'' is used. The ARK Contract Execution Service (ACES) is a community project that enables a two-way transfer of smart contracts between ARK and Ethereum network. Data or assets are exchanged using transactions. The data exchanged can be used for hashing or invoking a smart contract on the target chains. Using a set of encoded listeners, ACES can be applied to any chain to achieve interoperability. ARK is open-source.[17]

Lu *et al.* [87] developed Wanchain. Wanchain is a trading platform that enables interoperability by acting as a bridge between the interacting chains. In this work, first, the users from source and target blockchains have to register their token/assets on Wanchain. Afterward, users of the interacting blockchains are issued corresponding native coins (in the form of smart contracts) on Wanchain for trading. Transactions across the blockchains are accomplished using an asset template and locked accounts. Wanchain uses a modified proof of stake consensus algorithm with three types of nodes: vouchers, storemen, and validators. Vouchers are responsible

[16] https://github.com/blockcollider
[17] https://github.com/ArkEcosystem

for providing proof of transaction, whereas storemen are responsible for computing and merging the signature parts of the locked accounts. The smooth operations of Wanchain require storemen to stay online. The validators are responsible for validating the transactions on the Wanchain. However, there is a chance of collusion among these nodes if the gains of collusion exceed the participation cost. To preserve privacy in cross-chain transactions, Wanchain uses a one-time account with ring signatures to hide the identity of the smart contract initiator.

2) GENERAL INTEROPERABILITY

ICON republic [88] is a trading platform that aims to connect community-based independent blockchains. The core components of ICON republic are community, Community Node (C-Node), Community Representative (C-Rep), and citizen nodes. Various blockchains such as Bitcoin and Ethereum, etc. are considered as communities. C-Nodes are the building blocks of community. These nodes are responsible to determine the policies and consensus mechanism of each community. A C-Rep node represents their respective community inside the ICON republic ecosystem. A C-Rep node is elected on the basis of ICON incentive and an artificial intelligence-based scoring system. They act as validators in the ICON republic. Nodes that generate transactions are citizen nodes. Anyone can partake in the ICON republic as a citizen node by using the dApps available in the ecosystem. The underlying blockchain that connects different communities is called NEXUS. NEXUS is a multi-channel blockchain. It comprises light clients who represent their respective communities on the NEXUS chain. It acts as a decentralized exchange to transfer tokens between the interacting chains. A blockchain transmission protocol is used for cross-chain transactions. One of the key features of ICON republic is the availability of a number of use cases and pre-developed dApps for the participant nodes to use. Moreover, to accommodate the diverse need of various communities and users, ICON use loopchain. Loopchain is an enterprise blockchain with smart contract functionalities that can be customized according to the operational needs of the communities. The key feature of loopchain is the availability of a Smart Contract On Reliable Environment (SCORE). SCORE enables the deployment of smart contracts without a dedicated virtual machine. ICON Republic is open-source.[18]

Overledger [89] is a blockchain operating system capable of connecting heterogeneous as well as homogeneous blockchains. It is capable of performing operations on multiple blockchains simultaneously using multi-chain applications. These multi-chain applications are capable of executing smart contracts that are not dependent on a single blockchain. Coordination among multiple blockchains is achieved using a special set of nodes called connectors. A connector can be any entity or party with a minimum of one node on each of the interacting chains. These nodes are in-charge of communication as well as token and data transfers between the interacting chains. For communication, Overledger uses the two-phase commit protocol. Due to its layered approach, Overledger can be used on top of any blockchain. The layers in Overledger are the transaction layer, messaging layer, filtering & ordering layer, and application layer. The transaction layer is responsible for all the operations necessary to reach a consensus on the interacting chains. The messaging layer is used to retrieve transaction data or smart contracts. The filtering and ordering layer creates connections between various messages on the messaging layer. Moreover, it is also in-charge of validating the out-of-chain messages. The application layer updates the state of applications. It is noteworthy that, unlike other solutions, Overledger does not use a blockchain to enable cross-chain interactions. It uses a blockchain programming interface to enable interaction with the underlying blockchain. Overledger is open-source.[19]

3) BRIDGE CONTRACTS

Kwon and Buchman [20] developed Cosmos network. Cosmos links isolated blockchains (also known as zones). The primary zone in the cosmos network (i.e., the cosmos hub) is a multi-asset blockchain capable of extending the network to adapt to the advancements in novel blockchains. It allows instant transfer of tokens between zones securely. The exchange of tokens between zones does not require liquidity exchange between the interacting zones. However, token transfer among zones has to go through the cosmos hub. Moreover, the hub maintains a record of the tokens held by zones. Zones in cosmos networks communicate with each other by using an inter-blockchains communication protocol. This protocol enables zones to identify the number of transactions committed on the receiving chain. However, the sender and receiver zones must be able to keep up with the block header of one another.

All the zones in the cosmos network use Tendermint Byzantine fault tolerance algorithm. The algorithm is capable of processing thousands of transactions in a span of one or two seconds. A fixed number of validators validate the blocks with respect to their voting power. If any of the validators act maliciously, they are detected and penalized by the algorithm. As the cosmos network is a set of independent and isolated blockchains. It allows the zones to employ their own governance system. The zones do not have to employ the policies implemented in the cosmos hub. Similarly, the cosmos hub is not responsible for committing and executing transactions of the zones. Proprietary chains such as Bitcoin and Ethereum, etc. can also be connected to the Cosmos network by means of a dedicated zone known as Bridge-Zone. This requires the bridge validators to have a tindermint powered blockchain with a special application. A bridge-contract enables the transfer of tokens to (and from) proprietary chains and the Cosmos hub. Cosmos is open-source.[20]

[18] https://github.com/icon-project

[19] https://github.com/quantnetwork

[20] https://github.com/cosmos

Wood [21] developed Polkadot. Polkadot is designed to link parallel chains (also known as parachains) or data structures (not necessarily a blockchain). A relay chain administers all the transactions between the parachains. A registry is maintained in the relay chain to keep a record of the parachains. To add or remove a parachain from the registry, a referendum contract is placed. The network is maintained by four participants, namely collators, validators, fishermen, and nominators. Validators are selected using nominated proof of stake. They are allowed to nominate other nodes to validate blocks on their behalf. Nominators are stake-holding parties. Their function in the network is to place risk capital. Risk capital refers to the funds invested in high-risk rewards. Collators assist validators in bringing forth authentic blocks. Fishermen can be thought of as bounty hunters. Their job is to hunt bad actors who perform illegal actions and collect their reward. For cross-chain transactions, input and output queues are maintained at the parachains. Relay chain maintainers are responsible to carry transactions from the source chain output queue to the destination chain input queue.

A Cross-Chain Communication protocol (XCMP) [90] is proposed for the parachains to communicate. For two parachains to communicate, they are required to have a channel. Moreover, a pair can only have two channels. XCMP is in the development stage and a horizontal relay routed message passing protocol exists among the parachains. However, this protocol is resource intensive and stores all the messages at the relay chain. Proprietary chains such as Bitcoin and Ethereum, etc. can also be connected to the PolkaDot network using the PolkaDot bridges. The bridge acts as a trust-free gateway, allowing proprietary chains to post or route transactions from (or to) PolkaDot network. PolkaDot ecosystem consists of two types of bridges: bridge modules and bridge contracts. The bridge module enables external chains to act as a virtual parachain. Bridge contract is more like a bridge module, however, unlike bridge modules, a bridge contract functionality is limited to the parachains that support the execution of smart contracts. The project supports building bridges between PolkaDot ecosystem and proprietary chains. The future upgrades and changing user's requirements in the PolkaDot ecosystem are decided by the validators using referendum. PolkaDot is open-source [21] and its smart contracts are written in WebAssembly (WASM) [91].

4) DISCUSSION

A number of the existing solutions use native tokens of their respective chain to enable cross-chain transactions, i.e., off-chain transactions, trading platforms. To use the services of these chains in cross-chain transactions, the interacting users or blockchains must have an account and investment for trading and transaction fee. Wanchain [87] issue native tokens to the account holders of other chains by locking their respective tokens. This requires a continuous flow of tokens between the interacting chains. Moreover, for every transaction, Wanchain incurs an overhead of creating a one-time account.

In [20] and [21], all the transactions and chains are administered by the hub and relay chain. As the number of connected chains increases, the network performance may degrade because all the transactions are routed via a central entity. Similarly, the cost of syncing all the chains will also increase. In [20], the zones and Cosmos hub continuously send messages to each other in order to be aware of each other's state. Whereas, in [21] parachains are able to spam one another with transaction data. Currently, there is no mechanism to prevent parachains from spamming one another. In [86], the use of encoded listeners moves ACES towards centrality. Similarly, in [83], a special set of nodes called block rovers are incorporated to broadcast the recently mined transactions in the interacting chains. Block rovers are remote clients from the interacting chains.

Overledger [89] can connect any chain regardless of the underlying structure or technology. Similarly, ICON republic [88] use SCORE that does not require a dedicated virtual machine. This makes the interoperability solution not limited to a specific chain.

VI. FUTURE DIRECTIONS AND CHALLENGES

This section discusses the future directions and challenges associated with blockchain interoperability.

A. TYPES OF ASSETS EXCHANGED

The existing solutions on blockchain interoperability mostly focus on token or assets exchange between two or more chains. In our opinion, there is a need for an interoperable architecture that focuses on the exchange of data between chains. Moreover, assets or tokens can be exchanged for services as well. Data exchange can be used for many purposes such as invoking smart contracts, verifying transactions, etc. Interoperable blockchain architecture with data sharing capabilities will greatly improve blockchain interoperability.

B. SCALABILITY

Scalability is one of the major reasons due to which the widespread adaptation of blockchains is slower than expected. Though interoperability solutions are capable to achieve horizontal scalability by integrating a number of chains. Vertical scalability for cross-chain transactions is still far from reality due to the underlying consensus and validations process of each chain. For cross-chain transactions, one way to achieve vertical scalability is to have a dynamic list of participants from both chains to validate cross-chain transactions by suspending the local consensus mechanisms in the interacting chains.

C. PRIVACY AND SECURITY IN CROSS-CHAIN TRANSACTIONS

It is likely that all blockchains have their own mechanisms to guarantee security and privacy. However, the fact that there are always threats cannot be ignored. As in the case

[21] https://github.com/w3f/polkadot

of intermediary chains or forwarders who facilitate the interacting chains, transaction anonymity and user identity can be compromised. It is possible to breach the integrity of the interacting chains by breaching the security of the intermediary blockchain. Moreover, these facilitator chains between the interacting chains are recording each transaction they forward by means of their forwarders. Strict monitoring policies such as force to forget must be applied on the interacting chains to guarantee security and privacy in cross-chain transactions.

D. COMPATIBILITY AND RECOVERY MECHANISM

A blockchain interoperability solution must be compatible with the existing as well as new chains. It is evident that in the coming years the number of blockchains will increase. If the interoperable solutions are not compatible with the novel solutions, they may not survive in the long run. Moreover, none of the existing solutions discussed their recovery mechanism in the event of a failure.

VII. CONCLUSION

The popularization of blockchain technology has disrupted many application areas. There is high anticipation among researchers that the true potential of this technology will be revealed by achieving interoperability between isolated blockchains. This paper discussed the role of smart contracts in blockchain interoperability. The study shows that it is possible to invoke a remote smart contract from a local blockchain. Similarly, a virtualization-based approach can also be used to achieve interoperability. Based on our survey, it is concluded that smart contract-based blockchain interoperability solutions can be a potential game-changer for the widespread adoption of blockchain technology. Unfortunately, there is a dearth of literature analyzing the importance of smart contracts in blockchain interoperability solutions. Hopefully, this study will serve as a foundation for researchers working on smart contract based interoperable blockchains.

REFERENCES

[1] S. Nakamoto, "Re: Bitcoin P2P e-cash paper," in *The Cryptography Mailing List*. 2008.

[2] V. Buterin. (2016). *What is Ethereum?*. Accessed: Feb. 28, 2021. [Online]. Available: http://www.ethdocs.org/en/latest/introduction/what-is-ethereum.html

[3] R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: An introduction," *R3 CEV*, vol. 1, p. 15, Aug. 2016.

[4] *Quorum*. Accessed: Feb. 27, 2021. [Online]. Available: https://github.com/ConsenSys/quorum/wiki

[5] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Proc. Workshop Distrib. Cryptocurrencies Consensus Ledgers*, 2016, vol. 310, no. 4, pp. 1–4.

[6] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2009–2030, 3rd Quart., 2020.

[7] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices," *Appl. Sci.*, vol. 10, no. 2, p. 488, Jan. 2020.

[8] S. Aslam, M. P. Michaelides, and H. Herodotou, "Internet of ships: A survey on architectures, emerging applications, and challenges," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9714–9727, Oct. 2020.

[9] O. Samuel and N. Javaid, "A secure blockchain-based demurrage mechanism for energy trading in smart communities," *Int. J. Energy Res.*, vol. 45, no. 1, no. 2021, pp. 297–315.

[10] A. Shahid, A. Almogren, N. Javaid, F. A. Al-Zahrani, M. Zuair, and M. Alam, "Blockchain-based agri-food supply chain: A complete solution," *IEEE Access*, vol. 8, pp. 69230–69243, 2020.

[11] P. Bhaskar, C. K. Tiwari, and A. Joshi, "Blockchain in education management: Present and future applications," *Interact. Technol. Smart Educ.*, to be published.

[12] P. Singh, M. Masud, M. S. Hossain, and A. Kaur, "Cross-domain secure data sharing using blockchain for industrial IoT," *J. Parallel Distrib. Comput.*, vol. 156, pp. 176–184, Oct. 2021.

[13] P. Singh, M. Masud, M. S. Hossain, and A. Kaur, "Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid," *Comput. Electr. Eng.*, vol. 93, Jul. 2021, Art. no. 107209.

[14] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-based electronic healthcare record system for healthcare 4.0 applications," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102407.

[15] *Gartner*. Accessed: Jan. 26, 2021. [Online]. Available: https://www.gartner.com/en

[16] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, Jun. 2020.

[17] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille. (2014). *Enabling Blockchain Innovations With Pegged Sidechains*. [Online]. Available: http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains

[18] B. Chandra Ghosh, V. Ramakrishna, C. Govindarajan, D. Behl, D. Karunamoorthy, E. Abebe, and S. Chakraborty, "Decentralized cross-network identity management for blockchain interoperation," 2021, *arXiv:2104.03277*. [Online]. Available: http://arxiv.org/abs/2104.03277

[19] S. D. Lerner, "RSK," Tech. Rep., 2015.

[20] J. Kwon and E. Buchman, "Cosmos whitepaper," White Paper, 2019.

[21] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," White Paper, 2016.

[22] M. Spoke, "Aion: Enabling the decentralized internet," AION, White Paper, Jul. 2017.

[23] T. M. Mayer, C. Mai, and N. Jesse, "Tokrex: Meta-system for real-time intra-and cross-chain swaps," Tech. Rep., 2017.

[24] A. Culwick and D. Metcalf. (2018). *The Blocknet Design Specification*. [Online]. Available: https://www.blocknet.co/wp-content/uploads/2018/04/whitepaper.pdf

[25] D. Li, J. Liu, Z. Tang, Q. Wu, and Z. Guan, "AgentChain: A decentralized cross-chain exchange system," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 491–498.

[26] J. Lee, "Komodo: An advanced blockchain technology, focused on freedom," Komodo Platform, Komodo, Tech. Rep., 2018.

[27] E. Scheid, B. Rodrigues, and B. Stiller, "Toward a policy-based blockchain agnostic framework," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 609–613.

[28] P. Frauenthaler, M. Sigwart, C. Spanring, and S. Schulte, "Testimonium: A cost-efficient blockchain relay," 2020, *arXiv:2002.12837*. [Online]. Available: http://arxiv.org/abs/2002.12837

[29] A. Garoffolo, D. Kaidalov, and R. Oliynykov, "Zendoo: A zk-SNARK verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains," 2020, *arXiv:2002.01847*. [Online]. Available: http://arxiv.org/abs/2002.01847

[30] S. Ghaemi, S. Rouhani, R. Belchior, R. S. Cruz, H. Khazaei, and P. Musilek, "A pub-sub architecture to promote blockchain interoperability," 2021, *arXiv:2101.12331*. [Online]. Available: http://arxiv.org/abs/2101.12331

[31] P. Frauenthaler, M. Borkowski, and S. Schulte, "A framework for blockchain interoperability and runtime selection," 2019, *arXiv:1905.07014*. [Online]. Available: http://arxiv.org/abs/1905.07014

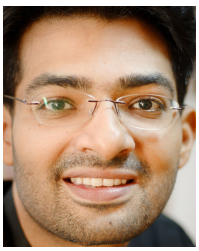
[32] N. Szabo, "Formalizing and securing relationships on public networks," *1st Monday*, vol. 2, no. 9, Sep. 1997.

[33] J. Chen, X. Xia, D. Lo, J. Grundy, and X. Yang, "Maintaining smart contracts on ethereum: Issues, techniques, and future challenges," 2020, *arXiv:2007.00286*. [Online]. Available: http://arxiv.org/abs/2007.00286

[34] *Solidity Programming Language*. Accessed: Mar. 1, 2021. [Online]. Available: https://docs.soliditylang.org/en/v0.7.4/
[35] *Neo Whitepaper*. Accessed: Mar. 1, 2021. [Online]. Available: https://docs.neo.org/docs/en-us/
[36] *IO, EOS, Technical White Paper*. Accessed: Mar. 1, 2021. [Online]. Available: https://github.com/EOSIO/Documentation
[37] J. Kwon, "Tendermint: Consensus without mining," *Draft V. 0.6, Fall*, vol. 1, no. 11, 2014.
[38] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," *IEEE Access*, vol. 7, pp. 50759–50779, 2019.
[39] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," 2019, *arXiv:1906.11078*. [Online]. Available: http://arxiv.org/abs/1906.11078
[40] V. Buterin, "Chain interoperability," R3 Research Paper, 2016.
[41] M. Borkowski, D. McDonald, C. Ritzer, and S. Schulte, "Towards atomic cross-chain token transfers: State of the art and open questions within tast," Distrib. Syst. Group TU Wien (Technische Universität Wien), Vienna, Austria, Tech. Rep., 2018.
[42] M. Borkowski, P. Frauenthaler, M. Sigwart, T. Hukkinen, O. Hladký, and S. Schulte, *Cross-Blockchain Technologies: Review, State of the Art, and Outlook*. 2019, pp. 1–5.
[43] T. Koens and E. Poll, "Assessing interoperability solutions for distributed ledgers," *Pervas. Mobile Comput.*, vol. 59, Oct. 2019, Art. no. 101079.
[44] A. Singh, K. Click, R. M. Parizi, Q. Zhang, A. Dehghantanha, and K.-K.-R. Choo, "Sidechain technologies in blockchain networks: An examination and state-of-the-art review," *J. Netw. Comput. Appl.*, vol. 149, Jan. 2020, Art. no. 102471.
[45] V. A. Siris, P. Nikander, S. Voulgaris, N. Fotiou, D. Lagutin, and G. C. Polyzos, "Interledger approaches," *IEEE Access*, vol. 7, pp. 89948–89966, 2019.
[46] N. Kannengießer, M. Pfister, M. Greulich, S. Lins, and A. Sunyaev, "Bridges between islands: Cross-chain technology for distributed ledger technology," in *Proc. 53rd Hawaii Int. Conf. Syst. Sci.*, 2020, pp. 1–10.
[47] M. H. Miraz and D. C. Donald, "Atomic cross-chain swaps: Development, trajectory and potential of non-monetary digital token swap facilities," *Ann. Emerg. Technol. Comput.*, vol. 3, no. 1, pp. 42–50, Jan. 2019.
[48] S. Johnson, P. Robinson, and J. Brainard, "Sidechains and interoperability," 2019, *arXiv:1903.04077*. [Online]. Available: http://arxiv.org/abs/1903.04077
[49] I. A. Qasse, M. Abu Talib, and Q. Nasir, "Inter blockchain communication: A survey," in *Proc. ArabWIC 6th Annu. Int. Conf. Res. Track*, 2019, pp. 1–6.
[50] H. Tam Vo, Z. Wang, D. Karunamoorthy, J. Wagner, E. Abebe, and M. Mohania, "Internet of blockchains: Techniques and challenges ahead," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1574–1581.
[51] S. Schulte, M. Sigwart, P. Frauenthaler, and M. Borkowski, "Towards blockchain interoperability," in *Proc. Int. Conf. Bus. Process Manage.* Cham, Switzerland: Springer, 2019, pp. 3–10.
[52] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," 2020, *arXiv:2005.14282*. [Online]. Available: http://arxiv.org/abs/2005.14282
[53] M. Herlihy, "Atomic cross-chain swaps," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2018, pp. 245–254.
[54] E. Fynn, A. Bessani, and F. Pedone, "Smart contracts on the move," in *Proc. 50th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2020, pp. 233–244.
[55] M. Black, T. Liu, and T. Cai, "Atomic loans: Cryptocurrency debt instruments," 2019, *arXiv:1901.05117*. [Online]. Available: http://arxiv.org/abs/1901.05117
[56] A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, and W. Knottenbelt, "XCLAIM: Trustless, interoperable, cryptocurrency-backed assets," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 193–210.
[57] *P2PHK Transactions*. Accessed: Dec. 21, 2020. [Online]. Available: https://en.bitcoinwiki.org/wiki/Pay-to-Pubkey-Hash
[58] G. FalaziEmail, U. Breitenbücher, F. Daniel, A. Lamparelli, F. Leymann, and V. Yussupov, "Smart contract invocation protocol (SCIP): A protocol for the uniform integration of heterogeneous blockchain smart contracts," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Cham, Switzerland: Springer, 2020, pp. 134–149.
[59] G. Falazi, A. Lamparelli, U. Breitenbuecher, F. Daniel, and F. Leymann, "Unified integration of smart contracts through service orientation," *IEEE Softw.*, vol. 37, no. 5, pp. 60–66, Sep. 2020.
[60] *BTC Relay*. Accessed: Dec. 2, 2020. [Online]. Available: http://btcrelay.org//
[61] *COMIT Project*. Accessed: Dec. 6, 2020. [Online]. Available: https://comit.network/docs/comit-protocol/comit-projects/
[62] *A Cryptofinance Platform*. Accessed: Dec. 2, 2020. [Online]. Available: https://uploads-ssl.webflow.com/Whitepaper.pdf
[63] N. Shadab, F. Houshmand, and M. Lesani, "Cross-chain transactions," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2020, pp. 1–9.
[64] B. Dai, S. Jiang, M. Zhu, M. Lu, D. Li, and C. Li, "Research and implementation of cross-chain transaction model based on improved hash-locking," in *Proc. Int. Conf. Blockchain Trustworthy Syst.* Singapore: Springer, 2020, pp. 218–230.
[65] M. Sigwart, P. Frauenthaler, T. Hukkinen, and S. Schulte, "Towards cross-blockchain transaction verifications," Tech. Rep., 2019. [Online]. Available: http://www.infosys.tuwien.ac.at/tast
[66] M. Sigwart, P. Frauenthaler, C. Spanring, and S. Schulte, "Preparing simplified payment verifications for cross-blockchain token transfers," Tech. Rep., 2019. [Online]. Available: https://dsg.tuwien.ac.at/projects/tast
[67] P. Frauenthaler, M. Sigwart, C. Spanring, and S. Schulte, "Leveraging blockchain relays for cross-chain token transfers," *Gas*, vol. 300, p. 600, Mar. 2020.
[68] E. Abebe, D. Behl, C. Govindarajan, Y. Hu, D. Karunamoorthy, P. Novotny, V. Pandit, V. Ramakrishna, and C. Vecchiola, "Enabling enterprise blockchain interoperability with trusted data transfer (industry track)," in *Proc. 20th Int. Middleware Conf. Ind. Track*, Dec. 2019, pp. 29–35.
[69] *Peace Relay*. Accessed: Nov. 2, 2020. [Online]. Available: https://peacerelay.io/
[70] I. Barinov, V. Baranov, and P. Khahulin. *Proof of Authority Network*. Accessed: Nov. 2, 2020. [Online]. Available: https://www.poa.network/
[71] R. Khalil, A. Gervais, and G. Felley, "NOCUST-a non-custodial 2nd-layer financial intermediary," IACR Cryptol. ePrint Arch., Tech. Rep., 2018, p. 642.
[72] Y. Ye and W. Wu, "Garou: An efficient and secure off-blockchain multi-party payment hub," 2020, *arXiv:2010.07555*. [Online]. Available: http://arxiv.org/abs/2010.07555
[73] L. Wang, J. Wu, R. Yuan, D. Zhang, J. Liu, S. Jiang, Y. Zhang, and M. Li, "Dynamic adaptive cross-chain trading mode for multi-microgrid joint operation," *Sensors*, vol. 20, no. 21, p. 6096, Oct. 2020.
[74] M. J. Amiri, D. Agrawal, and A. El Abbadi, "CAPER: A cross-application permissioned blockchain," *Proc. VLDB Endowment*, vol. 12, no. 11, pp. 1385–1398, 2019.
[75] K. Sai and D. Tipper, "Disincentivizing double spend attacks across interoperable blockchains," in *Proc. 1st IEEE Int. Conf. Trust, Privacy Secur. Intell. Syst. Appl. (TPS-ISA)*, Dec. 2019, pp. 36–45.
[76] P. Robinson, D. Hyland-Wood, R. Saltini, S. Johnson, and J. Brainard, "Atomic crosschain transactions for ethereum private sidechains," 2019, *arXiv:1904.12079*. [Online]. Available: http://arxiv.org/abs/1904.12079
[77] M. Darisi, J. Savla, M. Shirole, and S. Bhirud, "STEM: Secure token exchange mechanisms," in *Proc. Int. Conf. Adv. Cyber Secur.* Singapore: Springer, 2019, pp. 206–219.
[78] P. Robinson and R. Ramesh, "General purpose atomic crosschain transactions," 2020, *arXiv:2011.12783*. [Online]. Available: http://arxiv.org/abs/2011.12783
[79] M. Nissl, E. Sallinger, S. Schulte, and M. Borkowski, "Towards cross-blockchain smart contracts," 2020, *arXiv:2010.07352*. [Online]. Available: http://arxiv.org/abs/2010.07352
[80] B. Pillai, K. Biswas, and V. Muthukkumarasamy, "Cross-chain interoperability among blockchain-based systems using transactions," *Knowl. Eng. Rev.*, vol. 35, 2020.
[81] M. Sigwart, P. Frauenthaler, C. Spanring, and S. Schulte. *Towards Cross-Blockchain Smart Contracts*. Accessed: Feb. 27, 2021. [Online]. Available: https://dsg.tuwien.ac.at/index.html
[82] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, and Y.-C. Hu, "Hyperservice: Interoperability and programmability across heterogeneous blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 549–566.
[83] D. K. Bhavani. *Block Collider an Ultimate Unifier for Crypto*. Accessed: Nov. 21, 2020. [Online]. Available: https://www.blockcollider.org/
[84] *What is Borderless?*. Accessed: Nov. 21, 2020. [Online]. Available: https://docs.blockcollider.org/docs/

[85] H. Tian, K. Xue, S. Li, J. Xu, J. Liu, and J. Zhao, "Enabling cross-chain transactions: A decentralized cryptocurrency exchange protocol," 2020, *arXiv:2005.03199*. [Online]. Available: http://arxiv.org/abs/2005.03199
[86] *ARK Public Network, Point Click Blockchain*. Accessed: Nov. 23, 2020. [Online]. Available: https://ark.io/Whitepaper.pdf
[87] L. Jack, B. Yang, Z. Liang, Y. Zhang, D. Shi, E. Swartz, and L. Lu, "Wanchain," Tech. Rep., 2017.
[88] *ICON: Hyper Connect the World, ICON Foundation*. Accessed: Nov. 23, 2020. [Online]. Available: https://icon.foundation/?lang=en
[89] G. Verdian, P. Tasca, C. Paterson, and G. Mondelli, *Quant Overledger Whitepaper*. Accessed: Feb. 28, 2021. [Online]. Available: https://uploads-ssl.webflow.com/Quant_Overledger_Whitepaper_2019.pdf
[90] *PolkaDot XCMP*. Accessed: Nov. 23, 2020. [Online]. Available: https://wiki.polkadot.network/docs/en/
[91] *W3C, Web Assembly*. Accessed: Nov. 23, 2020. [Online]. Available: https://webassembly.org/

SAJJAD KHAN received the B.S. degree in computer science from the University of Peshawar, in 2012, and the M.S. degree in computer science from COMSATS University Islamabad, Pakistan, in 2019. He is currently pursuing the Ph.D. degree in computer science. His research interests include blockchain interoperability, distributed computing, optimization, and scheduling and forecasting using machine learning/deep learning techniques.



MUHAMMAD BILAL AMIN received the M.S. degree from DePaul University, Chicago, IL, USA, in 2006, and the Ph.D. degree from Kyung Hee University, South Korea, in 2015. He is currently a Korea Research Fellow and a Lecturer with the Department of ICT, University of Tasmania, Australia. He has an experience of more than ten years in the software industry, working for Fortune 500 companies in USA. He is an author of more than 50 publications. His research interests include blockchain, distributed systems, software engineering and architecture, and performance-based cloud applications.

AHMAD TAHER AZAR (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees from the Faculty of Engineering, Cairo University, Egypt, in 2006 and 2009, respectively. He is currently a Research Professor with Prince Sultan University, Riyadh, Saudi Arabia. He is also an Associate Professor with the Faculty of Computers and Artificial Intelligence, Benha University, Egypt. He worked in the areas of control theory and its applications, process control, chaos control and synchronization, nonlinear control, robust control, and computational intelligence. He served as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, from 2013 to 2017, and *ISA Transactions* (Elsevier), from 2018 to 2020. He is the Editor-in-Chief of the *International Journal of System Dynamics Applications* (IJSDA) (IGI Global, USA) and the *International Journal of Intelligent Engineering Informatics* (IJIEI) (Inderscience Publishers, Olney, U.K.). He is currently an Associate Editor of the IEEE SYSTEMS JOURNAL.

SHERAZ ASLAM (Member, IEEE) received the B.S. degree in computer science from Bahauddin Zakariya University (BZU), Multan, Pakistan, in 2015, and the M.S. degree in computer science with a specialization in energy optimization in the smart grid from COMSATS University Islamabad(CUI), Islamabad, Pakistan, in 2018. He is currently pursuing the Ph.D. degree with the DICL Research Laboratory, Cyprus University of Technology (CUT), Limassol, Cyprus, under the supervision of Dr. H. Herodotou. During his M.S. degree, he worked as a Research Associate with Dr. N. Javaid at CUT. He has authored more than 50 research publications in ISI-indexed international journals and conferences such as the IEEE INTERNET OF THINGS (IoT) JOURNAL, *Renewable and Sustainable Energy Reviews*, IEEE ACCESS, *Electrical Power System Research*, and *Energies*. He is also a part of European Union funded research project named as STEAM. His research interests include data analytics, generative adversarial networks, network security, wireless networks, smart grid, and cloud computing. He also served as a TPC member and an invited reviewer for international journals and conferences.