Journal of Theoretical and Applied Information Technology
15th March 2022. Vol.100. No 5
2022 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BLOCKCHAIN-BASED E-VOTING SYSTEM FOR ELECTIONS IN JORDAN

SARAH AL-MAAITAH[1], ABDULLAH QUZMAR[2], MOHAMMAD QATAWNEH[3]

[1]Department of Computer Science – The University of Jordan
[2]Department of Computer Science – The University of Jordan
[3]Department of Computer Science – The University of Jordan

sarah.khaldoon17@gmail.com, abdullahqz18@gmail.com, mohd.qat@ju.edu.jo

ABSTRACT

Any democracy must have a clear voting system that fits the demands of the people in order to deliver power to the proper person. Furthermore, current traditional voting methods have significant flaws, including a lack of security and transparency. This paper looks at how Blockchain (BC) technology may be used in E-voting systems to improve the voting process by addressing concerns like trust, privacy, and security. The proposed system uses a Hyper Ledger Fabric as a platform for creating Blockchain-based apps, software, and services with plug-and-play components including consensus, privacy, and membership services. We have analyzed the latency, response time, and throughput to make sure the system is performing well. As an outcome of the proposed system, we realized that proposed framework exceeds any other system in performance situations.

Keywords: *E-Voting, Blockchain, Voters, Voting System, Securit, Hyper Ledger Fabric.*

1. INTRODUCTION

The peoples of the world aspire to practice the democratic process freely, transparently and without the negative interference of governments in it. In most countries of the world, the electoral process is conducted using the traditional method, also known as paper elections, which are managed, monitored and implemented by government authorities or their representatives. Therefore, the electoral process may be accompanied by falsification of the will of the people, which may result in political and social problems that hinder the development of society [1]. Another problem facing the paper-pin elections is the slowness in conducting the electoral process, as well as the sorting and aggregation of electoral votes[1][2][14]. To solve some of the problems associated with the traditional election method, electronic voting or the use of information technology is used to speed up and facilitate the conduct of the electoral process. Despite the use of electronic voting to facilitate and monitor the electoral process, there is still a third party. Despite the use of electronic voting to facilitate and monitor the electoral process, there is still a third party that can modify the data, which leads to fraud, and accordingly, the issues of confidentiality, privacy and integrity must be taken into account during the conduction of the elections. These issues can be avoided by using Blockchain technology [1] [7]. Therefore, a novel system is highly needed to overcome the drawbacks previously mentioned. BC technology can be considered as a good candidate to solve many issues like security, trustless and transparency due to its attractive features such as immutability, distribution and decentralization [4][12][13].

This paper aims to implement a BC system to improve the voting process by addressing concerns like trust, privacy, and security. This means that the data in the e-voting election system must be transparent and available in the BC network without losses or tampering. In a BC system, the transactions are stored in a decentralized and immutable ledger where the transactions cannot alter without the alteration of the subsequent blocks of the system. This is because the BC uses several encryption and decryption ways [15] [16]. The paper's contribution is as follow:

- Proposing a new framework for the proposed system.
- Implementation and evaluation of the system presented in this paper.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The rest of this paper is organized as follows. Section 2 presents the theoretical background of BC technology and related works. Section 3 presents the proposed BC-based system for the E-voting system. Section 4 shows Simulation results and discussion. Finally, the conclusion is presented in section 5.

2. THEORETICAL BACKGROUND

The Blockchain technology was introduced by Satoshi Nakamoto in 2008 to underpin the Bitcoin, the first cryptocurrency. A BC technology can be defined as a distributed, decentralized and shared digital ledger that maintain a list of blocks. Over time, many recommendations have been proposed by several researchers to extend BC applications to non-financial domains like E-voting [1], used vehicle markets [11], drugs sector [3][9], etc. From the above definition, we can extract the following four essential features of BC:

• Decentralization: The BC is a decentralized system, which means that there is no single point of control responsible for security of the system, the control of the system is shared and managed through many independent entities such as computers or enterprises. To keep decentralization going every BC system must have a consensus algorithm to help the system make decisions, or else the core value of it is lost [10].

• Distribution and Sharing: The same copy of digital ledger is distributed and shared among many entities.

• Immutability: Immutability of BC means that the blocks which contain data or transactions can't be altered retroactively without the alteration of all subsequent blocks and the consensus of the system.

All of these features made a BC highly secure system and thus it became a good candidate to address many issues such as security, privacy, counterfeiting, transparency and trustless in several domains of our life, especially the e-voting systems [2][4][8].

There are numerous research papers that touched the field of election process, but few of those papers cover the e-voting systems based on Blockchain technology. Therefore, due to the lack of research in BC-based e-voting systems, this paper will highlight most of the research papers related to it. In [5] the authors proposed a BC-based electronic voting system, which comprises three parts: the voter side, electoral commissions, and the BC network. The goal of the proposed system is to solve the universal verifiability issues in Estonian voting system. The authors tried combine the double envelope encryption technique and Blockchain technology for our proposed electronic voting system.

In [6] the authors propose a novel electronic voting system based on Blockchain that addresses some of the limitations in existing systems and evaluates some of the popular Blockchain frameworks for the purpose of constructing a Blockchain-based e-voting system. The proposed system improves the security and decreases the cost of hosting a nationwide election. The concept of smart contracts were used in the proposed system to assure total validity for both voters and the election itself. The smart contract is divided into three sections to be covered: roles, election-related agreements, and transactions. The authors demonstrated that the system can be built using a variety of frameworks. However, the authors used Exonum, Quorum, and Geth to create a private network.

In another notable work, [7] proposed a Blockchain-based voting system, called BroncoVote, that preserves voter privacy and increases accessibility, while keeping the voting system transparent, secure, and cost-effective. The implementation of the proposed system was deployed on Ethereum's Testnet to demonstrate usability, scalability, and efficiency. Blockchain technology has many features, which gain its reputation below are some of them [11] [13]:

1) Decentralization: This means that all the nodes are sharing the same information, which is distributed in a digital ledger, without the control of a third party.

2) Immutable: This properly the most important feature, as the transactions are stored in the ledger chain can't be altered easily. Once the transaction is committed and push to the chain it will not change unless all the nodes agree, and it will take a huge amount of processing since it deals with the hash function as a secure chain.

3) Distributed: All the nodes in the network are sharing the same copy of the ledger.

4) Secure: BC is secure due to fact that it uses a hash function, encryption, and decryption tools [15].

5) Open Source: Anyone can get source code and try to modify it to come up with a new thing and then publish it to the world and get the advance from it. The BC can be used in different areas of our life, below are some applications that are getting Blockchain in it:

1) Payment Systems: perhaps this sector is based totally on the Blockchain, the first digital currency coins appear and were the evaluation.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2) Voting: We all have seen the US election scandal between Trump and the Russian government. Therefore, if Blockchain is used in the voting process then none of that will happen. As each node will be given specific details to make the election process works in the right direction, where a special algorithm can be used without the human get involved in the process.

3) Pharmaceutical Industry: Everyone knows how this sector is suffering from many issues like drug counterfeiting etc. therefore, BC is widely used to trace the process of manufacturing drugs [11, 12].

3. THE PRPOSED BLOCKCHAIN-BASED SYSTEM

Jordan's election is one of the most significant political events that take place every four years. Because of the time and money required, governments and the Independent Election Commission, as well as people, face a significant cost and time burden. Fraud, the lack of confidence, the lack of security, and the lack of privacy all contribute to the need to overhaul the existing system. Traditional methods are still ineffectual in meeting the needs of the people. This section provides and discusses the suggested BC-based approach for Jordan E-voting system.

3.1 Voting System Life Cycle in Jordan

The election process is separated into many parts, each of which is linked to the one before it. The steps below show how the process works, and figure 1 is used to make it easier to understand:

1. The electoral process starts with the announcement of the polling date, followed by the publication of a royal will to hold the elections.
2. All eligible persons are included in the people sheets tables based on their personal information and verifiable places of residency.
3. The nomination procedure for membership lasts three days, according to the eligibility conditions for candidacy.
4. The results are computed, seats are distributed, and the names of the victors for each electoral district are announced during the polling and sorting phase.
5. Following polling and tabulation, the results will be announced.

Each step, from the second to the last, demands some action with a large number of stakeholders who have a connected directory for election preparation. Our system will keep track of these steps and log them without missing anything or enabling a third party to change them. The voters and candidates, as well as the system data components, are all stored in an immutable, decentralized, and distributed ledger.

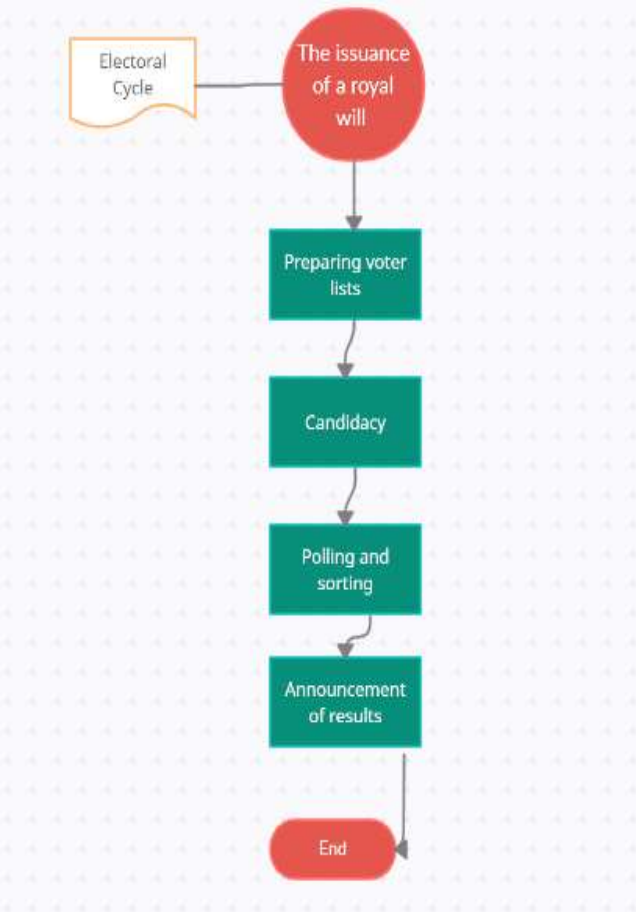

Figure 1: Voting life cycle.

3.2 System Architecture of Proposed System

The proposed system seeks to discover a solution to ensure that elections are conducted in a fair and transparent manner. Its fundamental goal is to produce a fair and unbiased result. We can have elections without worrying about the outcome because of Blockchain technology. Our suggested framework is depicted in figure 2. There will be six main components of the system: Stakeholders, Frontend, and Backend, Hyper ledger Blockchain, Database, and Consensus Algorithm [1]. The proposed system was conducted to provide a solution to help in Jordan election as the main goal is to protect the election as a process and for people, so using Blockchian was the main key to help.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Below are the main steps that used as a flow for the proposed system to be present:
1. Understanding Blockchain features and types to be selected.
2. Identify the system stakeholders and investigate each entity's role, responsibilities, and information that is critical to the system and that this stakeholder can supply. Then link each stakeholder to the transactions that are relevant to them.
3. Explain the significance of Blockchain as a solution for reducing election issues, as well as the appropriate kind for the actual situation, taking into account the nature of the environment.
4. Propose a new framework and put it in place with an appropriate consensus process and the necessary validation to develop a complete, secure, and trustworthy solution that satisfies the demands of the election sector.
5. Use a secure consensus algorithm.
6. Summarize the experiment's findings in order to assess the suggested system's criteria.

The stakeholders vote using the client-side app; voters and candidates are the main stakeholders in our system; their roles are limited because they only vote; the front end for them is implemented as a UI that will be developed in REACT.JS to be able to be handled by the backend Node.JS server, which will be in charge of all responsibilities. Stakeholders may engage with the application using any device by entering the URL in the search box, logging in, and then voting, with the possibility to see the results for each area or all regions. To make data retrieval easier, these transactions are stored in the (BC) Database. The backend system will use a Hype ledger fabric SDK with Node.JS, an open-source, cross-platform JavaScript (JS) environment, to link the front-end with the back-end and handle all API calls. Node.JS will handle all API requests from voters and candidates, which will be sent to the server and then returned to interact with the Hype ledger network. We're using a Consortium Blockchain because we don't want unauthenticated people to be able to view or edit the data, and we want the results to be open to everyone who wants to see them. We can ensure that only authorized persons have access to data using Blockchain technology.

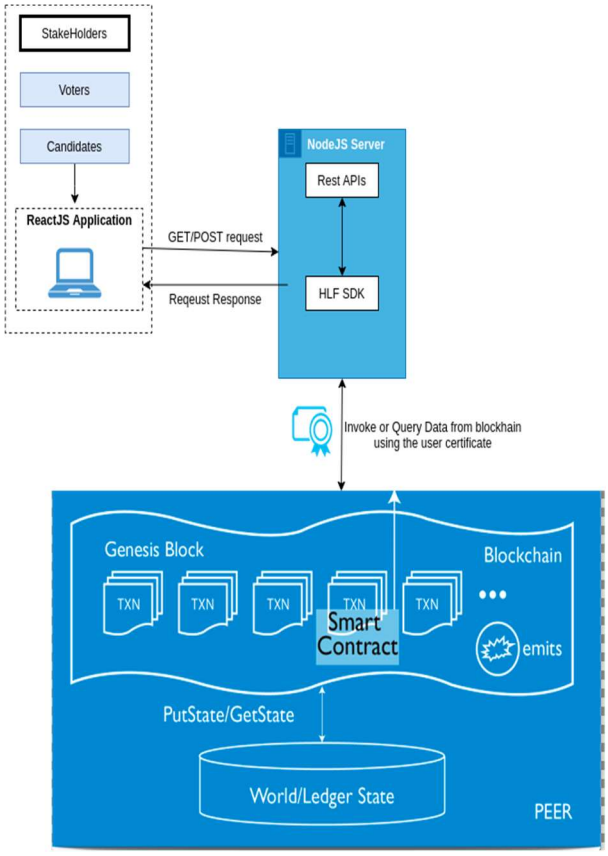

Figure 2: Architecture of the Proposed System.

3.3 Smart Contracts

In general, a smart contract provides the transaction logic that determines the lifespan of a business entity contained in the global state in executable code, and creates the rules between different organizations. Smart contracts are used by applications to generate transactions, which are subsequently recorded on the ledger. At its most basic level, a Blockchain records transactions that change the state of a ledger in an immutable way. A smart contract may programmatically access two portions of the ledger: a Blockchain, which immutably records the history of all transactions, and a world state, which stores a cache of the current value of these states, as the present worth of an item is generally required. Smart contracts may query the immutable Blockchain transaction record, as well as add, remove, and modify states in the global state. Figure 3 shows a snapshot of the programmed Smart Contract using Golang Programming Language.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
package main

import (
"bytes"
"encoding/json"
"strconv"
"strings"

"github.com/hyperledger/fabric-contract-api-go/contractapi"
area "github.com/hyperledger/fabric-samples/chaincode/voting-chaincode/entities/area"
user "github.com/hyperledger/fabric-samples/chaincode/voting-chaincode/entities/users"
vote "github.com/hyperledger/fabric-samples/chaincode/voting-chaincode/entities/vote"
voteevent "github.com/hyperledger/fabric-samples/chaincode/voting-chaincode/entities/voteevent"
)

type SimpleContract struct {
contractapi.Contract
}

func (sc *SimpleContract) AddArea(ctx contractapi.TransactionContextInterface) string {

_, params := ctx.GetStub().GetFunctionAndParameters()
if len(params) != 1 {
return `{"status": 400 , "message" :"Incorrect no of arguments .Please Provide 1 argument for adding a area."}`
}
Code := "AREA_" + params[0]
areaAsByte, _ := ctx.GetStub().GetState(Code)
if areaAsByte != nil {
return `{"status": 400 , "message": "area number already exists "}`

}

area := area.Area{
ObjectType: "area",
Number:     params[0],
}
areaAsBytes, _ := json.Marshal(area)
```

Figure 3: Smart Contract Transaction.

3.4 The Proposed Consensus Mechanism

Consensus algorithms enable a group of machines to work together as a cohesive entity that can survive the failure of some of its members. As a result, they're essential for building dependable large-scale software systems. To test the proposed system, we used a well-established consensus approach [7] [8].

The RAFT protocol, which is a replicated log management consensus technique, was designed by Diego Ongaro and John Ousterhout (both of Stanford University). It gives outcomes that are comparable to (multi-Paxos) and as efficient as Paxos, although it has a different structure than Paxos; this makes RAFT easier to grasp than Paxos and provides a better foundation for developing real-world systems. RAFT is based on the assumption that [n/2 + 1] of the total nodes are always operational. To perform the Raft consensus method, verifying nodes can choose one of three roles: follower, candidate, or leader. The two basic sorts of communications that these nodes communicate to each other are RequestVote for voting a leader node and AppendEntries for forwarding requests to other nodes. The important components of the agreement, such as the leader, are isolated using RAFT. The RAFT protocol is used in various works, and RAFT is a CFT ordering service based on it. RAFT uses a "leader and follower" paradigm, in which a leader node is chosen (per channel) and the followers copy the leader's decisions. Because its design allows several organizations to contribute nodes to a distributed ordering service, RAFT ordering services should be simple to set up and administer. Using the Hyper ledger, which has a more powerful and efficient RAFT version already. The proposed consensus algorithm will use peer endorsement and validation as a means of validation. The block's transactions will be ordered according to the ordering. Figure 4 displays the proposed Consensus algorithm workflow.

The steps below illustrate how to validate and add transactions to the chain [1]:

1. During consensus execution, the leader receives a significant number of requests for transactions and writes them to a log entry list. Who signs the smart contract and executes it, as well as constructing Read/Write (RW) sets.
2. Then the reader sends the AppendEntries message to all followers along with the RW sets, which contains each recorded transaction (r) and the previous transaction's index (pi) in the list.
3. When the follower receives the AppendEntries message, if (pi) is the latest transaction's index, he will write (r) to his log entry list. Otherwise, the leader will have to find the most recent transaction on which he and the contradicting follower agree, and then this follower will delete all transactions after the discovered transaction and resynchronize the log entry list with the leader. These methods are in place to guarantee that the transaction sequence is the same across all verifying nodes.
4. After confirming that all nodes' transaction lists are equal, the leader node chooses an index from the list, commit all previous transactions to this index, check the transactions (during the validate phase), and insert the valid ones into a block.

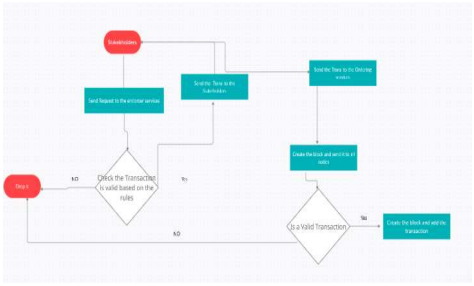

Figure 4: Consensus algorithm Workflow.

3.5 Couch DB

The world state is physically implemented as a database to allow for easy and effective storage and retrieval of ledger states. The ledger values can be simple or complex and the world state database design can adapt to accommodate this, allowing different values to be implemented efficiently. LevelDB and CouchDB are now available as global state database alternatives. When ledger states are structured as JSON documents, CouchDB is an

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ideal choice since it allows for more advanced searches and updates of richer data kinds that are frequently seen in business processes. Although there is still a one to one relation link between a peer node and a CouchDB instance, CouchDB is implemented in its own operating system process. Any of this does not affect a smart contract.

3.6 Transactions Flow

The proposed system's general transactional flow is described in this section. Figure 5 presents the transaction Validation Architecture Diagram. The voter begins a transaction, which is transmitted to peers A and B, who are users' representatives. According to the endorsement policy, each transaction must receive the permission of both peers. The transaction proposal is then put together. An application utilizing a compatible SDK (Node) utilizes one of the available APIs to generate a transaction proposal. To read and/or update the ledger, a chain-code function will be run with the given input parameters. The SDK works as a shim, packaging the transaction proposal into the appropriate architected format and providing a unique signature for it using the user's cryptographic credentials. The SDK works as a shim, packaging the transaction proposal into the appropriate architected format and providing a unique signature for it using the user's cryptographic credentials.

Peers who are endorsing each other check signatures and complete the transaction. The endorsing peers verify that (1) the transaction proposal is well-formed, (2) it hasn't been submitted before (replay-attack protection), (3) the signature is valid, and (4) the submitter (voter) is lawfully permitted to perform the requested operation on that channel (each endorsing peer verifies that the submitter complies with the channel's Writers policy). The transaction proposal inputs are provided by the endorsing peers to the called chain-function codes. The chain-code is then applied to the current form DB to provide transactional results, which include a response value, read set, and write set (i.e. key/value pairs that indicate if an asset should be created or edited). At this moment, there have been no changes to the ledger. The SDK gets the collection of this data, together with the signature of the endorsing peer, and parses the payload as a "proposal response" for consumption by the application.

Proposal responses are evaluated. The software checks to verify if the proposal replies are the same and verify the supporting peer IDs. The application will usually just examine the query response before transmitting the event to the ordering service if the chain-code is merely searching the ledger. The client application checks to verify if the declared endorsement policy has been fulfilled before submitting the transaction to the ordering service to update the ledger (i.e. did peer A and peer B both endorse). Even if an application chooses not to check responses or transmits an unendorsed transaction, peers will nevertheless adhere to the endorsement rules and enforce them.

The client assembles a deal by assembling endorsements. The application "broadcasts" the transaction proposal and response to the ordering service within a "transaction message." The transaction will comprise the read/write sets, signatures of endorsing peers, and the Channel ID. The ordering service does not need to review the entire content of a transaction to perform its function; instead, it simply accepts transactions from all channels in the network, arranges them chronologically by channel, and creates blocks of transactions per channel. The transaction has been committed and validated. The transaction blocks are "delivered" to all peers on the channel. The transactions in the block are checked to ensure that the endorsement policy is followed and that the ledger state for reading set variables has not changed since the transaction was executed. The transactions in the block are classified as valid or invalid. The ledger has been continually updated. The block is added to the channel's chain by each peer, and the write sets of each valid transaction are committed to the current state database. Each peer sends out an alert to the client application, informing it that the action (invocation) has been added to the chain in an immutable manner, as well as whether it has been validated or invalidated.

With all of the advantages and characteristics of Blockchain, there may be certain constraints that must be addressed in order to get an ideal solution, such as the cost of validators as we increase the number of validators. Also, illiteracy in the use of this technology, since it will be difficult for elderly and uneducated individuals to utilize or adapt it. In addition to official acceptances, as it may be difficult for governments to begin working with it.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

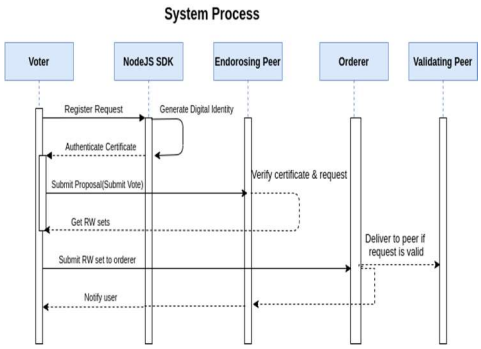

Figure 5: The transactions flow.

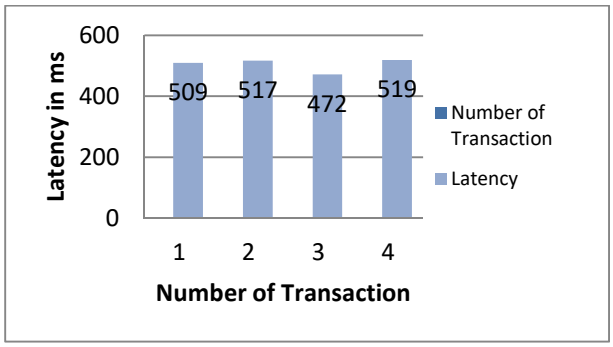

Figure 6: Time to reach consensus on transaction (Latency).

4. SIMULATION AND RESULTS

In this section, the simulation results of the proposed system are discussed in detail.

1. Simulation Environment

The proposed system was implemented using different tools Atom IDE v1.57.0 / Hyper Fabric v2.0.1/ Node JS v15.6.0 / Charles Web Debugging Proxy v4.6.2 / Postman v 8.1.0 / CouchDB v2.3.1/ GitHub for uploading the source code v3.2.0 / Gatling Performance Tool v3.3.1 / Grafana Monitoring Tool v6.7.5. The proposed system used a specialized dataset for voters and candidates provided by "independent election commission Jordan".

2. Results and Discussion

In this section, the overall performance of the proposed system is given. The system is tested and evaluated according to transaction latency, transaction throughput, Responses time needed for writing a transaction, scalability, and security and privacy.

4.2.1 Transaction Latency

Transaction latency is the amount of time takes for the network to reach a consensus. We used Postman to test the APIs that would perform the transaction with the help of the Charles application. The Lenovo E590 laptop was utilized for testing, and it runs Windows 10 with a Core i7 8th Gen CPU, 16 GB RAM, 64-bit operating system, and an x64-based processor. Figure 6 depicts the amount of time takes for a transaction to achieve consensus peers and then return. The timing disparities are attributable to the network utilized and the number of transactions included inside each Block, indicating that our results are acceptable.

4.2.2 Transaction Throughput

Throughput can be defined as the rate at which acceptable transactions are accepted per unit time. This rate applies to the whole transaction, i.e. it is committed to all network transactions rather than just one. Remove the total number of invalid transactions from the total transactions to get the total committed transactions. Figure 7 depicts the time required to accept legitimate transactions. The initial transaction is regarded as important since it will serve as the starting point for all subsequent transactions; yet, the outcomes are often satisfactory.

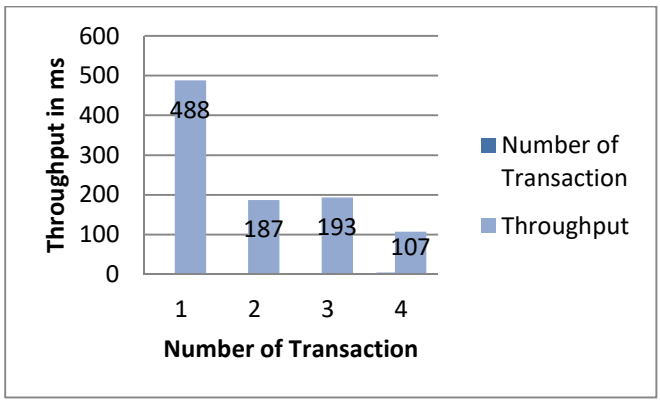

Figure 7: Time to accept valid transactions (Throughput).

4.2.3 Responses time needed for writing a transaction

The amount of time needed to write the transaction after authorization is shown in figure 8. For each request, whether it was a (Write/Read) method, the latency, throughput, and response time will be measured using the Charles/Gatling Performance tool from Scala. The results are satisfactory, and they demonstrate the quickness of the proposed system.

Journal of Theoretical and Applied Information Technology
15th March 2022. Vol.100. No 5
2022 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

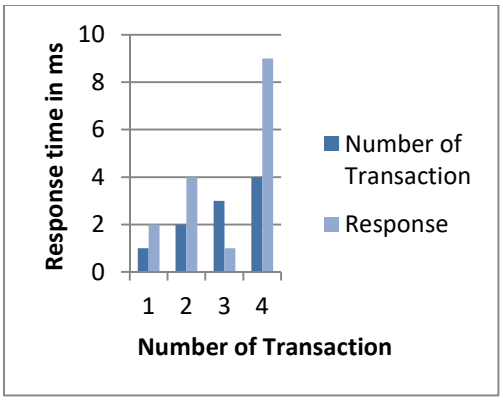

Figure 8: Time to add transactions (Response).

4.2.4 Scalability

Scalability of the system means that while the number of users increases no extra constraints on the system. Rise in number of stakeholders will only strengthen the system's resilience, because the proposed strategy splits these stakeholders into specific validator classes. To analyze performance, we used the Gatling Performance tool and the Grafana Monitoring tool to generate a UI view of our own Dashboard. Therefore, we used 280 virtual users, each of whom needed 4 seconds to reach the Steady-state before becoming active users, giving them an hour to work before ramping down to finish the test. The results of the Gatling test are shown in figures 9, 10, and 11.

Figure 9: Numbers of Active Users.

Figure 10: Grafana Monitoring tool.

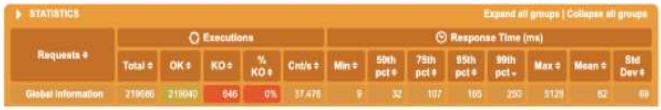

Figure 11: Global Statistics.

4.2.5 Security and Privacy Analysis

Privacy and Confidentiality: The use of Digital Certificates (DCs) for network nodes, which are given to each node in the network for log in and using Universally Unique Identifiers (UUIDs) for each transaction protects the network nodes true identities from non-network members and ensuring that they are not disclosed to the public. Because the transaction can only be added to the chain after it is validated. Access may be limited to some sensitive transaction types and only allow stakeholders to share such data because this is a Hybrid Blockchain.

Integrity: Each block contains a hash of its content to safeguard its integrity. Each block replaces the previous hash with a new one to confirm that nothing has been tampered with. In the proposed system, the Hash function SHA256 was used to hash the block content and then added the hash result to the block for hashing.

Availability: Due to the distributed nature of Blockchain and the hybrid type used in the proposed system, each node may serve clients even if other nodes fail or become unavailable. We scan all Blockchain blocks and validate each transaction in each block to verify the chain and check for consensus rules; if any rule is breached, the chain will be invalidated. Table 1 present different studies with compered with the proposed system.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1. Differences Between Prior Approaches And The Proposed System

Paper number	Difference between the previous proposal and our system
[4]	The small scale from their side was an issue, in our system we have a good scalability and this can applied to any voting system in the world.
[15]	The centralizing sources which they rely on is not existing in our system the admin only add voters, candidates, and create the election day without interfering in the voters or the results.
[17]	The proposed system lacks the performance latency which was higher, our system is performing well as the time it needs to performed a transaction is less
[18]	They have only provided a view of their idea without any implementation, in our case we have provide a new framework with more features.

The proposed system shows that is can be applied in the Jordan election due to the outcome results as it will reduce the money, time since Blockchain will handle the all the operations without the need to have third part to interfere. Also, the proposed system will allow people to vote in any location without having to be present in the current location. Using the proposed system will reduce election costs by 90%, as there will be no need to do all of the election preparation work that was previously required when voting by paper and pin. Because there is a need to apply such a system to eliminate concerns, and Jordan has been seeking to create new methods to advance the digital world to generate technology that matches its needs, this study was exclusively applied to the Jordan election.

5. CONCLUSION

In this study, a Hybrid Block chain was used with a modified consensus algorithm called RAFT to create a secure and reliable voting system and avoid fraud during the election process. The system presented a collection of transactions, each with its own set of stakeholders to reflect the election's actions and occurrences. The experiments show the usability and efficiency of the implemented system in terms of confidentiality, data integrity, privacy, and data privacy. Furthermore, the suggested system is efficient in terms of the time it takes to validate and append transactions to blocks.

REFERENCES:

[1] Sarah Al-Maaitah, Mohammad Qatawneh, Abdullah Quzmar. E-Voting System Based on Blockchain Technology: A Survey. 2021 International Conference on Information Technology (ICIT).

[2] Mohammad Qatawneh, Wesam Almobaideen, Orieb AbuAlghanam (2020). Challenges of Blockchain Technology in Context Internet of Things: A Survey. International Journal of Computer Applications, Vol 175 (16).

[3] Abdullah Quzmar, Mohammad Qatawneh, Sarah Al-Maaitah. Reducing Counterfeit Drugs with Blockchains: A Survey. 2021 International Conference on Information Technology (ICIT).

[4] Ahmad Afif Monrat, Olov Schelen, and Karl Andersson(August 19, 2019). A Survey of Blockchain from the Perspectives of Applications, Challenges, and Opportunities. Digital Object Identifier 10.1109/ACCESS.2019.2936094.

[5] Cosmas Krisna Adiputra, Rikard Hjort, and Hiroyuki Sato (2018). A Proposal of Blockchain-based Electronic Voting System. Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4).

[6] Friðrik Þ. Hjálmarsson, Gunnlaugur K. Hreiðarsson, Mohammad Hamdaqa, and Gísli Hjálmtýsson (2018). Blockchain-Based E-Voting System. In the 11th International Conference on Cloud Computing IEEE.

[7] Dagher, Gaby G, Marella, Praneeth Babu, Milojkovic, Matea, and Mohler Jordan (2018, January). BroncoVote: Secure Voting System Using Ethereum's Blockchain. In the 4th International Conference on Information Systems Security and Privacy (ICISSP), 96-107.

[8] Sheping Zhai, Yuanyuan Yang, Jing Li1, Cheng Qiu1 and Jiangming Zhao. (2018). Research on the Application of Cryptography on the Blockchain. IOP Conf. Series: Journal of Physics: Conf. Series1168 (2019) 032077.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[9] Baker Alhasan, Mohammad Qatawneh, Wesam Almobaideen. Blockchain Technology for Preventing Counterfeit in Health Insurance. 2021 International Conference on Information Technology (ICIT).

[10] Ongaro, Diego, and John Ousterhout.(2013). In search of an understandable consensus algorithm (extended version). Annual Technical Conference.

[11] Sara El-Switi, Mohammad Qatawneh. Application of Blockchain Technology in Used Vehicle Market: A Review. 2021 International Conference on Information Technology (ICIT).

[12] Mais Haj Qasem, Mohammad Qatawneh. Parallel Hill Cipher Encryption Algorithm. International Journal of Computer Applications, 179(19), 2018.

[13] Orieb Abualghanam, Mohammad Qatawneh, Wesam Almobaideen (2019). A Survey of Key Distribution in the Context of Internet of Things. Journal of Theoretical and Applied Information Technology, Vol 97(22).

[14] Sabo Ahmada, Siti Alida John Bt Abdullahb and Rozita Bt Arshadc. (2015). Issues and Challenges of Transition to e-Voting Technology in Nigeria. Public Policy and Administration Research gISSN 2224-5731(Paper) ISSN 2225-0972(Online) Vol.5, No.4, 2015

[15] Ahmad Bany Doumi, Mohammad Qatawneh. Performance Evaluation of Parallel International Data Encryption Algorithm on IMAN1 Super Computer. International Journal of Network Security & Its Applications (IJNSA), 11(1), 2019.

[16] Heba Harahsheh, Mohammad Qatawneh. Performance Evaluation of Twofish Algorithm on IMAN1 Supercomputer. International Journal of Computer Applications, 179(50), 2018.

[17]Aneta Poniszewska-Marańda, Michał Pawlak and Jakub Guziur (2020. Auditable blockchain voting system – the blockchain technology toward the electronic voting process. Int. J. Web and Grid Services, Vol. 16, No. 1, 2020.

[18] Asassfeh, M. R., Qatawneh, M., & AL-Azzeh, F. M. (2018). Performance evaluation of blowfish algorithm on supercomputer iman1. International Journal of Computer Networks & Communications (IJCNC), 10(2).