Received May 29, 2020, accepted June 15, 2020, date of publication June 29, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005663

Consortium Blockchain-Based Decentralized Stock Exchange Platform

HAMED AL-SHAIBANI, NOUREDDINE LASLA, (Member, IEEE), AND MOHAMED ABDALLAH, (Senior Member, IEEE)

Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University (HBKU), Doha, Qatar

Corresponding author: Hamed Al-Shaibani (halshaibani@mail.hbku.edu.qa)

ABSTRACT The global implementation architecture of the traditional stock market distributes responsibilities and data across different intermediaries, including financial and governmental organizations. Each organization manages its system and collaborates with the others to facilitate trading on the stock exchange platform, and typically buy-sell orders go through different parties before settlement. This design architecture that involves a complex chain of intermediaries has several limitations and shortcomings, such as a single point of failure, a longer time for financial settlements, and weak transparency. Blockchain technology consists of a network of computer nodes that securely share a common ledger without the need of having any kind of intermediaries. In this paper, we present a novel blockchain-based architecture for a fully decentralized stock market. Our architecture is based on a private Ethereum blockchain to create a consortium network leveraging organizations that are already involved in the traditional stock exchange to act as validating nodes. In our architecture, the stock exchange trading logic is completely implemented on a smart contract, while considering the existing governmental market regulations. Since the new platform does not introduce significant changes to the stock exchange trading logic and does not eliminate any of the traditional parties from the system, our proposal promotes efficient adoption and deployment of decentralized stock exchange platforms. In addition, we present a proof of concept implementation of the new architecture, including the smart contract for trade exchange, as well as a virtualization-based test network to assess the platform performance. The test network consists of virtual nodes that run the developed stock exchange smart contract where we measure the buy-sell orders throughput and latency under different network sizes and trading workload scenarios. The obtained results have shown that the proposed trading platform can reach a throughput of 311.8 tx/sec, which is equivalent to 89% of the optimal throughput when the sending rate is 350 tx/sec. This throughput is largely sufficient to meet the requirement of major stock exchanges, such as Singapore stock market.

INDEX TERMS Blockchain, smart contract, stock exchange, trading.

I. INTRODUCTION

The stock market is a platform composed of financial and governmental organizations that participate in exchanging shares, bonds, or other securities in a transaction known as a trade. The growth of this market has a positive and direct impact on the financial growth of a country's economy since it offers opportunities that attract investors to trade and exchange shares. Studies conducted in USA [1] and Pakistan [2] examine this relationship by comparing the stock market performance with the Gross Domestic Product (GDP). The authors conclude that the performance of the stock exchange is directly proportional to the performance of a country's economy. For instance, Pakistan achieved growth in both the GDP and stock market of about 30% and 6.08%, respectively, between 2003 and 2008. Therefore, the stability and security of the stock platform are vital in increasing the confidence to invest and trade and eventually results in better economic growth for the country.

Despite the wide popularity and adoption of the conventional stock exchange platform architecture, it suffers from different limitations and shortcomings as follows [3], 1) due to the centralization architecture of the stock exchange platform, each participating system such as brokers and the stock

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

exchange is considered as a single point of failure, 2) there is inconsistency in the data managed by each system resulting in errors and extended recovery time when failures happen, 3) the availability of the system on a daily basis is limited which can impact exercises such as auditing the data as well as providing transparent access to the users throughout the day, and 4) the long time the system takes to perform financial settlements; typically it takes three days after trading happens to achieve the settlement. There have been many attempts by major vendors who build trade execution and matching engines in the financial sector to address the above limitations. For instance, Nasdaq, which is one of the leading tradings and matching technology vendors, is offering different services and products such as hosting, managing and providing services to the complete end to end trading processes. They also provide surveillance systems integrated with other systems to allow their clients to monitor and regulate the process of trading and settlement. However, this approach still suffers from limitations such as the long settlement time, the limited level of data transparency, and the defined trading hours. Most importantly, clients might require hardware specifications or the need to follow an enforced architecture that involves the distribution of the systems involved in the trading process across multiple organizations where each manages its system separately. For instance, some stock market regulators enforce specific cash settlement and clearance solutions, which is a different system than what Nasdaq provides [4]. This results in maintaining multiple systems, which increases the chance of having a single point of failure among the participating systems as well as increasing the complexity of the overall trading platform system architecture.

According to [5], blockchain can solve many of the identified limitations affecting the traditional stock exchange platform, such as the lack of transparency, the long settlement time between brokers and the central bank, and the high transaction fees paid to brokers for each generated trade. In [3], the centralized architecture of Bucharest stock exchange market has been analyzed to address its limitations with the main objective of addressing the issue of the high fees the investor pays to the broker for each successfully executed trade. The authors define the new stock market in a smart contract and deploy it into the Ethereum public network. This implementation requires a form of payment, in Ethereum cryptocurrency (Ether), for each performed transaction. Their conclusion shows that decentralizing the Bucharest stock exchange platform can help in reducing the total transaction fees.

The research objective and implementation approach to decentralize the Bucharest stock exchange platform varies with our objective and the approach we took in several key areas. First, we are using a consortium blockchain network in which all participants are known and trusted, and there is no form of cryptocurrency fees that will be used to pay the miners in the network. Second, our main objective is to optimize the performance of the decentralized system rather than reducing the fees, as we measure the throughput and latency to ensure our implementation meets the required level of the stock market platform. Also, our consensus algorithm is based on Proof of Authority (PoA). It provides better performance in terms of execution time and power efficiency in comparison with the public network consensus algorithms such as Proof of Work (PoW) used by the decentralized Bucharest stock exchange. The consensus algorithms will be discussed in more detail later in this paper. TABLE 1 provides definitions of the acronyms used in the paper.

TABLE 1. List of acronyms.

Abbreviation	Definition
B_{time}	Validation time difference between last and first blocks
CB	Central Bank
CSD	Central Securities Depository
DPoS	Delegated Proof of Stake
FMA	Financial Market Authority
FOK	Fill or Kill Order
GDP	Gross Domestic Product
IOC	Immediate or Cancel Order
N	Number of Transactions
NIN	National Investor Number
OMT	Order Management System
P2P	Peer-to-Peer Network
PoA	proof of Authority
PoB	Proof of Burn
PoI	Proof of Importance
PoS	Proof of Stake
PoW	Proof of Work
SE	Stock Exchange
TPS	Transaction per Second

In this paper, we propose a consortium blockchain-based stock exchange platform that meets the performance requirement of the stock exchange platform while also addressing the limitations of the traditional stock exchange. Our proposal is based on Ethereum blockchain technology in which all necessary business regulations and rules defined in a smart contract shared across a permissioned blockchain network with the participating financial and governmental institutions. We perform experimental tests by deploying the smart contract on virtual nodes and measuring the network performance under different workloads by increasing the number of generated trades and the number of validating nodes. Our results show that this architecture does meet the required performance of the stock exchange platform in terms of latency and throughput under different test scenarios.

The remaining of this paper is organized as follows: Section II provides an overview of the traditional stock exchange platform. Section III presents an overview of blockchain technology and discusses the different implementations and consensus algorithms of blockchain for public and private implementations. Section IV discusses the related work focusing on the implementation design based on blockchain for stock market and an E-auction system. Our proposed blockchain-based stock exchange framework is presented in Section V, where we discuss the system architecture and the smart contract defining the functionalities and business logic of stock exchange. In section VI,

TABLE 2. Major entities of a traditional stock market.

Entity	Description
Investor	An individual who would like to trade (buy/sell shares)
Listed Company	A company listed in the market and has shares or other securities to offer for trading
Broker	A firm that acts on behalf of the investor by placing orders directly on the Stock Exchange platform
Custodian	A firm that is responsible for securing the investor's finances and needed in case of international investors
Stock Exchange (SE)	An institution that hosts a matching engine system that receives orders and generates trades
Central Securities Depository (CSD)	A financial institution that manages investors accounts and post trade cash settlements
Government	It is responsible for validating the authenticity of investors data
Financial Market Authority (FMA)	An institution that regulates and monitors how the stock market operates
Central Bank (CB)	it is responsible for executing the net payment settlement transactions between the custodians and brokers

we evaluate the performance of the proposed architecture in terms of transaction throughput and latency. Finally, Section VII concludes the paper.

II. TRADITIONAL STOCK EXCHANGE OVERVIEW

The stock market can be defined as "an aggregation of buying and selling offers corresponding to an asset" [6]. The asset has a form of bonds, shares, or other securities that the market offers. The person who trades in the stock market is called investor or trader and needs first to open a trading account with the Central Securities Depository (CSD), which takes the responsibility of managing investors' trading accounts and personal data. Due to market regulation, investors cannot directly place an order into the system and need to go through a third party, namely brokers. In the case of international traders, a special financial entity, named Custodian, is employed to place orders in the local market. The matching engine of the entered buy and sell orders is hosted by the Stock Exchange (SE) entity. The Central Bank (CB) manages the financial settlement between brokers and custodians. All the participating governmental and financial organizations need to follow the rules and regulations defined by the Financial Market Authority (FMA). FMA is also responsible for continuously monitoring the stock exchange platform and reviewing the data.

A summary of the different entities involved in the traditional stock market with their respective description is given in TABLE 2.

A. TRADING OVERVIEW

The traditional stock market is a centralized platform as shown in FIGURE 1, which presents this architecture and the flow of events that take place when a new investor participates in the platform. First, the investor needs to open an investor account from CSD and obtains his/her National Investor Number (NIN). The investor then needs to open a trading account with a broker by providing the mandatory NIN account. Once the investor information is validated, he/she can place orders of buying or selling shares through the associated broker services such as the website or the mobile application. The broker takes the responsibility of using its Order Management System (OMT), which acts as an interface with SE to submit the investor's order. Once a successful trade is generated for that order, SE sends the acknowledgment message to the broker, who then notifies the user via the different services provided by the broker. The shares owned by the investor are updated in the broker account as well as the investor account held in CSD system. The market regulator FMA has access to both SE and CSD systems to monitor the market and validate the trades during and after trading hours.

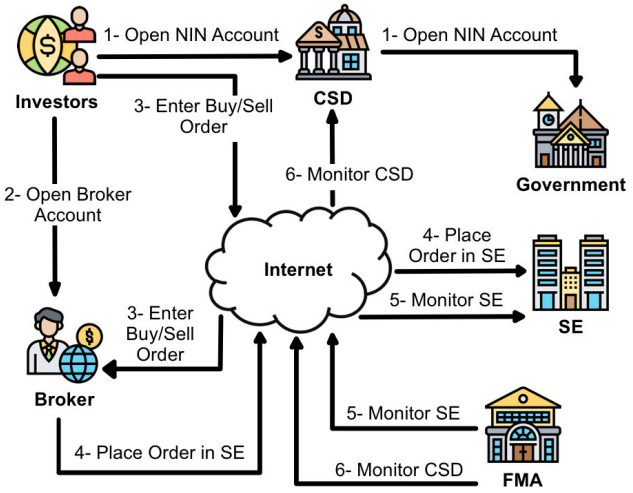


FIGURE 1. Centralized stock exchange platform flow of events.

B. TRADING HOURS AND PHASES

There are usually four different phases that a stock market goes into in most implementations, as shown in FIGURE 2. The market starts with a 30 minutes pre-open phase in which investors can enter their orders, but no trades are generated. Based on bids and offers entered, opening prices for listed securities are calculated, so when the market opens, those calculated prices will be the buy/sell prices used by the investors. The next phase is market opening for trading in which the listed securities can be traded, and orders entered in the pre-open phase gets executed. This is the main phase of the stock exchange, where orders keep entering, and trades get generated. The duration of this phase is approximately 3 hours and 30 minutes, and this time can vary from one stock exchange to another as per the regulations of the country hosting it. The market then prepares for closing and enters the pre-close phase, which is estimated to last for 10 minutes.

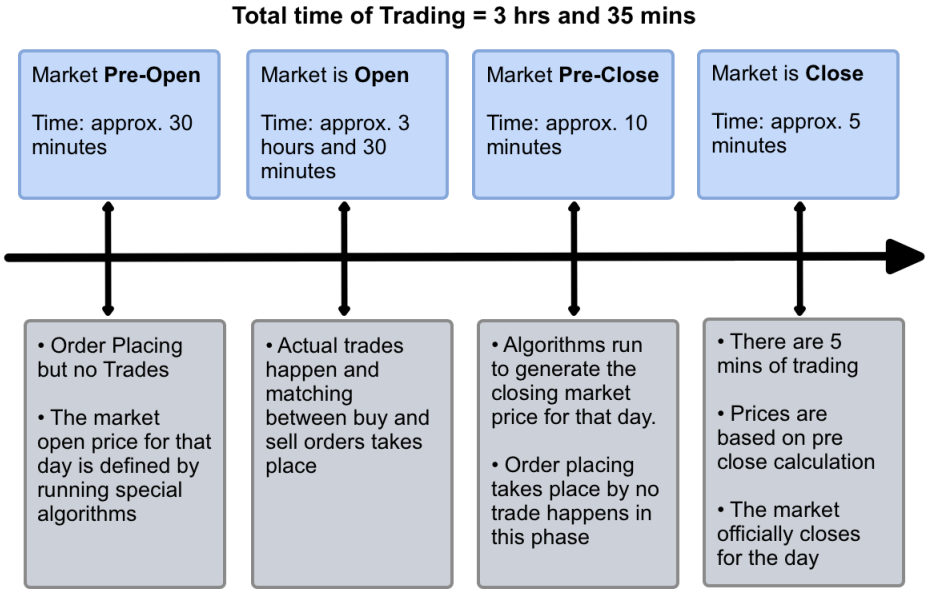


FIGURE 2. Stock exchange market trading hours and market phases.

In this phase, algorithms run to generate the closing prices for listed securities, and investors can still enter their orders, but no trade is generated. Finally, the market enters the closing phase, which is estimated to last for 5 minutes in which entered orders get executed. The market then closes for the day.

C. ORDER TYPES

Investors can place different types of orders in terms of buying or selling shares [3]. These types are listed below to explain how a successful match between buy and sell transactions happen, as some orders allow the investors to specify conditions on which if triggered, the order gets executed and hence, results in generating trades:

- **Market Order:** The buyer would like to buy the shares with the current price of the share in the market. The same applies to the seller. This order type does not guarantee the highest financial gain from the transaction but ensures the order is immediately executed when entered
- **Limit Order:** The buyer sets a maximum limit he/she is willing to pay to buy a particular share. The order gets executed for any sell offer that is equal to or less than this limit. In the case of the seller, the limit will be the minimum price he/she is willing to sell with. The order gets executed for a buy order with a price higher than or equals to the minimum limit.
- **Validity Defined Order:** This order can be associated with either a limit or market order types. The entered order will remain valid for a single day (Day Order) or until a certain date (Good Till Date Order). In most stock exchange markets, the system cancels the order in approximately 62 days if no validity date is provided (Open Order).
- **Fill or Kill (FOK) Order:** Either execute the full order (Sell or buy all indicated shares) or cancel the order
- **Immediate or Cancel (IOC) Order:** The order is immediately executed, and the remaining quantity that has not been fulfilled will be canceled.

III. BLOCKCHAIN OVERVIEW

To address the limitations and shortcomings of the traditional stock exchange platform, we opt for Blockchain technology. Blockchain can be defined as a ''network of computers, all of which must approve a transaction has taken place before it is recorded, in a 'chain' of computer code. The details of the transfer are recorded on a public ledger where anyone on the network can see the information [7]. It consists of blocks with each containing a pointer in the form of a hash of the previous block and verified transaction data protected with hash signatures [8]–[10]. Transactions in blockchain are broadcasted in the network and are validated by a process known as mining that is performed by special nodes in the network known as miner nodes [7]. Miner nodes are specific nodes that append a new block to the chain once the block becomes full. It is extremely difficult to change a block in the chain as it requires to have subsequent blocks to be recreated, and hence this mechanism prevents modification and maintains a high level of security. FIGURE 3 shows the content of the first three blocks.

As shown, each block consists of the list of transactions, the hash of the previous block (except for the first block), a nonce value, which is a number that can be used only once. In some blockchain implementations such as Bitcoin the nonce is altered by the miner such that the hash of the block is equal to or less than a certain numerical target value provided by the network as a challenge. The block also contains the hash of the block itself.

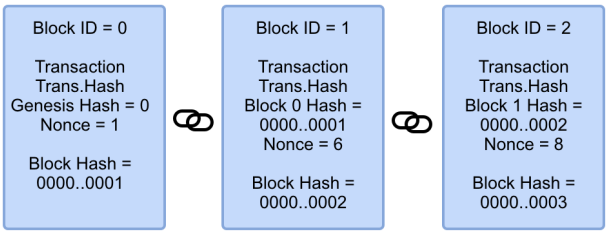


FIGURE 3. The first three blocks in a Blockchain.

In order to keep track of all transactions, blockchain ledger is used in a network where participants have access to the same ledger replicating the transactions among all peer nodes in that network. This replication ensures that the overall system built on blockchain can resume if multiple participating nodes failed to connect to the network. The nodes in the network use addresses or identifiers known as public keys to be distinguished by, and hence, defined roles, privacy, and anonymity can be efficiently maintained [11]. Miner nodes rely on the fact that all transactions in the network are duplicated across all nodes involved. Therefore, ''Distributed Consensus'' needs to be achieved, meaning that an agreement on the validity of the blockchain is achieved by all nodes involved and they all share the same version of the Blockchain [12].

Some implementations of Blockchain such as Ethereum, uses protocols to present the logic that need to be followed and this is known as a smart contract. According to [13] a smart contract can be defined as ''the computer protocols that digitally facilitate, verify, and enforce the contracts made between two or more parties on blockchain''. The smart contract ensures that the defined logic is validated and needs to be followed. It does not need a centralized entity to validate the defined conditions in the smart contract since once it is deployed, all participating nodes in the network need to follow the defined logic in it.

A. BLOCKCHAIN IMPLEMENTATION TYPES

In Blockchain, all nodes need to reach a state of agreement or consensus on the next block that needs to be added to the chain, especially that in a peer to peer network such as blockchain these nodes don't trust each other. However, there are many consensus mechanisms that can be used depending on the implementation type and the blockchain technology used. There are mainly two types of implementation for Blockchain network: permissionless and permissioned.

1) PERMISSIONLESS BLOCKCHAIN (PUBLIC)

Permissionless or public implementation of blockchain allows any user to become a node and connect to the network through the internet. The implementation utilizes the concept of Peer-to-Peer network (P2P) that uses distributed architecture in which no client takes the form of an administrator. All clients on the network are connected by flat topology where each peer shares the same rights and privileges as other peers and have access to the same resources as other peers [7], [14], [15].

2) PERMISSIONED BLOCKCHAIN (PRIVATE)

Unlike the permissonless blockchain, nodes in permissioned blockchain are identified and authenticated. In some implementations, an entity takes the responsibility of managing the roles and responsibilities of the nodes and granting them permission to the data accordingly.

B. CONSENSUS ALGORITHMS

There are several consensus algorithms for the implementation of public blockchain network such as Proof of Work (PoW), Proof of Stake (PoS), Proof of Burn (PoB), Delegated Proof of Stake (DPoS), and Proof of Importance (PoI). PoW and PoS are the two most famous and commonly used consensus algorithms [16]. PoW is an intensive hashing mechanism that provides a difficult mathematical challenge for the block miners to solve, and whoever manages to solve the challenge first will become the block miner [17]. This protocol ensures integrity among all nodes but suffers greatly when it comes to its performance time, processing, and energy power required [17]. On the other hand, PoS consensus protocol is more power-efficient and reduces mining costs. This protocol takes less time than PoW to validate a transaction as it relies on validators taking part in voting for the next block, and the weight of each validator's vote is dependent on how much it deposited in that system. Nodes that are allowed to create a block act as validators who need to deposit some cryptocurrency as a stake in the network that will be locked to have the chance of being selected as the next block miners. The more stake a validator has in the network, the higher the chance of it being selected to validate the new block. Such protocol ensures acting correctly since any validator who violates the network rules or acts maliciously will lose its stake deposited in the network [18]. PoS has several advantages such as consuming less power and energy, better performance time, and the mechanism of having a stake that can be lost for any malicious behavior is expected to pressure validators to act genuinely more than in PoW.

In the case of permissioned blockchain networks, PoA and RAFT are popular consensus algorithms where the participants are known and trusted in a private network. According to [17], PoA is an algorithm that attracted a lot of attention due to its offered performance resulting from lighter exchanged messages. It operates in rounds where several nodes are elected, with one of them acting as a mining leader charged with the task of proposing the new block and eventually reaching consensus. These elected nodes are called ''authorities,'' and each has its unique ID in which if we have N authorities, at least N/2 +1 are assumed to be honest. This algorithm follows a ''mining rotation schema'' to distribute the block creation among the authorities in a fair manner, and for each round step, a mining leader from the authority is elected to mine the new block [17].

TABLE 3. Comparison between the consensus algorithms.

Consensus Algorithem	Energy Consumption	Computational Power	Throughput	Byzantine fault tolerance	Network Type
PoW	High	High	Low	Yes	Permissionless
PoS	Low	Low	High	Yes	Permissonless
PoA	Low	Low	High	Yes	Permissioned
RAFT	Low	Low	High	No	Permissioned

In [19], the author argues that RAFT consensus is easy to understand and implement, which makes it efficient to use when building applications and systems. It works by having a set timer for all authorized nodes, which can validate new blocks in ''terms'' that can be seen as rounds that get repeated over time. For a given term, as the timer runs out for the first authority, it enters what is called ''candidate state'', in which it votes for itself to become a leader and broadcasts requests to other authorities to vote for it. If the majority positively voted for the candidate node, then it becomes the leader of that term. Once a leader is elected, its role is to replicate the transaction logs across all other nodes. The logs reach finality and get committed by the leader if and only if it reached the majority of the nodes, once this happens, the leader will commit the log and asks the rest to do the same via a broadcast message. In case the majority of the nodes are offline, the leader will not be able to commit the logs, and there is a high risk of losing the log if the leader and the remaining nodes went offline [19].

TABLE 3 provides a comparison between the permissionless and permissoined consensus algorithms presented in this paper. For the proposed solution, all network participants should be known and trusted. The selected consensus algorithm should allow an authorized participant to act as an administrator for the overall platform since FMA regulates the stock market, and its role is required to be perceived. Moreover, the network should be Byzantine fault tolerance in case some of the network validators act maliciously. PoA consensus algorithm satisfies these requirements. Moreover, for our proof-of-concept implementation, the Geth implementation of PoA, named Clique, is adopted. Clique has a rotation schema for leader election, such that in each round, the leader of the round announces the block and it gets added to the blockchain by the receiving nodes [17].

IV. RELATED WORK

According to [20], implementations of blockchain in the financial sector focus on four main areas, which are improving the transaction processing time, having sustainability for banking and financial transactions, improving financial data privacy and security, and automating financial contracts. For transaction time improvement, the authors highlight that the current banking systems rely on centralized databases that require several days to achieve financial settlements for the executed transactions [21]. The solution that blockchain offers to solve this problem, according to the authors, is to automate financial transaction settlement by setting up a single account structure that will be used by financial institutions, as well as speeding up international fund transfers [21]. Sustainability is another problem that banks and financial institutions suffer from, especially when a bankruptcy of one bank can have a strong impact on the overall financial sectors. The authors in [24] argue that implementing blockchain can lead the financial sector to achieve stability, especially when the decentralized ledger of money is independent of financial regulations of countries and regions. Financial data security and privacy currently face many challenges due to the nature of the centralized data storage that banking and financial instructions rely on [22]. This can lead to data breaches that does reveal not only financial data, but also personal and demographic data that were also stored in the centralized storage. In addition, banking transactions do not provide sufficient anonymity or extending the freedom of privacy that clients would like to have. Blockchain addresses these two issues by decentralizing the data and ensuring they are securely stored in the participating nodes, which add high complexity to unauthorized attempts to alter or access the stored data. Each participant is authorized to perform changes according to the role assigned while maintaining anonymity on transactions performed [23]. Finally, authors in [20] highlight that blockchain automates financial contracts in terms of execution by eliminating the need of a third party in the middle and allowing a financial transaction to be triggered between the two involved parties. To demonstrate such a feature, money transfer usually takes a couple of days, especially in developing countries, as some controls and regulations need to be verified. When such a transaction is implemented using a financial contract in blockchain, it will no longer require a third party intervention as long as both parties perform their roles as defined in the contract. The financial transaction will be securely executed, and the money will be transferred within minutes [24].

We have analyzed two particular implementations that resemble a close similarity to our idea. The first paper discusses the concept of decentralizing the stock market platform by using Blockchain technology while the second paper utilizes the concept of smart contract in blockchain to build a bidding platform.

A. DECENTRALIZING BUCHAREST STOCK MARKET PLATFORM

In [3], the authors discuss the limitation of the traditional stock market and propose a solution to implement the trading platform on Blockchain. Their research objective is to showcase how transaction fees can be reduced if blockchain

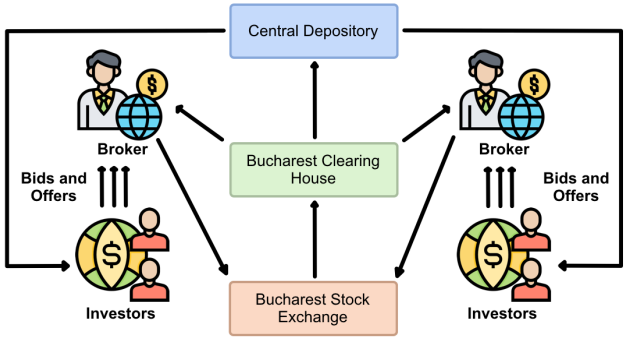


FIGURE 4. Bucharest centralized stock exchange platform [3].

technology is used as a trading platform instead of the traditional stock exchange platform the Bucharest stock market uses. To test their experiment, the authors have implemented two systems with the first one being modeled according to Bucharest centralized stock exchange platform as per FIGURE 4 in which all orders entered by different brokers are gathered in a single system. The second system implemented is a decentralized blockchain based solution that uses a smart contract to simulate the stock exchange platform. This design does not require to have brokers to enter orders, and instead, investors can interact directly with the system and enter the order themselves. By doing so, the fees paid to brokers are eliminated, and the fees the investors pay per transaction in the proposed blockchain based trading system overall is less than the fees paid in the traditional stock exchange platform. The authors conclude that the fees in the decentralized system will increase as the number of orders in the order book increases since the transaction complexity will become higher. Therefore, the decentralized system will be giving a better transaction fee than the centralized system when the order book is partially full.

B. BIDDING SYSTEM BASED ON BLOCKCHAIN SMART CONTRACT

An e-auction system has several elements that are in common with the stock exchange platform. It consists of bidders, auctioneers, and third-party intermediaries who provide the platform that connects bidders to auctioneers and allows posting products, checking the highest bidding price, and declaring the winner with the highest bidding price. The authors in [10] suggest building an e-auction system without having intermediaries between the sellers and buyers by using Ethereum based smart contract. Their objective is to solve two main problems the current e-auction systems have, which are the limited level of security offered by the online platform and the high transaction fees users have to pay. The authors claim that their blockchain based solution addresses the first problem by ensuring security related to data shared among the different users of the system is appropriately managed and perceived. The second problem is addressed by reducing the transaction cost by removing any intermediary in the system. FIGURE 5 shows a flowchart representing the bidding process taking

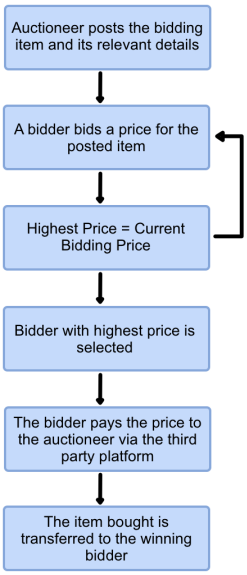


FIGURE 5. Flowchart showcasing the bidding process [10].

place from start to finish. First, the seller posts the bidding information and the starting price. The bidders bid the price in the sealed envelope, and when it is received by the auctioneer, the sealed envelope price that is the highest is announced as the current highest price. If no price received higher than the current bidder's highest price or the ending time is due, it is announced as the winning price, and the auctioneer can send the product and receive the money from the winning bidder [7]. By applying the proposed blockchain based e-auction platform as an experiment, the authors conclude that the smart contract can enforce confidentiality, non-repudiation, and prevention of unauthorized alteration of entered bidding orders.

TABLE 4 showcases the main differences between the proposed platform and the already discussed related work. For instance, our main research objective is to improve the performance of the system in terms of availability, security, and transparency by adopting a consortium blockchain based on Ethereum with PoA consensus algorithm. We are maintaining all key participants in the traditional stock exchange platform to be part of the proposed platform. We are not introducing major changes that conflict with the roles and regulations imposed by the government. For the case of the decentralized Bucharest stock exchange, the research objective is to reduce the transaction fees paid to the brokers by making significant changes to the existing architecture and eliminating the broker completely from the platform. The new proposed architecture is based on permissionless Ethereum network that uses PoW consensus algorithm. This new platform introduces new fees that are less than the fees paid to the brokers in the traditional stock exchange for cases with partially full order book. The objective of the second related work is to build a secure e-auction system without having intermediaries. The authors used a permisionless Etherem network with a PoW consensus algorithm and made changes to the

TABLE 4. Comparing our paper with related works.

Research Subject	Objective	Consensus Algorithm	Network Type	Pay to Miners
Decentralizing Bucharest Stock Market Platform	Transaction fee reduction	PoW	Permissionless	Yes
Bidding System Based on Blockchain Smart Contract	Transaction fee reduction and security	PoW	Permissionless	Yes
Consortium Blockchain-Based Decentralized Stock Exchange Platform	Performance improvement	PoA	Permissioned	No

traditional bidding platform by removing intermediaries managing it. Both related works rely on using a public blockchain network which has poor performance and cannot handle the required throughput and latency of the current stock exchange.

V. PROPOSED BLOCKCHAIN-BASED STOCK EXCHANGE FRAMEWORK

In this section, we describe our proposed decentralized stock exchange platform that is based on a consortium blockchain between financial and organizational entities that are already part of the traditional stock market. We first give an overview of the system architecture, define the roles and responsibilities of the participating entities, and finally present the smart contract holding and managing the stock exchange trading logic.

A. SYSTEM ARCHITECTURE

As shown in FIGURE 6, our system is composed of a consortium blockchain network, a smart contract, and financial and organizational entities. The consortium blockchain facilitates transactions between the different participating entities and manages the stockeExchange smart contract that handles the stock trading logic. We select a permissioned blockchain as the entities are all known and also because private version of blockchain is more effective in terms of transaction throughput and latency. The consortium network is composed of a set of authorized participants (validators) which are the CSD, FMA, Broker, Government, and SE. Each of them has specific roles and responsibilities as per the traditional stock exchange platform. The StockExchange smart contract defines all the trading logic as well as the different functions that can be performed by the participating entities, such as, create broker, create new investor, assign share to investor, etc. Each participating entity has a private key along with the associated address and public key that are used for authentication. Therefore, the smart contract ensures that each entity is only allowed to trigger functions according to its associated privileges. TABLE 5 summaries the StockExchange smart contract functionalities and the entities authorized to execute each of them. The detail description of the role of each of the participating entities is given in the following:

- **FMA:** it is responsible for creating and maintaining the smart contract as well as defining all the trading logic and functionalities. It also monitors the trading process and ensures that all defined rules and regulations are properly maintained. It interacts with the smart contract to create and maintain companies with shares and to create and maintain brokers.

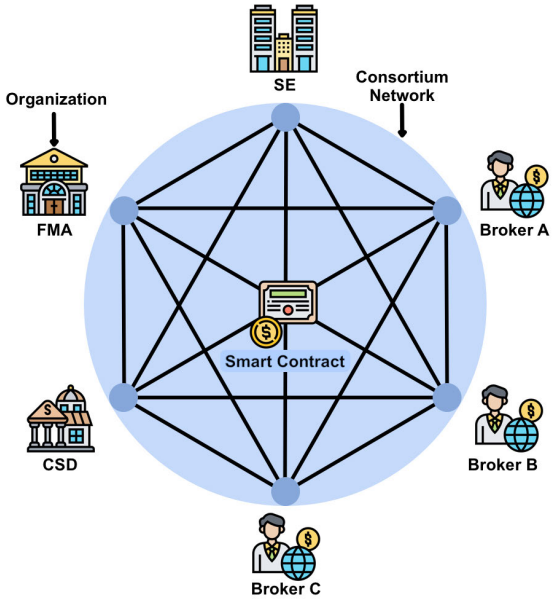


FIGURE 6. System architecture.

TABLE 5. StockExchange smart contract authorizations.

Functionalities	Entities				
	FMA	*SE*	*Broker*	*CSD*	*Government*
Create/maintain smart-contract	Yes	No	No	No	No
Create/maintain list of brokers	Yes	No	No	No	No
Create/maintain list of companies	Yes	No	No	No	No
Buy/sell shares	No	No	Yes	No	No
Generate trades	No	Yes	No	No	No
Create new investor account (NIN)	No	No	No	Yes	No
Associate broker to investor	No	No	Yes	No	No
Assign share to NIN	No	No	No	Yes	No
Validiate NIN	No	No	No	Yes	Yes

- **CSD:** is responsible for creating and maintaining investor accounts. It interacts with the smart contract to create investor accounts and assign shares to them.
- **Broker:** it takes the role of trading on behalf of investors. It interacts with the smart contract by associating investors to it and entering buy and sell orders for the associated investors. Brokers are also authorized to assign shares from CSD investor accounts to the investor trading account managed by the broker. Each investor can have multiple trading accounts managed by a different broker for each, while each investor must have a single unique investor account (NIN).
- **Government:** it validates the investor data sent by CSD
- **SE:** it is responsible for matching orders queued in the order book, and generating trades.

B. StockExchange SMART CONTRACT

We define a smart contract called ''StockExchange'' to include the business logic and the authorization roles of each participating entity. The smart contract manages the buy and sell orders and generates respective trades whenever a buy order offers a price that is equal to or more than the sell order's price. The different steps of the trading, implemented in the smart contract, are detailed below:

1) Let QB be a queue of all buy orders sorted in ascending order such that i represents the index of the maximum element in the queue denoted as B^P. Let B^Q denotes the quantity of the shares in B^P.
2) Let QA be a queue of all sell orders sorted in descending order such that y represents the index of the minimum element in the queue denoted as A^P. Let A^Q denotes the quantity of the shares in A^P.
3) We assume that all orders entered are of the types limit order or market order and that partially matched orders are possible in cases where $B^Q \neq A^Q$
4) If $B^P \geq A^P$ and $B^Q = A^Q$, both orders are fully matched, and a trade is generated. Both i and y indices are decremented by 1.
5) If $B^P \geq A^P$ and $B^Q \geq A^Q$, B^P is partially matched with A^P, and a trade is generated. The value of B^Q is updated such that $B^Q = B^Q - A^Q$ and y index is decremented by 1.
6) If $B^P \geq A^P$ and $B^Q < A^Q$, A^P is partially matched with B^P, and a trade is generated. The value of A^Q is updated such that $A^Q = A^Q - B^Q$ and i index is decremented by 1.

FIGURE 7 shows the sequence diagram between the participating entities and the smart contact, including all the steps required before generating trades and matching buy/sell orders. The detail description of each of the diagram steps are given in the following:

1) FMA defines the list of all brokers that the stock market consists of by calling the ''addBroker'' function. The system replies with a message showing the successful creation of the broker.
2) FMA defines the companies that are listed in the stock market along with their details such as number of shares they consist of and their prices. The function ''addCompany'' is used for this purpose.
3) CSD Validates the investor's data integrity by sending it to the government. The government replies to the smart contract to update the investor validation status.
4) CSD assigns to each validated investor a new investor account number ''NIN'' by using the function ''addNin''.
5) The broker associates an investor to its account using the function ''AssociateBrokerToInvestor''. The smart contract then validates by checking the NIN Account subsystem to ensure that the NIN exists. If it does, the NIN gets associated to the broker account successfully.
6) The broker assigns shares that are stored in the investor's NIN account maintained by CSD to the trading account maintained by the broker, by calling the function ''AssignShareToNin''.
7) Buy orders are entered by the broker into the smart contract. Once these orders are entered, the ''StockExchange'' subsystem logs and stores the order in a sorted queue and tries to match these orders with existing sell orders pending in the sell queue list. If a successful match is generated, the system replies back to the broker that successful trades have been generated for the entered orders. If no match could be generated, the broker will be informed that the orders have been successfully entered the system.
8) This step is similar to step 7 as brokers enter sell orders into the smart contract. If a successful match is generated, the broker is informed about it or else; the broker will be informed that the orders have successfully entered the system.

C. SECURITY AND SYSTEM EFFICIENCY ANALYSIS

The proposed blockchain-based stock exchange architecture ensures the following security and system efficiency:

1) **Transparency:** the level of transparency provided by using blockchain guarantees that all transactions and data maintained by the system are visible to the authorized participants and cannot be manipulated. However, any change requires consensus and commitment from all network participants before it gets validated. In contrast, the traditional stock exchange suffers from insufficient transparency level as each party has its system and can hide or manipulate the data before sharing it with other participants.
2) **High availability:** the proposed architecture addresses the single point of failure by ensuring high availability through decentralizing the data across multiple participants. The smart contract can still be executed even if some nodes were disconnected from the network. Contrary to the traditional stock market, if any of the system participants is unavailable, the whole market is affected.
3) **Network efficiency:** in the stock exchange, the quality of network connectivity has a critical impact on investors' profits. For instance, an order sent by an investor through his/her associated broker can be delayed by the network if the broker has connectivity issues, or it is physically located far from the SE. Orders that were entered later by other brokers, with better network connectivity or located physically closer to the SE, will be executed first. This results in a financial loss to the investor despite entering the order first and can cause a lack of fairness and trust in the overall platform. The blockchain network provides better connection utilization between the different participants since nodes are distributed in different physical locations. The node

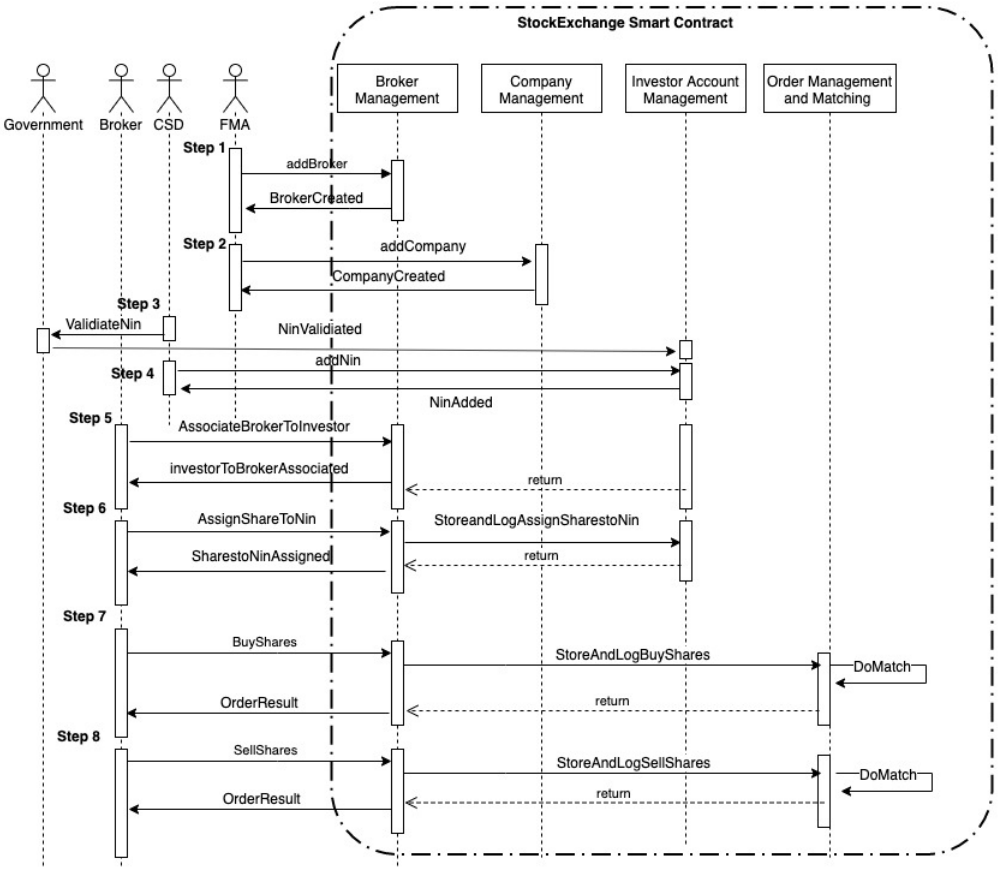


FIGURE 7. Sequence diagram between participating entities and the StockExchange smart contract.

physically located closest to the users interacting with the smart contract will receive the transactions and broadcast them to the remaining nodes in the network.

4) **Consistency:** since the ledger is shared across different participants, they all have the same version of the data, and any change that happens in one node will be immediately reflected in the ledgers of the other nodes. This solves the issue of having conflicting data that are not synchronized across the participating systems as it is in the traditional stock exchange platform. For instance, if an investor updates his/her personal data directly with CSD without updating it in the broker system too, a delay in authenticating transactions happens thus, impacting the investor's profit.
5) **Cost efficiency:** in our proposed architecture, contrary to the traditional stock exchange, all the participating entities use the same common software and platform, which consists of an Ethereum smart contract. This solution architecture is much simpler and cost-effective as it considerably decreases the overall system complexity and cost for maintenance and technical support. In addition, the proposed architecture is highly available and does not require a separate disaster recovery environment. This saves a high cost compared to the traditional stock market, where each participating entity needs to have a specific disaster recovery site.
6) **Flexible configuration:** the proposed architecture provides more flexibility and scalability in comparison with the traditional stock exchange platform when it comes to adjusting the functionalities and introducing new changes to the trading logic. Since the proposed design architecture consists of the StockExchange smart contract, new and existing functionalities, as well as authorization, can all be managed in one place. The smart contract can then be shared in the network without requiring participants to make changes in the hardware and storage, which makes it much easier to adopt.
7) **Smart contract security:** In order to design secure smart contracts, authors in [24] and [25] recommend a set of analysis tools to identify security issues and vulnerabilities in the smart contract code. Among the most famous analysis tools, we selected SmartCheck [24] to assess the proposed "StockExchange" smart contract. SmartCheck allowed us to identify multiple security-related issues and optimize some functions in

our initial design. Such issues include the extra gas consumption due to the use of multiple loops and bad array manipulation, which, if not appropriately addressed, can lead to a storage overlap attack where it collides with other data in the storage. Moreover, the tool provided multiple recommendations, such as upgrading the solidity code to the latest version as well as emphasizing on the declarations of public and private modifiers.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed blockchain-based stock exchange platform, in terms of transaction throughput and latency, to showcase its capability in handling the transaction load of the current stock market. We validate our results with Singapore Exchange, which is one of the emerging markets offering a diversity of listed securities for tading. For this purpose, we developed a testing framework that consists of three main modules: network module, transaction generation and listening module, and performance evaluation module. The description of each module is given below:

1) **Network module:** This module is used to create the consortium blockchain test network that hosts our defined ''StockExchange'' smart contract. It consists of the entities that resemble the stock exchange participants, which are SE, FMA, Brokers, government, and CSD. These entities are represented as Ethereum nodes by using Docker container technology, where each node runs the Geth Ethereum client. The selected consensus algorithm is PoA (see Section III for more details).
2) **Transaction generation and listening module:** This module is implemented using a JavaScript API that serves as a generator of transaction workload and listens to the blockchain for block confirmation events. By continuously listening to the network, this module records information such as block number, validation time, and the number of transactions per block. The transaction workload consists of buy and sell orders, and the total number of generated transactions at each round of testing is configurable. To ensure that each pair of buy and sell orders generates a trade, we generate for each buy order a corresponding sell order. The workload generator and data listener module interact with a special gateway node in the network that receives the transactions and broadcasts them to the network.
3) **Performance evaluation module:** This module is used to analyze the information stored in the data listener module and measure the performance of each experiment by calculating the throughput and latency for entered orders and generated trades. The throughput or number of transactions per second (TPS) is calculated as the total number of transactions (N) divided by the time it takes to validate them, which is the time difference between the block with the first transaction and the block with the last transaction:

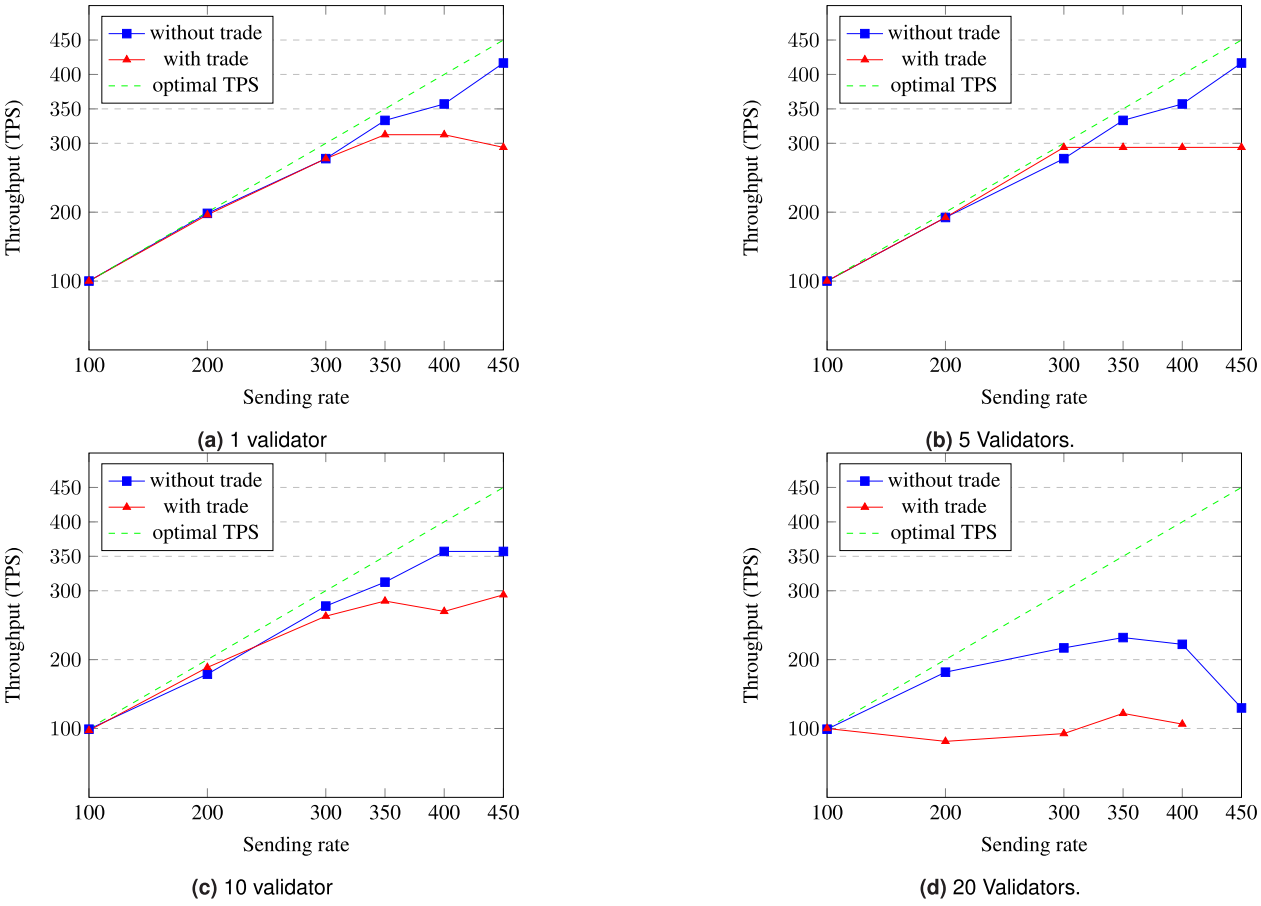
$$TPS = N/B_{time},$$

where B_{time} is the difference in validation time between the last and the first blocks.
The latency is a measurement that shows the difference in time between the time a transaction is sent and the time it gets validated in a block. It is calculated as the total time it takes to process X number of transactions divided by X.

A. EXPERIMENT

We have implemented our proposed stock exchange platform, which has been built on top of a consortium blockchain network, using Solidity, the de-facto scripting language to write smart contracts in Ethereum. The created smart contract consists of the following main functions:

1) **addBroker:** adds a new broker to the system by entering its name, its symbol, and the maximum amount of money it is allowed to spend buying shares in a single trading session.
2) **addCompany:** a new company is added after entering its name, symbol, its total number of shares, and the price per share.
3) **ValidateNIN:** investor's data received by CSD is sent to the government for validation. This data consists of the investor name, age, nationality, and ID number. The government responds in the form of true or false value, which CSD uses as a condition to either proceed or cancel the creation of the new NIN account.
4) **addNin:** for each validated investor, a unique investor number is assigned. This investor number is associated with the investor's personal data, including the total number of shares owned by the investor.
5) **AssociateBrokerToInevestor:** it assigns a broker to an investor by entering the broker's name, symbol, investor name, and NIN.
6) **AssignShareToNin:** shares are assigned to a given NIN and the total number of shares in the NIN is updated.
7) **buyShares:** a buy order that has the company's symbol, number of shares, price, and NIN enters a queue of buy orders. For each new buy order, the queue is sorted such that the order with the highest price is placed first, followed by the rest in descending order.
8) **sellShares:** a sell order that has the company's symbol, number of shares, price, and NIN enters a queue of sell orders. For each sell order, the queue gets sorted such that the order with the lowest price is placed the first, followed by the rest in acceding order.
9) **DoMatch:** this function is called as part of each ''buyShares'' and ''sellShares'' functions. It takes the first item in the buy orders queue and compares it with the first item in the sell orders queue. If the price of the buy order is more or equal to the price of the sell



(a) 1 validator

(b) 5 Validators.

(c) 10 validator

(d) 20 Validators.

FIGURE 8. Throughput vs sending rate (tx/sec).

order, then a trade is generated. The matched orders are removed from the queues and the queues again get sorted. The NIN accounts of the buyer and seller will be updated accordingly.

Several experiments have been conducted to measure the performance of our Stock exchange trading platform in terms of throughput and latency in which we adjusted our workload and network size for every round of testing. Six different workloads have been used in the form of the sending rate of transactions per second, which are: 100, 200, 300, 350, 400, and 450 tx/sec. The network size has also been adjusted such that the blockchain consisted of 1, 5, 10, and 20 validators for each test scenario. The time to construct two consecutive blocks has been fixed to 2 seconds, and the total number of transactions has also been fixed to 10,000 transactions where 5,000 represent buy orders, and the remaining 5,000 transactions are sell orders.

We categorized our test cases into the following workload scenarios:

1) "With Trades": in this scenario, orders are entered such that each pair of buy and sell orders generates a trade. It requires high computational power as the entered orders trigger the doMatch function that requires removing the matched orders from the queue and sorting the queues again as well as updating the buyer and seller NIN accounts accordingly.
2) "Without Trade": In this scenario, the buy and sell orders are not matched, and hence, no trade is generated. In terms of computational needs, this scenario yields the best throughput as it skips the doMatch function, which has to sort and to update investor accounts.

The experiments are conducted on a workstation machine with Intel(R) Xeon(R) Gold 6130 CPU, 2.10 GHz, 64 core CPU, 256GB RAM, and running Ubuntu 18.04.2.

FIGURE 8 illustrates the measured throughput under different sending rates and number of validators. In the case of a single validator node shown in FIGURE 8a, the throughput is very close to the sending rate up to 350 tx/sec. This is also valid in scenarios with 5 and 10 validators as shown in FIGURE 8b and FIGURE 8c, respectively. However, when the number of validators increases, the throughput gets considerably affected, as shown in FIGURE 8d with 20 validating nodes. This is due to the limited available computation power, as all the nodes in different scenarios share the same workstation machine. To emphasis the effect of computation power on the throughput, TABLE 6 shows the average throughput for transactions with and without trades

TABLE 6. Average throughput (with and without trades) for different network sizes.

Number of Validators \ Transaction Rate	100 tx/sec	200 tx/sec	300 tx/sec	350 tx/sec	400 tx/sec	450 tx/sec
01	100	197.20	287.53	322.91	325.75	355.35
05	100	192.30	285.93	313.71	325.62	355.35
10	98.04	183.82	270.40	298.80	313.55	325.62
20	99.50	131.55	154.79	176.95	164.29	64.90

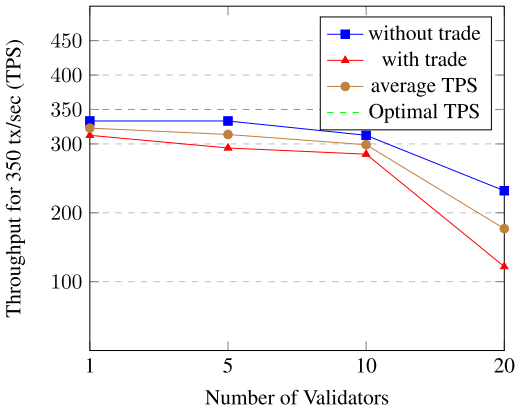


FIGURE 9. Throughput at 350 tx/sec Vs. number of validators.

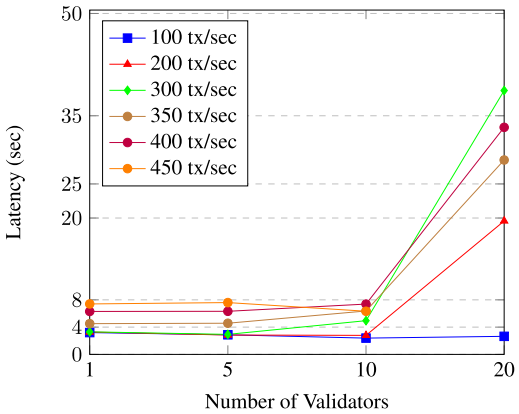


FIGURE 10. Latency Vs number of validators.

for different network sizes. Our finding shows that our system can support networks up to 10 validators and transaction rates up to 350 tx/sec. We plot in FIGURE 9 the result of the experiment when considering the two previously defined workload scenarios and their average value at a transaction rate of 350 tx/sec. The worst throughput is noticed for the case of the first workload scenario (with trade) as it requires more computation resources to complete the trade. For a network with up to 10 validating nodes, the average throughput is about 311.8 tx/sec. It is equivalent to 89% of the optimal throughput, which is the ratio of the average throughput value (311.8 tx/sec) to the optimal throughput value (350 tx/sec).

FIGURE 10 illustrates the effect of the different sending rates and the number of validators on the average transaction latency. The results show that the latency is inversely proportional to the throughput. For a network with up to 10 validating nodes, and a sending rate up to 350 tx/sec, the average latency is about 5.5 seconds. Considering the block time, which is 2 seconds and the time needed to propagate the block in the network, this can be considered as a reasonable delay. However, the latency significantly increases for large network sizes and high sending rates, where it can reach 40 seconds. It can be related to the following two reasons:

1) The higher the sending rate, the larger the block size, and hence, it will require more time to propagate the block to all the nodes in the network.
2) The computational resources play a major role in the network's ability to handle high sending rates. This is due to the fact that each transaction needs to go into several steps such as validation, propagation to the network, execution, and its inclusion in a new block that will then be propagated again to the network to be executed by the other nodes. These steps require sufficient computational power to be able to handle high sending rates.

TABLE 7. Trading data for Singapore exchange for the month of April 2020.

Number of trading Days	21
Total number of Trades	10,285,596 Trades
Average Number of Trades per day	489790.2857 Trades
Trading Hours (hours of trading per day)	7 Hours
Average Number of Trades per Hour	244895.1429 Trades/hour
Average Number of Trades per minute	4081.585714 Trades/min
Average Number of Trades per second	68.02642857 Trades/sec
Estimated number of Transactions per second	204 tx/sec

The obtained results have shown that our proposed trading platform can reach a transaction throughput of about 311.8 tx/sec. By analyzing the trading data obtained from Singapore Stock Exchange during the month of April 2020 [26], TABLE 7 shows that the total number of performed trades is 10,285,596 for a period of 21 trading days with 7 trading hours each day, which results in 489790.2857 trades per day. If all these trades are to be processed by the platform within the same two hours of a certain day, we will have 244895.1429 trades per hour, which results in 68.02642857 trade per second. Since each generated trade consists of buy and sell orders matched together, the estimated total number of generated transactions per second is 3 times the total number of trades/sec, which is equivalent to 204.0792 tx/sec. It is clear that our proposed platform can easily meet the requirement of this market by only considering the available computation resources used during the experiment. We believe that increased performance could be achieved if more computation resources can be used during the experimental evaluation.

VII. CONCLUSION

In this paper, we have presented a new blockchain-based architecture for a fully decentralized stock market platform. Our architecture is based on Ethereum smart contract that is implemented on a consortium and permissioned network. To be aligned with the regulations of the stock market, we chose the validating nodes to be the financial and governmental organizations that are already involved in the traditional stock exchange platform. This new architecture addresses the limitations of the traditional stock exchange platform such as the single point of failure in the participating systems by replicating the data and smart contract across all participating nodes, the complexity and inefficiency of the data management which our solution solves by providing a shared ledger that can be easily updated and maintained, the limited level of transparency since now all transactions can be seen, the limited daily time to access the platform's data as now it is easier to monitor the blockchain and access it throughout the day, and offering a faster financial and cash settlement time instead of the three days needed after the trading session. In order to evaluate the performance of our system, several experiments were conducted where the throughput and latency were evaluated. We have used different workloads and network sizes to evaluate the performance and found that the achieved performance can meet the requirement of the stock market platform for network sizes up to 10 validators and up to a sending rate of 350 tx/sec. However, we found that for larger workloads or network sizes, the performance significantly declines due to the limited computational resources used in the experiment. However, since the proposed solution will run on a consortium permissioned network, we believe that the participating entities will be capable of accommodating the necessary computation resources in order to meet the latency and throughput levels of the stock exchange. We plan to conduct further study to address privacy-related concerns and include cryptography encryption in the same ledger such that only allowed participants can see their relevant transaction data. Our future work will also cover further enhancements in the proposed smart contract. For instance, we will cover the possibility of introducing new changes to an already deployed smart contract without causing disturbance to the overall stock exchange platform.

REFERENCES

[1] B. Comincioli, "The stock market as a leading indicator: An application of Granger causality," *Univ. Avenue Undergraduate J. Econ.*, vol. 1, no. 1, pp. 1–14, 1996.
[2] M. S. Nazir, M. Nawaz, and U. Gilani, "Relationship between economic growth and stock market development," *Afr. J. Bus. Manage.*, vol. 4, pp. 3473–3479, Dec. 2010.
[3] C. Pop, C. Pop, A. Marcel, A. Vesa, T. Petrican, T. Cioara, I. Anghel, and I. Salomie, "Decentralizing the stock exchange using blockchain an ethereum-based implementation of the bucharest stock exchange," in *Proc. IEEE 14th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2018, pp. 459–466.
[4] N Inc. (2019). *Trading and Matching Technology Provides Flexible, Multi-Asset Trading Capabilities for Marketplaces of all Sizes*. [Online]. Available: https://www.nasdaq.com/solutions/trading-and-matching-technology
[5] L. Lee, "New kids on the blockchain: How Bitcoin's technology could reinvent the stock market," *SSRN Electron. J.*, vol. 12, no. 2, pp. 81–132, 2016.
[6] V. V. Bhandarkar, A. A. Bhandarkar, and A. Shiva, "Digital stocks using blockchain technology the possible future of stocks?" *Int. J. Manage.*, vol. 10, no. 3, pp. 44–49, Jun. 2019.
[7] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *Proc. IEEE Technol. Eng. Manage. Conf. (TEMSCON)*, Jun. 2017, pp. 137–141.
[8] M. Samaniego, U. Jamsrandorj, and R. Deters, "Blockchain as a service for IoT," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Dec. 2016, pp. 433–436.
[9] T. Lundqvist, A. de Blanche, and H. R. H. Andersson, "Thing-to-thing electricity micro payments using blockchain technology," in *Proc. Global Internet Things Summit (GIoTS)*, Jun. 2017, pp. 1–6.
[10] Y.-H. Chen, S.-H. Chen, and I.-C. Lin, "Blockchain based smart contract for bidding system," in *Proc. IEEE Int. Conf. Appl. Syst. Invention (ICASI)*, Apr. 2018, pp. 208–211.
[11] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized BlockChain for IoT," in *Proc. 2nd Int. Conf. Internet-of-Things Design Implement.*, Apr. 2017, pp. 173–178.
[12] M. Conoscenti, A. Vetro, and J. C. De Martin, "Blockchain for the Internet of Things: A systematic literature review," in *Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2016, pp. 1–6.
[13] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019.
[14] A. Kaushik, A. Choudhary, C. Ektare, D. Thomas, and S. Akram, "Blockchain—Literature survey," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 2145–2148.
[15] H. Kuzuno and C. Karam, "Blockchain explorer: An analytical process and investigation environment for bitcoin," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, Apr. 2017, pp. 9–16.
[16] M. Salimitari and M. Chatterjee, "A survey on consensus protocols in blockchain for IoT networks," Sep. 2018, *arXiv:1809.05613*. [Online]. Available: https://arxiv.org/abs/1809.05613
[17] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain," in *Proc. Italian Conf. Cyber Secur.*, Jan. 2018, p. 11. [Online]. Available: https://eprints.soton.ac.uk/415083/
[18] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018.
[19] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Conf. USENIX Annu. Tech. Conf. (USENIX ATC)*. Berkeley, CA, USA: USENIX Association, 2014, pp. 305–320. [Online]. Available: http://dl.acm.org/citation.cfm?id=2643634.2643666
[20] J. Jaoude and R. Saade, "Blockchain applications—Usage in different domains," *IEEE Access*, vol. 7, pp. 45372–45373, 2019, doi: 10.1109/ACCESS.2019.2902501.
[21] G. William Peters and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the Internet of money," 2015, *arXiv:1511.05740*. [Online]. Available: http://arxiv.org/abs/1511.05740
[22] Q. K. Nguyen, "Blockchain–A financial technology for future sustainable development," in *Proc. 3rd Int. Conf. Green Technol. Sustain. Develop. (GTSD)*, Nov. 2016, pp. 51–54.
[23] S. Singh and N. Singh, "Blockchain: Future of financial and cyber security," in *Proc. 2nd Int. Conf. Contemp. Comput. Informat. (IC3I)*, Dec. 2016, pp. 463–467.
[24] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," *IEEE Access*, vol. 7, pp. 50759–50779, 2019.
[25] M. Demir, M. Alalfi, O. Turetken, and A. Ferworn, "Security smells in smart contracts," in *Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2019, pp. 442–449.
[26] SGX. (Apr. 2020). *Market Statistics Report*. [Online]. Available: https://www2.sgx.com/research-education/historical-data/market-statistics

HAMED AL-SHAIBANI received the B.Sc. degree (Hons.) in computer science from Qatar University, Doha, Qatar, in 2010, and the M.Sc. degree in strategic business unit management from HEC Paris, Doha, in 2016. He is currently pursuing the Ph.D. degree in computer science and engineering with Hamad Bin Khalifa University, Doha. His main research interests include blockchain, cybersecurity, and networking.

NOUREDDINE LASLA (Member, IEEE) received the B.Sc. degree from the University of Science and Technology Houari Boumediene (USTHB), in 2005, the M.Sc. degree from the Superior Computing National School (ESI), in 2008, and the Ph.D. degree from USTHB, in 2015, all in computer science. He is currently a Postdoctoral Research Fellow with the Division of Information and Computing Technology, Hamad Bin Khalifa Univeristy, Qatar, with expertise in distributed systems, network communication, and cyber security.

MOHAMED ABDALLAH (Senior Member, IEEE) received the B.Sc. degree from Cairo University, in 1996, and the M.Sc. and Ph.D. degrees from the University of Maryland at College Park, in 2001 and 2006, respectively. From 2006 to 2016, he held academic and research positions at Cairo University and Texas A&M University at Qatar. He is currently a Founding Faculty Member with the rank of Associate Professor with the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). His current research interests include wireless networks, wireless security, smart grids, optical wireless communication, and blockchain applications for emerging networks. He has published more than 150 journals and conferences and four book chapters, and co-invented four patents. He was a recipient of the Research Fellow Excellence Award at Texas A&M University at Qatar, in 2016, the Best Paper Award in multiple IEEE conferences including the IEEE BlackSeaCom 2019, the IEEE First Workshop on Smart Grid and Renewable Energym in 2015, and the Nortel Networks Industrial Fellowship for five consecutive years, from 1999 to 2003. His professional activities include an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE OPEN ACCESS JOURNAL OF COMMUNICATIONS, a Track Co-Chair of the IEEE VTC Fall 2019 conference, a Technical Program Chair of the 10th International Conference on Cognitive Radio Oriented Wireless Networks, and a Technical Program Committee Member of several major IEEE conferences.