Engineering and Technology Journal
Journal homepage: https://etj.uotechnology.edu.iq

E-voting System Based on Ethereum Blockchain Technology Using Ganache and Remix Environments

Hind S. Hassan*, Rehab Hassan, Ekhlas K. Gbashi

Computer Science Dept., University of Technology-Iraq, Alsina'a street, 10066 Baghdad, Iraq.
*Corresponding author Email: cs.19.76@grad.uotechnology.edu.iq

HIGHLIGHTS

- A smart contract-based blockchain-based e-election system that operates on Ethereum and provides a set of rules for communication and contract decision-making among participants.
- Blockchain technology has the potential to move beyond the constraints of centralized voting systems, demonstrating that it is not only quick and cheap but also safe and secure.
- Ganache, the Truffle framework, NPM, metamask, and Remix ide were among the technologies utilized in this project.

ARTICLE INFO

Handling editor: Rana F. Ghani

Keywords:
Blockchain; Ethereum; E-Voting; Truffle; Ganache; Remix.

ABSTRACT

Abstract—In the second decade of the twenty-first century, blockchain is considered one of the most popular computer technologies. Blockchain is a zero-trust network, making it a potent tool for various services provided that people are ready to believe and invest in it. In the Ethereum world, the blockchain runs on smart contracts, self-executing applications that come at the cost of security. As technology is used increasingly, Election is facing new issues of trust and management. Therefore, E-voting systems are increasingly acceptable because they are more accurate, reliable, practical, and secure. This research purpose is to propose a decentralized elections application based on Ethereum; the application was developed using the Truffle development framework. The actions of the software were written into an Ethereum smart contract, which was then deployed on the Ethereum network. A web interface had been used to read the user's vote before it was broadcast to the Ethereum network through the web3.js API. The ganache was utilized as the Ethereum client. Metalmark was used as a wallet on a website, and the remix was used to deploy the smart contract on the main network, the results of implementing the proposed system show that the cost of each transaction is not stable, its increases with the increase the network load, and the throughput ends up at 14 transactions per second.

1. Introduction

In contemporary society, e-voting systems are becoming more popular. It has a considerable chance of lowering administrative expenses as well as raising the rates of participation. In the case when utilized in elections, e-voting systems need to be accurate, reliable, practical, and secure [1]. Even though numerous electronic voting has become popular in several nations, there is no acceptable, trustworthy, or efficient system for people in the national elections since it necessitates some contradictory qualities. For authentication and privacy protection, the election requires one or more authorities, but a voter cannot trust a government or authority that will never be hacked and always respect the law [2].

Blockchain technology is an emerging field, but it has a solid cryptographic base, allowing applications to benefit from such capabilities and produce adaptable security solutions. Satoshi Nakamoto developed the blockchain in 2008 as a global ledger for the Bitcoin cryptocurrency [3]. It is a distributed decentralized database that keeps an exhaustive list of continuously expanding and growing data records made secure against unauthorized tampering [4].

The main aim of this research is to design and implement a proposed electronic voting system based on blockchain technology to achieve the principle of security that is capable of offering fairness and privacy, providing transparency, and preventing fraud and manipulation. Ethereum blockchain is one of the most popular types of blockchain that relies on the proof of work protocol. Our contributions were implementing the electronic voting system using Ethereum smart contract in the truffle environment and using the ganache environment as a local blockchain, in addition to implementing the smart contract in

https://doi.org/10.30684/etj.2023.135464.1273
Received 14 December 2022; Accepted 11 February 2023; Available online 3 March 2023
2412-0758/University of Technology-Iraq, Baghdad, Iraq

the remix environment in order to deploy our smart contract in the main blockchain, General Structure of the Research consist of the following topics:

1) Section one: The research starts by explaining the general overview of electronic voting systems based on Ethereum blockchain technology, remembering the aim of the research, and explaining our contribution.
2) Section two: in this section, we reviewed some of the related work and make a comparison of them depending on some criteria.
3) Section three: this section explains the significant problems in the traditional E-voting systems.
4) Section Four: In this section, we explained the reasons for using blockchain technology to build an electronic voting system.
5) Section five: This section explains the Ethereum Blockchain, its structure, Structure and its mining.
6) Section six: Explain the implementation of the proposed system.
7) Sections seven and eight: These sections explain the measurement, results, and discussion.

2. Literature Review

There are many previous studies based on blockchain technology in electronic voting systems, and here is a review of some of them, Table 1, shows a comparison among them:

Table 1: A Comparison of the Related Works

Research Name	Type of Transaction	Blockchain Type	Consensus Protocol	Robustness	transparency	Scalable	Speed	Privacy	Low Cost	
An End-to-end Voting-system Based upon Bitcoin	Permissionless, public	bitcoin	Proof of work	√	√	×	×	√	×	[1]
Internet Voting Using Zcash	Public, private	Zcash	Proof of work	√	√	×	×	√	×	[2]
Trustworthy Electronic Voting Using Adjusted Blockchain Technology	Public, private	consortium blockchain	Can be Implemented in various protocols	√	√	×	Private better than public	√	×	[3]
A Modernized Voting System Using Fuzzy Logic and Blockchain Technology	public	blockchain	unknown	√	√	×	×	√	unknown	[4]

2.1 An End-to-end Voting-system Based upon Bitcoin

(Bistarelli, and et. al.,) [5]. The significant idea of the study is that voting data can be stored on a blockchain and linked to a Bitcoin token. All voters will be made aware of the addresses of the nominated candidates. The token is sent to the appropriate candidate's Bitcoin address to cast a vote. The number of tokens sent to the associated Bitcoin address determines how many votes the candidate receives.

2.2 Internet Voting Using Zcash

(Pavel Tarasov & Hitesh Tewari,) [6]. This study employed Zcash technology, a decentralized blockchain payment seeking anonymity for transactions. Zcash depends on zero-knowledge proofs. It makes the transaction data public. When a voter clicks on the ballot link, they are taken to the ballot page, where they must enter a receiving t-address and a z-address to send their vote. The voter uses a z-address to secure the anonymity of their vote. If the candidate utilizes a z-address, the transaction between the voter and the candidate is private. The vote count and audit, which take place after the count to assess the election process and confirm that the integrity of the election has not been compromised, are the last stages of the voting procedure.

2.3 Trustworthy Electronic Voting Using Adjusted Blockchain Technology

(Basit Shahza & Jon Crowcroft,)[7].The authors of this study proposed a system using efficient hashing techniques to ensure that the data is secure. This work presented the concept of block creation and block sealing. Implementing a block-sealing idea made the blockchain adaptable to the voting process requirements. The authors advised adopting a consortium blockchain, which guarantees that the blockchain is held by a government board and cannot be accessed by other parties without authorization.

2.4 A Modernized Voting System Using Fuzzy Logic and Blockchain Technology

(Mousumi Mitra & Aviroop Chowdhury,) [8]. This study proposed a new idea that used weighted logic to determine the number of points that must be assigned to each candidate. Every voter and candidate would have a digital wallet. After the vote is cast and the points are decided, the points are used to programmatically transfer coins from the wallet of the voters to the wallet of the candidates. There would be one transaction for every chosen candidate. Each candidate would receive an equivalent number of coins based on the fuzzy logic-derived points.

3. The Problems Statements of The Traditional E-Voting Systems

This research addresses the problems and limitations of the traditional voting systems, including:

1) Security Attack: Elections always need a high level of security to safeguard voter privacy and the fairness of the process.
2) Lack of confidence in elections may be depressing voter turnout.
3) Vote-buying, when a candidate or a political party tries to buy a voter's vote in an upcoming election, makes the voter open to exploitation and pressure.
4) Lack of transparency is fundamental to a successful voting system.

4. Motivation for Using Blockchain Technology

Blockchain technology is fundamentally a decentralized system, which eliminates the need to deal with a third-party organization or a central administrator.

The system database may be protected using Blockchain technology since each transaction has its evidence of validity and authority to impose limitations.

Every action is recorded on the blockchain, and the data included in the records are available to all Blockchain participants and cannot be changed or withdrawn.

The blockchain's immutability is achieved by agreeing on and sharing transactions. Once a transaction is connected to the blockchain, it will be difficult to change or delete it [9,10].

5. Ethereum Blockchain

The Ethereum effort started in 2014 and is now the second most well-known blockchain project after Bitcoin. It was described in Vitalik Buterin's paper and addressed various constraints of Bitcoin's programming language. The key advantages over blockchain structure include full Turing-completeness, which means that Ethereum supports all forms of computations, such as loops, then supports the state of the transaction [11]. With Ethereum, developers can create random consensus-based applications with standardization, ease of development, feature completeness, and interoperability offered via various models. Ethereum aims to combine and enhance the concepts of scripting, altcoins, and on-chain meta-protocols. It achieves this by building a blockchain with a built-in Turing-complete programming language that anyone could use for creating smart contracts and decentralized applications with their own random ownership rules [12]. There are two main Ethereum accounts, which are: the Externally Owned Account (EOA), which allows a user to send a transaction with other users in a direct manner, and the Contract Account (i.e., the Smart Contract), which enables the user to send internal transactions according to the contract's run code [13].

5.1 Ethereum Virtual Machine (EVM)

A programmable blockchain exists in Ethereum. Instead of providing the users with a group of predefined operations (such as the bitcoin transactions), Ethereum lets the user design his procedures regarding any level of complexity. Acting that way provides a platform for a wide range of decentralized blockchain applications [14]. Each of the opcodes has a particular cost (gas) based upon the necessity of the instruction's resources, as well as fees paid to the miners. Executing such instructions on the EVM requires a certain gas amount, and execution will be terminated when the gas ends. To run smart contracts and pay for the computational cost, like CPU and energy, the user of EVM fills gas with Ether [15].

5.2 Smart contract

A contract has been designed to carry out certain tasks. Nick Szabo coined the term "smart contract" in 1994, and it refers to a piece of software that performs a series of tasks in a blockchain system using the consensus protocol. A smart contract is a computer program written in a high–level language like Solidity that may be used in a variety of industries to eliminate third–party transactions and automate processes [16,17].

5.3 A decentralized application (DApp)

is software that operates on a peer-to-peer network of computers without using a central computer to send or receive data. Decentralized apps use blockchain and smart contracts to enforce their agreements between parties [12] .

5.4 Ethereum Blockchain Structure

We discover that a block B has transactions T, a header H, and Ommer block, sometimes known as uncles U: $B = (H,T,U)$ Ethereum blockchain structure is visualized in Figure 1, as it may be seen, the block header links various tries, and the body contains the transitions.

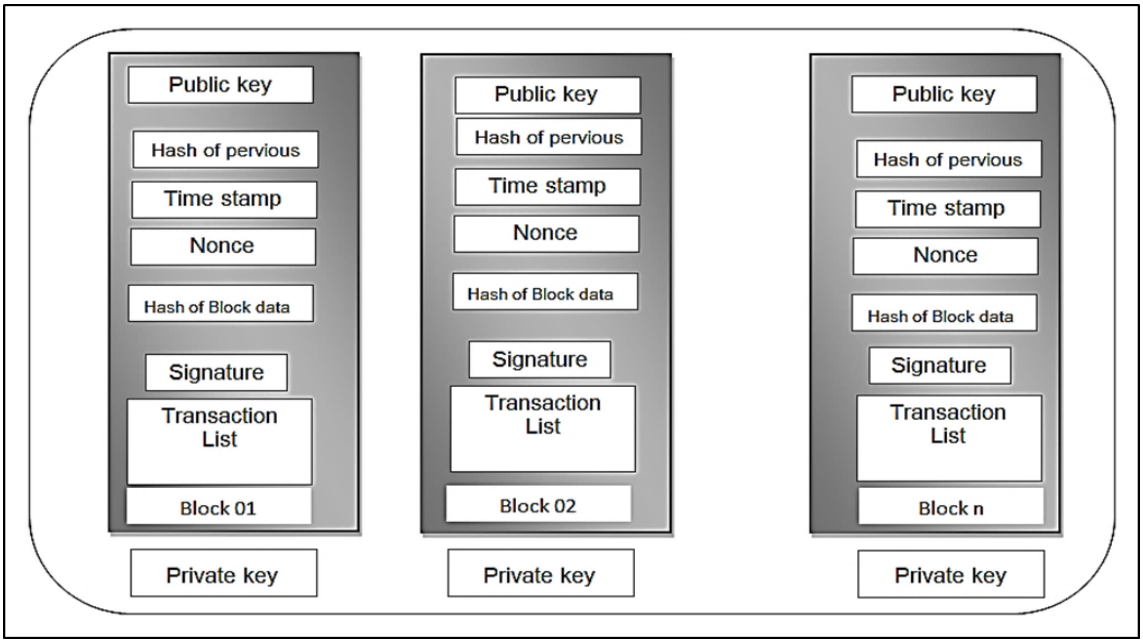

Figure 1: Ethereum Blockchain Structure

5.4.1 Header *H*:

5.4.1.1 The complete list of headers *H* fields are [18]

1) ParentHash: Parent block's Keccak 256-bit hash is contained in this header. Such a field connects a chain of blocks.
2) OmmersHash: This header is the list of Ommers (Uncles) blocks contained in the block's Keccak 256-bit hash.
3) Beneficiary: This header, which comprises a 160-bit address, is the account address of the user who mined this block and received a reward.
4) StateRoot: The root node of the state trie's Keccak 256-bit hash is contained in this header. It is determined following the processing and completion of all transactions.
5) ReceiptsRoot: The Transaction Receipt Trie root's Keccak 256-bit hash is contained in this header.
6) TransactionsRoot: The root node regarding the transaction trie's Keccak 256-bit hash is contained in this header. The list of the transactions that are included in the block is represented by the transaction trie.
7) Receipts root: This header contains the Keccak 256-bit hash regarding the transaction receipt trie's root node. All of the receipts from the transactions in the block are collected in this trie. After each transaction is performed, transaction receipts are created, and they include helpful post-transaction data. The logger address and log topics from each transaction receipt's log entry in the block's included transaction list make up the logs bloom filter.
8) Difficulty: The current block's difficulty is indicated by this header.
9) Number: This block's ordinal number is represented by the scalar value of this header. Each new block adds one to the chain's total number.
10) GasLimit: This header is the scalar value containing an accumulated gas limit required to process all transactions in this block.
11) GasUsed: This header is a scalar value that contains the total amount of real gas utilized to perform the transactions in this block.
12) Timestamp: This header contains the block initialization time in UNIX epoch time.
13) Extra data: Any data pertaining to the block may be stored in this header.
14) Mixhash field: When paired with the nonce, this header's 256-bit hash can be utilized to demonstrate that enough computational work was done to create this block.

5.4.2 Nonce

In conjunction with the mixedhash field, this header's 64-bit hash (a number) is utilized to demonstrate that sufficient computational effort was expended to construct this block.

5.5 Ethereum Transaction

The life cycle of a transaction in the Ethereum network is depicted in Figure 2. Users must first log onto an Ethereum account to receive and send transactions. All transactions are collected in the Mempool at this point. Then, to mine a block, a

miner chooses a transaction from the Mempool and decides whether to reject or confirm it. When miners successfully validate a block, they add it to the blockchain and update the chain across all Ethereum network nodes [19].

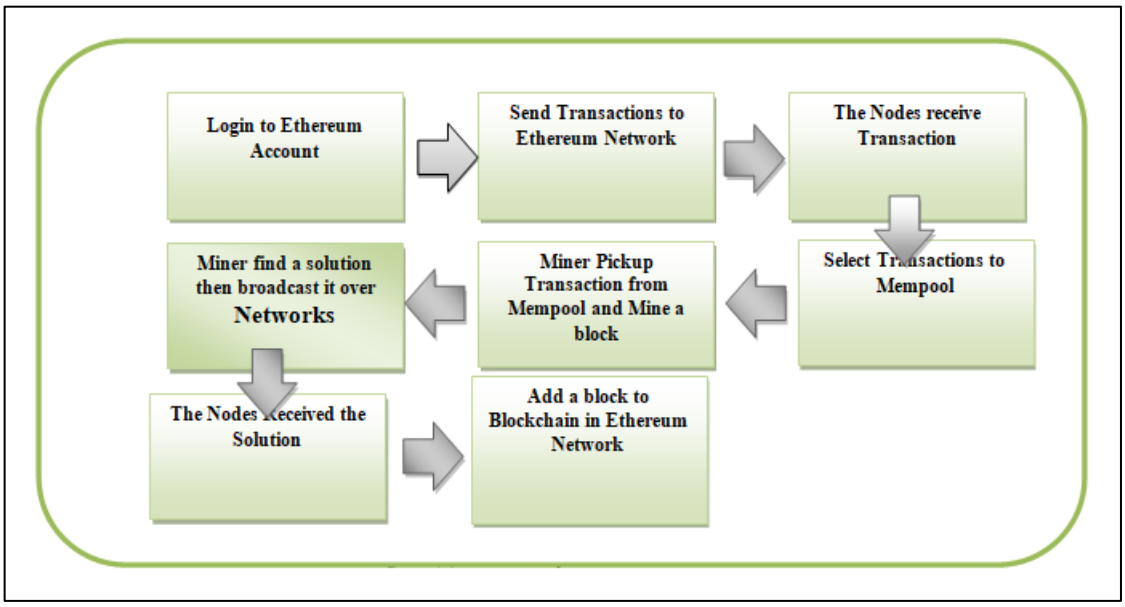

Figure 2: The Lifecycle of the Ethereum Transaction

5.6 Ethereum Blockchain Mining

Even though there are a few distinctions, the Bitcoin and Ethereum blockchains are comparable in various aspects. The primary difference between Bitcoin and Ethereum's blockchain architecture is that, in contrast to Bitcoin, Ethereum blocks include copies of the most current state and transaction list [12]. The following describes the fundamental Ethereum block validation algorithm, as shown in Figure 3.

1) Check that the previous block referenced is both valid and present.
2) Make sure the timestamp is less than 15 minutes in the future, higher than the timestamp regarding the referenced previous block.
3) Check the validity of difficulty, block number, uncle root, transaction root, and gas limit (as well as other valid Ethereum-specific concepts).
4) Check that a block's PoW is valid.
5) Let S[0] be the state at the previous block's end.
6) Assuming that TX represents the transaction list of the block, with n transactions. For each i in 0...n-1, set S[i+1] = APPLY(S[i],TX[i]). Return an error when any applications return an error or if the block's total gas consumption up to this point is greater than the GASLIMIT.
7) Assume that S_FINAL is S[n]; however, the addition of the block reward that has been paid to the miner.
8) I was checking that the final state root specified in a block header matches the Merkle tree root regarding state S_FINAL. If so, the block is valid; if not, it isn't valid. Since the state is stored in the tree structure, a small portion of the tree must be modified after each block. Because almost all trees must be the same between two adjacent blocks data could be stored once and referenced twice via pointers (hashes of subtrees). To do this, a unique sort of tree referred to as a "Patricia tree" has been utilized. Along with a tweak to the Merkle tree idea that makes it possible for nodes to be added and removed, rather than updated, efficiently [20].

6. Implementation of The Proposed System

6.1 Design Considerations

When designing the system, the following points were taken into consideration

- Illegal candidates should not be permitted to use the electronic voting system.
- A voter should only be permitted to vote once, and the system should prevent voters from voting multiple times.
- It should provide the greatest privacy to voters while also ensuring that their votes are not tracked.
- No one should be able to influence the outcome of the vote.
- No single authority should be able to control counting in the system.

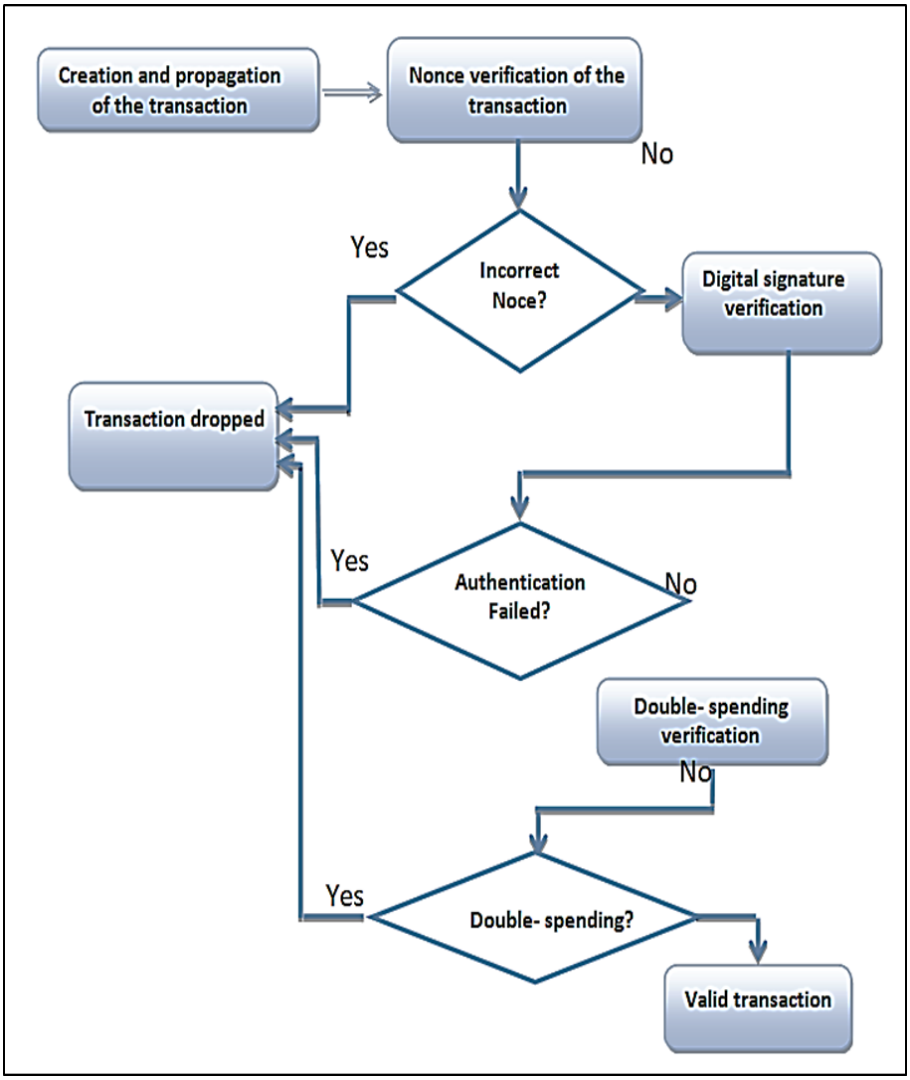

Figure 3: Transaction Verification Process Flowchart

6.2 Developing Tools

To implement this decentralized application, the following tools are required:

6.2.1 NPM

With no less than 1.7 million packages, the Node.js Package Manager (also known as npm) supports one of the largest developer ecosystems in the world and it plays a crucial part in the JavaScript community [21]. With each package frequently depending on many others, such packages give developers significant libraries and features without them having to "reinvent the wheel" [22]. NPM's most distinctive feature is that it enables users to not just install multiple versions regarding a package, yet also utilize multiple versions. throughout one execution run [23].

6.2.2 Node.js

Node.js. has been created by R. Dahl in the year 2009, and sponsored by joyent, which is the company Dahl was working for. Basically, Node.js uses Google V8 2 engine to run JavaScript code on the server side [24]. One recent JavaScript technology is represented by Node.js. It creates scalable, quick network applications. Node.js is effective and lightweight, which makes it ideal for real-time data-intensive applications which operate across distributed devices. It utilizes an event-driven, non-blocking I/O approach. Corporations are rapidly realizing the significance of Node.js and 5 main PAAS providers have supported Node.js.[25]. With the use of the NPM package repository's thousands of modules, Node.js provides a convenient trade-off between application performance and developer productivity, enabling developers to create apps rapidly and ready for production [26].

6.2.3 Web3.js (Ethereum JavaScript API)

With the use of HTTP, WebSocket, or IPC, you can communicate with a remote or local Ethereum node utilizing the web3.js library collection.

6.2.4 Ganache

A local development blockchain called Ganache is employed to simulate the actions of a public blockchain, smart contracts are released using Ganache, which is also utilized to execute testing. To test the smart contracts on the local bases of the blockchain, Ganache provides ten accounts with 100 Ether [27].

6.2.5 Truffle framework

For use with Ethereum smart contracts, a truffle is a potent tool. A command-line program called truffle has a built-in smart contract compiler. It serves as a platform for testing automated contracts, and administers networks, and packages in addition to being utilized for the compilation, deployment, and linking of smart contracts [28 , 29].

6.2.6 Solidity

Solidity can be defined as a contract-oriented, high-level language that can be utilized for the implementation of smart contracts. A high-level, statically typed programming language called Solidity is Turing-complete. It is intended to target the EVM and was influenced by C++, JavaScript, and Python [30]. In addition to supporting libraries, inheritance, and sophisticated user-defined types, Solidity is statically typed. Solidity enables the creation of contracts for various purposes, which include crowdfunding, blind auctions, voting, multi-signature wallets, and more [31].

6.2.7 Metamask

With a GUI, Metamask can be defined as a user-friendly, open-source, solution for Ethereum transactions. The framework browser could be used to run Ethereal Dapps without a full Ethereum hub. In essence, Metamask acts as a bridge between a browser and Ethereum blockchain [32].

6.2.8 Remix IDE

For JavaScript-based smart contracts, Remix IDE (Integrated Development Environment) represents a well-liked browser-based IDE.

6.3 E-voting System Architecture

Building the system consist of creating a smart contract, deploying them on the Ethereum network, and `constructing a voting client-side application. We have an HTML, CSS, and JavaScript front-end client. Rather than using a back-end server, the client will connect to an Ethereum blockchain. Depp's code was written in Solidity.

1) Ganache Initialization :first, start the Ganache to start the blockchain
2) Create the smart contract: The responsibility for reading from and writing to the Ethereum blockchain will fall to this smart contract. It will enable us to keep track of all votes and voters, as well as a list of candidates running in the election. It will also administer all election regulations, such as the requirement that accounts only vote once.
3) Test Voting: using the Mocha testing framework and the Chai assertion library to write our tests in JavaScript. The test was created for two purposes:
   - Determines whether the contract was formed with the appropriate number of applications.
   - Ensure each person has the appropriate identification, name, and vote total. Also, ensure that our function prohibits a vote from being cast twice.
4) Deploy the smart contract: make the smart contract available to users of an Ethereum network, transmitting an Ethereum transaction containing the smart contract's-built code to no one in particular.
5) migrate the smart contract: Migrations are JavaScript files that assist in the deployment of smart contracts on the Ethereum network.
6) To vote, users must pay a small fee in the form of gas. This gas fee can be made in a variety of ways. Metamask is one of the methods we will employ here. It is a Google Chrome or Mozilla Firefox extension. Metamask enables us to access the decentralized blockchain via our browser and execute Ethereum contracts without having to run an entire Ethereum node. To log into Metamark, we must select one of the ten accounts offered by ganache.

The transaction execution steps can explain in the following steps as shown in Figure 4:

1) the user selects the candidate and clicks Vote on the HTML page.
2) The vote click event is handled by a JavaScript component. After that establishing a connection with a node of the Ethereum network using the Web3.js module of Ethereum communication creates a proxy to the Voting smart contract and runs its castvote() function while passing chosen candidate along. Since a function call modifies the voting Dapp's state and necessitates a digital signature that is issued at the time of the call, it has been known as a "transaction."
3) "Local" Ethereum node that has validated voting transaction verifies it. The validation transaction entails verifying several factors, including that sender has enough ether for covering the whole cost of executing the transaction, that the digital signature matches the address of the sender, and that castVote() function

won't fail due to the submitted data. The current node publishes the transaction to each one of its peer nodes if validation is successful (which are nodes it is connected to). If the validation fails, the transaction just disappears and is not published again.

4) The transaction finally reaches multiple "mining nodes" after being correctly validated. A mining node represents a unique type of node that actively processes incoming transactions hoping to receive a reward rather than just validating them.
5) A cryptographic puzzle called Proof of Work (PoW) utilizes the newly constructed block as input. The mining node has the right to append the new block to the blockchain and then claim the reward for the job that has been completed in the form of a set amount of ether, the cryptocurrency used by Ethereum if it solves the puzzle.
6) A mining node announces a new block to its peer nodes as soon as it successfully appends it to the blockchain. The block will be verified by every one of the receiving nodes. After processing transactions in the block, it will verify that the transaction root hash and state tree root hash that had been reported on the block correspond to the matching local transaction Patricia-Merkle trie and local Dapp state Patricia-Merkle trie. The receiving node will additionally confirm the accuracy of the hash of the prior block reported on the present block. The node will broadcast this block to the peers in the case where validation was successful, and so forth.
7) The node where the transaction has been first submitted eventually receives the block. The smart contract will publish an event of VoteConfirmation after the voting transaction has been carried out throughout block validation, which will be received via the voter's web UI, which is registered to listen to this event.
8) A message of confirmation is shown on the screen to let the user know their vote submission was effective as soon as the web UI's JavaScript handles the VoteConfirmation event.

6.4 The Details of The Proposed System Implementation

6.4.1 Typically, the system consists of two models

6.4.1.1 The Administrator Module

The Administration module is made for the authorized person or admin of the organization. Admin can insert the names of candidates, and Deletion of the names of candidates.

6.4.1.2 The User or Voter Module

in this module, the User or voter will able to see the names of all electing Candidates and vote for the candidate. The implementation of smart contracts consists of setting up an Ethereum-based blockchain, using the Ganache tool, that will create a blockchain with ten accounts already configured, with 100 Ethers each. Then, running a smart contract on this blockchain requires uploading it to the Ethereum Virtual Machine through one account. Therefore, the steps to set up, and deploy a smart contract are the following:

- Configuration of a local blockchain with nodes (virtual machines) and accounts, using Ganache.
- Develop a smart contract using Solidity language.
- Compile the Smart contract code using truffle.
- Deploy the smart contract: make the smart contract available to users of an Ethereum network.

6.4.1.3 Migrate the Smart Contract

Migrations are JavaScript files that assist in the deployment of smart contracts on the Ethereum network. These files are in charge of staging our deployment activities and are created with the premise that your deployment requirements would vary over time.

6.4.1.4 Client-Side Election

To carry out the voting, users need to pay a small gas amount. This gas payment can be done in multiple ways. One of the ways that we will be using here is Metamask. To log into Metalmark, we must select one of the ten accounts offered by Ganache, then we create a form that allows accounts to vote in an HTML file. Each candidate's id, name, and vote total are listed in this app. It has a ballot box where we can select our preferred candidate. We use the web3 library, a JavaScript package that enables the communication between our client-side application and the blockchain.

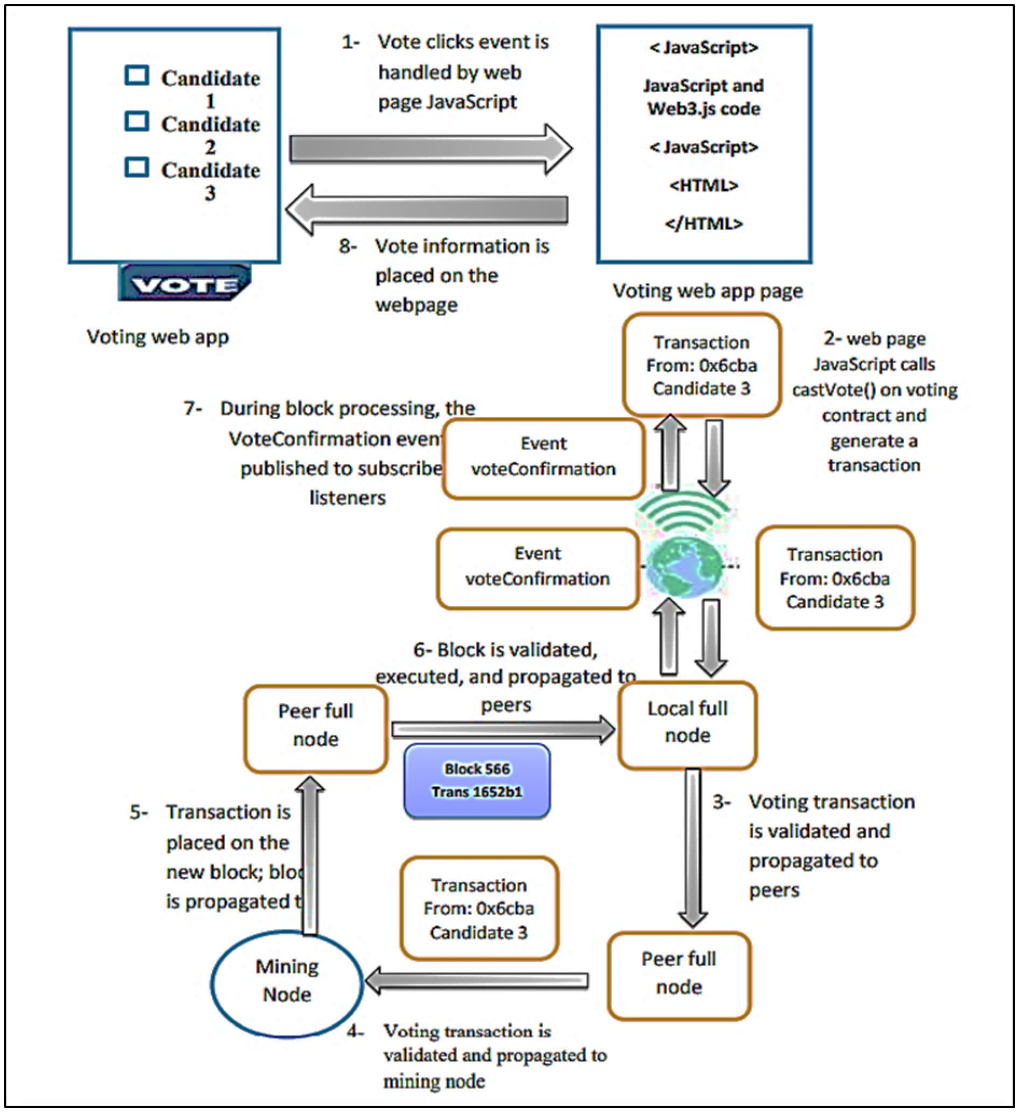

Figure 4: the Ethereum transaction steps

7. Measurements and Results

7.1 Measurement

Measurements used are

7.1.1 transaction cost

which is referred to as the gas cost as well, and it means fees that are required for the successful execution of a contract on the platform of the blockchain [33].

$$\text{Gas Fee} = \text{Gas Price} \times \text{Gas Limit}$$

7.1.2 Block time

which can be defined as a measure of time that is taken by validators or miners within the network for the verification of the transactions within a block and the production of a new block in this blockchain [34].

7.1.3 Blockchain difficulty

which specifies how difficult it is to mine the following block [35].

7.1.4 Transaction per Second (TPS)

Stransaction throughput that is referred to as the TPS (transactions per second). TPS represents the ratio of valid transactions that are initiated within a specified duration of time [36].

$$\text{transactions Per Block} = \text{block size} / \text{average transaction size}$$

7.2 Results

7.2.1 Test voting smart contract in the local blockchain (ganache)

For this test, we created 20 Ethereum wallets loaded with 100 ETH for each round of election and assigned 20 voting keys to the 20 voters. We used these keys throughout this evaluation. At each round of the election, we reset the blockchain and deployed the voting smart contract. Table 2, shows the deployment, and the details of running the 10 accounts. Figure 5 shows the transaction fees that are paid for each smart contract are constant, the detail shown in Table 3.

Table 2: The details of running 20 accounts in the Ganache blockchain

account name	account address	private key	estimate gas fee	max fee	real fee
acc1	0xfD9252e326C3f390Dc31bD3 e458A85313A8375CC	8544aa4f84d3277557787d39eca5a5068c4fc 8c3983b8373252bddfa4e60e4db	0.0019 96	0.00 1996	0.001 99634
acc2	0x9183b198343b60d80F56d2A F4656f9ac7697EA00	d10039d723b6983c741e7770b8aac3142649 74d6523ad48ec7da886db6f31ba2	0.0019 96	0.00 1996	0.001 99634
acc3	0x798cDC88F5A3eD1C2D6c1 D8CF750cD30cC002421	04d10780b51f6842a89193872e1dafee8ee77 2034c4fc446bb2a8dfcdad00806	0.0019 96	0.00 1996	0.001 99634
acc4	0x803DB26fABA3c0F7eb3c58 69D35623EdA40b33dA	04d10780b51f6842a89193872e1dafee8ee77 2034c4fc446bb2a8dfcdad00806	0.0019 96	0.00 1996	0.001 99634
acc5	0xA7d742b2A40dACE42b9C0 bd7E877e6c91cE3683e	919801c56c5e1b97321ea9d9d59ac4e3f3d3 dab17bd5f2a63b5ddc5efdf73b7b	0.0019 96	0.00 1996	0.001 99634
acc6	0x092c2873C4838E4350A023 D9Ee1C8E4A227F4bCC	405f853c145e196a6523facdec9c39bdb41b2 d0a25dd259465f23c0a909d617f	0.0019 96	0.00 1996	0.001 99634
acc7	0x2aA45144b22EA130F111670 1783845CAeEa1f694	558b1b692f646d669230a4995c921c9c9c06 6e16af5230d18763fd0d5a6af57b	0.0019 96	0.00 1996	0.001 99634
acc8	0x8175CD2f23ECAB518a1122 eF00D944b5d282e629	5ef6b4db69c36def134cc36e24d240966207f 5b0ff2337d171eb0a77cf76cb90	0.0019 96	0.00 1996	0.001 99634
acc9	0xAdAC26e92185FF42FFc64c 799A66B68312B25967	ab484b83f1130a15630341d6a0290666840a cb4b8282dad4f9a46f317c2fc2ad	0.0019 96	0.00 1996	0.001 99634
acc10	0x19cD3EDe78DC58634809B DB929BE25064AD13490	e830bac38aeb1530a3c4ec45038c7af96893c 26ef2a03fa3a187c2630800d7ba	0.0019 96	0.00 1996	0.001 99634

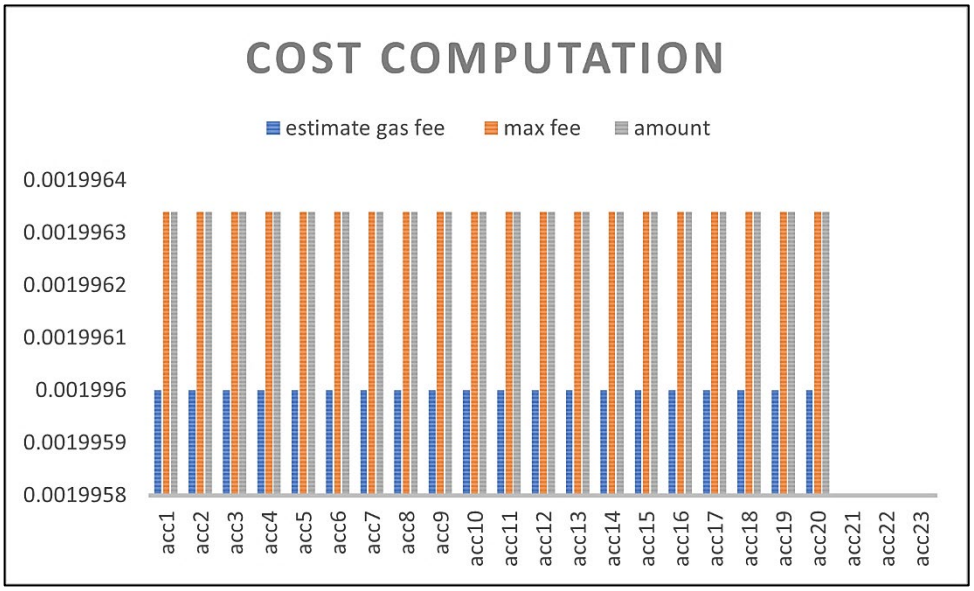

Figure 5: Cost Computation in the Ganache blockchain

Table 3: the details of transaction cost in the Ganache blockchain

Field name	Value
estimate gas fee	0.001996
max fee	0.00199634
real fee	0.00199634

7.2.2 Test the voting smart contract using Ethereum mainnet network

Due to the mainnet's large scale (about 8, 000 nodes) and the high price of Ether, we executed many transactions at different times, the execution in the mainnet needs a real ether, theTable 4 shows sample details of the execution. The result of implementing the smart contract on the mainnet is as the following, also explained in Table 5

7.2.2.1 The transaction cost

The price of Ether is not fixed for one day, it changes with the change of the network activity. When the number of transactions in the network increases, the price of Ether also increases as shown the Figure 6 and Figure 7, sometimes reaching more than $150 for one transaction and it goes back down to $10 or $11. In general, notes that the transaction price on Ethereum goes up a bit over time as shown the Figure 8.

7.2.2.2 Gas price

We note that the price of gas turn between 8**.**830704778 Gwei and 17**.**365398166 Gwei, the price of gas has a direct impact by the transaction fee. As shown in Figure 9.

7.2.2.3 Block time

there are 8000 nodes in the Ethereum mainnet, and the block time for it is between 12 to 30 seconds, it's around 16 sec on average.

7.2.2.4 Total difficulty

Total Ethereum difficulty is 58750000000000, As more hashing power has been added to Ethereum mining network, the difficulty has to increase to ensure blocks aren't being produced too fast. The difficulty has been increased in the case where preceding blocks have been produced faster than the block time that has been specified and reduced in the case where previous blocks have been generated at a slower pace compared to the specified block time. In the last two years, we notice that the amount of difficulty is constantly increasing, as shown in Figure 10, highest avg difficulty of 15,101.178 ether.

7.2.2.5 Throughput

Ethereum has very tiny blocks and ends up at about 14 transactions/second.

Table 4: A sample data of the executing transactions in the Mainnet network

Txhash	Blockno	DateTime	ContractAddress	TxnFee(ETH)	TxnFee(USD)
0xcd38223d46a5876e66dfd8a5e9b5ca9a06cee0bb0107e64694a4abc2ac289337	15541407	9/15/2022 20:16	0x6c180aac976109579316462dbfe7011ac4a500c3	0.00354193	4.615170379
0xf2a7ff638810e82db20fd19b4d25b7544abbffbb568b04326f6a2f71199f78c2	15568520	9/19/2022 15:42	0xd9af160ab00b99fdbeb7d66093299e2f306ad172	0.005034221	6.559640751
0xa78199e0be030b8acc61f237cc09331e547ae0abac2516764b56945abece56c2	15915967	11/7/2022 5:10	0xa2ee06594ad308ca49adb6020471a4f5944bd14c	0.005926684	7.722528043
0x36ce86f007de1e328283bc76ad3f80609ee430b9a1cf35eb40ef015e6b0ae022	15916015	11/7/2022 5:20	0x7b1a90a18bf4a0bb9de0710d3a643ee21b397636	0.005632585	7.339314476
0x6b05ba5bbf0d69b5be3fcd2524083e58552c69123fe5679e08991689cc0d978f	15930147	11/9/2022 4:39	0x2723aba84ceaecfbb2a7518066c273291276c30b	0.009120583	11.8842107
0xe19065dc7b4421bdd0c5163893582906bd9dc1e6436293fe606475d76c8cf5bd	15930168	11/9/2022 4:44	0x8248cad354506ef67860caa8fb6683754f24f12d	0.007587099	9.886065274
0xd65fc154b9dd069122d74f0081f96d29093e4c5c1d49ce9f773abf8cbde497e3	15930193	11/9/2022 4:49	0xe9c113021d6d3f728a0304784cc65fec1e068b97	0.006362861	8.290871604
0xd65fc154b9dd069122d74f0081f96d29093e4c5c1d49ce9f773abf8cbde497e4	15930193	11/10/2022 3:41	0xe9c113021d6d3f728a0304784cc65fec1e068b98	0.007436418	9.573895305
0xd65fc154b9dd069122d74f0081f96d29093e4c5c1d49ce9f773abf8cbde497e5	15930193	11/11/2022 5:29	0xe9c113021d6d3f728a0304784cc65fec1e068b99	0.010362463	12.68947364
0xd65fc154b9dd069122d74f0081f96d29093e4c5c1d49ce9f773abf8cbde497e6	15930193	11/12/2022 10:34	0xe9c113021d6d3f728a0304784cc65fec1e068b100	0.008386311	10.43256729

Table 5: The performance of the executed Transaction in the Mainnet network

No	Field name	Value of the field
1	Block time	16 sec
2	Average Transaction fee	0.0071 Ether (8.899$)
3	TPS	14 transactions/second
4	Total Difficulty	58750000000000
5	Gas Limit	30,000,000
6	Average Gas price	12.20173 Gwei
7	Burnt & Txn Savings Fees	0.00021 Ether

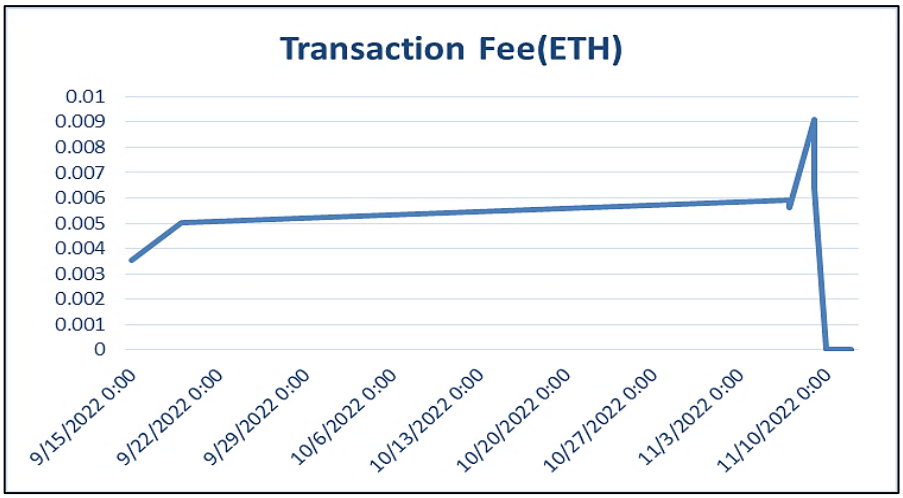

Figure 6: The transaction fee (Ether) in the Ethereum Mainnet network

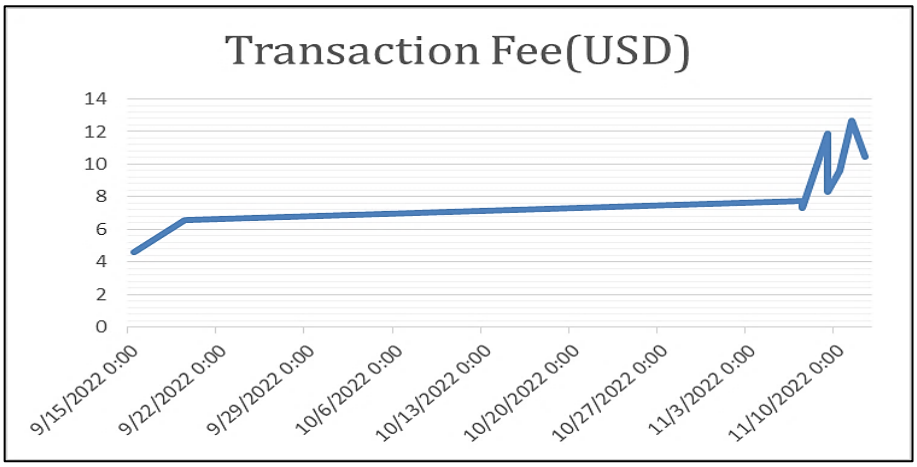

Figure 7: The transaction fee in (Ether)

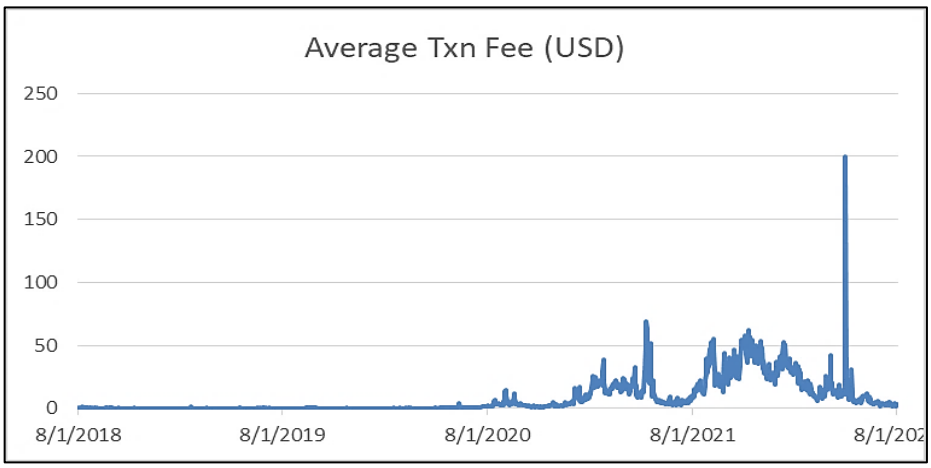

Figure 8: The changing in the transaction fee in the Ethereum Mainnet network during 4 years

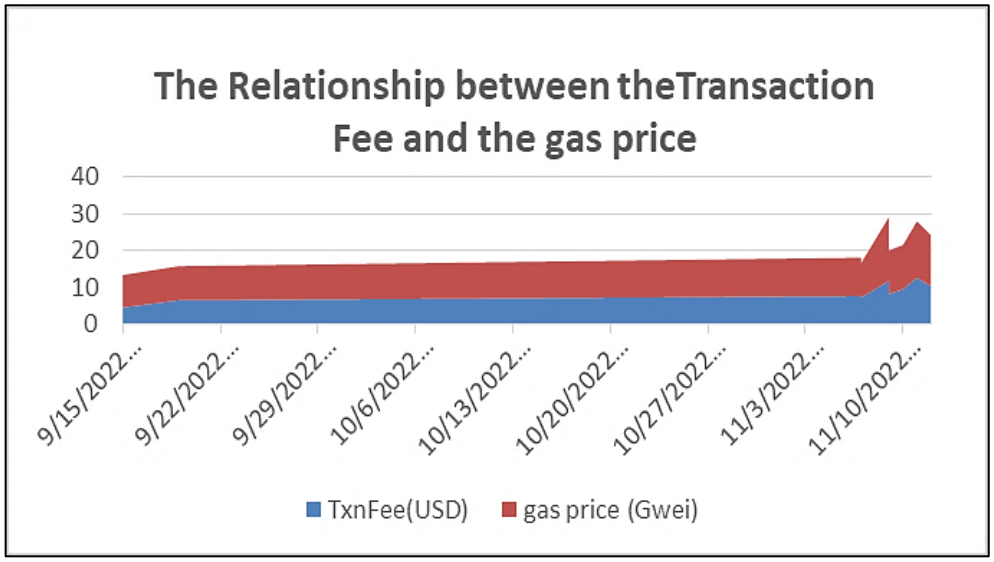

Figure 9: The Relationship between the transaction Fee and the gas price

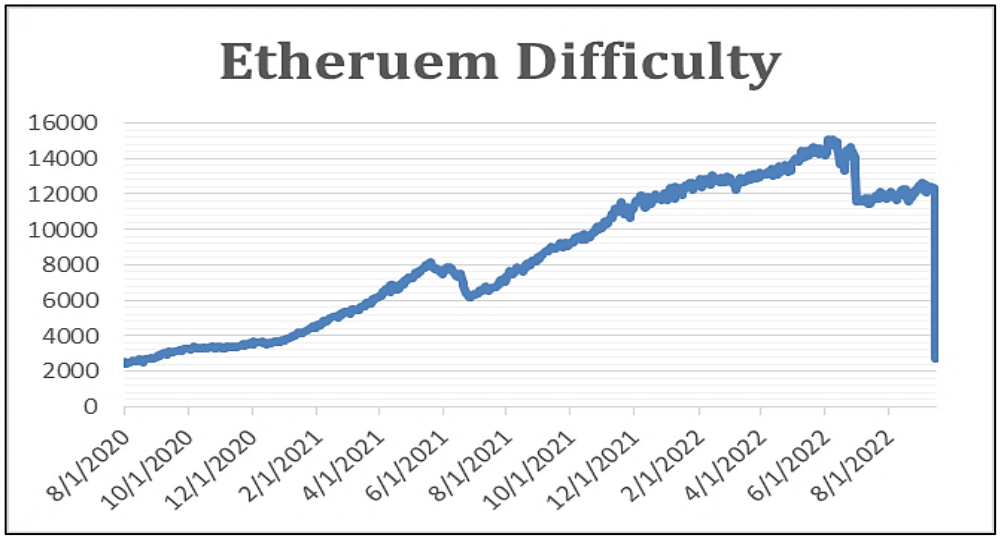

Figure 10: The changing of the transaction fee in the Ethereum Mainnet network over 2 years

8. Discussion

This research describes a smart contract on the blockchain that operates on Ethereum and provides a set of rules for communication and contract decision-making among participants. It suggests that blockchain technology has the potential to move beyond the constraints of centralized voting systems, demonstrating that it is not only quick and cheap but also safe and secure, making it more trustworthy and exact than previous techniques. Ganache, the Truffle framework, NPM, and metamask were among the technologies utilized in this project. A virtual client was used to test this implementation as well as deployed it in the main network using remix ide.

The results show that the proposed system has the following advantages:

- ability to assess the voter's rights and anonymity.
- the voter's ability to verify the authenticity of his vote. As well as the entire counting process of the votes is open to all to be monitored thus.
- reducing any chances of manipulations of the votes and the results are seen in real-time. can solve security and transparency.
- fairness and trust issues.

The Ethereum speed is faster than Bitcoin and Zcash blockchains, the TPS in bitcoin was 7 transactions per second, and in Zcash is 6 transactions per second, we make a comparison among different types of blockchains as shown in Table 6

Ethereum still suffers from Challenges due to its limited capacity for handling transactions. In the case where an excessive number of users attempt to push a transaction through simultaneously, it bogs down the system and results in slower times of transaction and greater fees for each one of the transactions. In the Proof of Work consensus, those block times must be quite

high for the minimization of the odds of several validators that simultaneously produce a new valid block in a simultaneous manner.

Table 6: Comparison Among general characteristics of bitcoin, zcash, Hyperledger blockchain

	Etheruem	**bitcoin**	**zcash**	**Hyperledger fabric**
Mode of operation	Permissionless, public	Permissionless, public	Public, private	Prementioned, private
Consensus	Proof of work	Proof of work	zero-knowledge proofs (ZKPs)	Can be Implemented in various ways
Currency	Ether	bitcoin	zec	none
scalability	High node scalability, high-performance scalability	High node scalability, low-performance scalability	High node scalability, low-performance scalability	low node scalability, high-performance scalability
Max block size	Unlimited	1MB	2MB	256KB
TPS	15 transactions	7 transactions	3 transactions	Unlimited

9. Conclusion

Blockchains are a new database type, they have solved some issues in centralized systems, like transactions with no need for any middlemen, time that is spent on every one of the transactions, and special or unintentional data modification or deletion in a Blockchain.

Through this work, we have come up with a set of conclusions that included:

1) On top of that, Ethereum has been inhibited by a 15 transactions/sec shared rate. This is because Ethereum keeps utilizing the Proof-of-Work and Ethereum d-Apps compete for limited single blockchain sources.
2) the cost of performing a transaction on Ethereum is typically between $5 and $160 per transaction, depending on network congestion.
3) the Ethereum block time averages around 10 to 20 seconds. In the Proof of Work consensus, those block times must be quite high for the minimization of odds of several validators that produce new valid block in a simultaneous manner.
4) Ethereum was more scalable than Bitcoin and Zcash blockchains.

10. Future Works

Several directions for future work are based on the results and the discussion of the research:

- we need to perform a security analysis of the Potential security attacks.
- One of the main disadvantages of the PoW is computing resource waste. To solve that issue, attempting to come up with a mechanism of hybrid consensus of PoS and PoW.
- Using the barcode of the voter identification card for authorization the voters because it's more secure.

Author contribution

All authors contributed equally to this work.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability statement

The data supporting this study's findings are available on request from the corresponding author.

Conflicts of interest

The authors declare that there is no conflict of interest.

References

[1] U. Jafar, M. J. A. Aziz, and Z. Shukur, Blockchain for the electronic voting system—review and open research challenges, Sensors, 21 (2021) 5874. https://doi.org/10.3390/s21175874

[2] W.-J. Lai, Y.-C. Hsieh, C.-W. Hsueh, and J.-L. Wu, Date: A decentralized, anonymous, and transparent e-voting system, 2018 1st IEEE Int. Conf. Hot Information-Centric Networking, 2018,18374604. https://doi.org/10.1109/HOTICN.2018.8605994

[3] S. Gupta and M. Sadoghi, Blockchain transaction processing, arXiv preprint arXiv:2107.11592, 2021. https://doi.org/10.48550/arXiv.2107.11592

[4] K. Wüst and A. Gervais, Do you need a blockchain?, 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), 2018, 18232822. https://doi.org/10.1109/CVCBT.2018.00011

[5] S. Bistarelli, M. Mantilacci, P. Santancini, and F. Santini, An end-to-end voting-system based on bitcoin, in Proceedings of the Symp. Appl. comp., 2017, 1836–1841. https://doi.org/10.1145/3019612.3019841

[6] P. Tarasov and H. Tewari, Internet voting using zcash, Cryptology ePrint Archive, 2017.

[7] B. Shahzad and J. Crowcroft, Trustworthy electronic voting using adjusted blockchain technology, IEEE Access, 7(2019) 24477–24488. https://doi.org/10.1109/ACCESS.2019.2895670

[8] M. Mitra and A. Chowdhury, A Modernized Voting System Using Fuzzy Logic and Blockchain Technology, Int. J. Mod. Educ. Comput. Sci., 12 (2020) 17-25. https://doi.org/10.5815/ijmecs.2020.03.03

[9] A. Bahga and V. K. Madisetti, Blockchain platform for industrial internet of things, J. Softw. Eng. Appl., 9 (2016) 533–546. https://doi.org/10.4236/jsea.2016.910036

[10] J. Golosova and A. Romanovs, The advantages and disadvantages of the blockchain technology, 2018 IEEE 6th Workshop on Advances in Information, Electro. Electr. Eng., 2018, 18356489. https://doi.org/10.1109/AIEEE.2018.8592253

[11] D. Vujičić, D. Jagodić, and S. Ranđić, Blockchain technology, bitcoin, and Ethereum: A brief overview 2018 17th Int. Symp. Infoteh-Jahorina , 2018, 17732472. https://doi.org/10.1109/INFOTEH.2018.8345547

[12] V. Buterin, A next-generation smart contract and decentralized application platform, white paper, 3 (2015) 1–2.

[13] S. Rouhani and R. Deters, Performance analysis of ethereum transactions in private blockchain, in 2017 8th IEEE int. conf. Softw. Eng. Serv. Sci ., 2017 , 17746076. https://doi.org/10.1109/ICSESS.2017.8342866

[14] Xiwei Xu , Ingo Weber and Mark Staples, Architecture for blockchain applications. Springer, 2019. https://doi.org/10.1007/978-3-030-03035-3

[15] B. Salam Al-E'mari, Y. Sanjalawe, and S. Manickam, A Labeled Transactions-Based Dataset on the Ethereum Network, in Advances in Cyber Security: Second International Conference, ACeS 2020, Penang, Malaysia, December 8-9, 2020, Revised Selected Papers. 1347, 2021, 61–79.https://doi.org/10.1007/978-981-33-6835-4_5

[16] B. K. Mohanta, S. S. Panda, D. Jena, An overview of smart contract and use cases in blockchain technology, 2018 9th Int. Conf.Comput. Commun. Netw. Technol., 2018, 1–4. https://doi.org/10.1109/ICCCNT.2018.8494045

[17] S. Raval, Decentralized applications: harnessing Bitcoin's blockchain technology. O'Reilly Media, Inc., 2016.

[18] Mohanty, D., 2018. Ethereum Architecture, in Ethereum for Architects and Developers, Springer, pp. 37–54. https://doi.org/10.1007/978-1-4842-4075-5_2

[19] K. Wang, Q. Wang, and D. Boneh, "ERC-20R and ERC-721R: Reversible Transactions on Ethereum," arXiv preprint arXiv:2208.00543, 2022. https://doi.org/10.48550/arXiv.2208.00543

[20] W. Ethereum, Ethereum Whitepaper, Ethereum., 2014.

[21] B. Chinthanet et al., What makes a good Node. js package? Investigating Users, Contributors, and Runnability, arXiv preprint arXiv:2106.12239, 2021. https://doi.org/10.48550/arXiv.2106.12239

[22] J. Ali, Instant Node Package Manager. Packt Publishing Ltd, 2013.

[23] J. Oberschweiber, A package manager for Curry, Masterarbeit. Christian-Albrechts-Universität zu Kiel, 2016.

[24] F. Doglio, Pro REST API Development with Node. js. Apress, 2015.

[25] K. Lei, Y. Ma, Z. Tan, Performance comparison and evaluation of web development technologies in php, python, and node. js, 2014 IEEE 17th international conference on computational science and engineering, 2014, 14887184. https://doi.org/10.1109/CSE.2014.142

[26] H. Sun, D. Bonetta, C. Humer, and W. Binder, Efficient dynamic analysis for Node. js, in Proceedings of the 27th International Conference on Compiler Construction, 2018, 196–206. https://doi.org/10.1145/3178372.3179527

[27] K. Bhosale, K. Akbarabbas, J. Deepak, A. Sankhe, Blockchain based secure data storage, Int. J. Eng. Res., 6 (2019) 5058–5061.

[28] S. Khan, A. Arshad, G. Mushtaq, A. Khalique, and T. Husein, Implementation of Decentralized Blockchain E-voting, EAI Endorsed Transactions on Smart Cities, 4 (2020)1–12. http://dx.doi.org/10.4108/eai.13-7-2018.164859

[29] S. A. Renu and B. G. Banik, Implementation of a secure ridesharing DApp using smart contracts on Ethereum blockchain, Int. J. Saf. Secur. Eng., 11 (2021) 167–173. https://doi.org/10.18280/ijsse.110205

[30] P. Hegedűs, Towards analyzing the complexity landscape of solidity based ethereum smart contracts, in Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain, 2018, 35–39. https://doi.org/10.1145/3194113.3194119

[31] Chris Dannen, Introducing Ethereum and solidity, Springer, 2017. https://doi.org/10.1007/978-1-4842-2535-6

[32] M. J. A. Baig, M. T. Iqbal, M. Jamil, and J. Khan, Design and implementation of an open-Source IoT and blockchain-based peer-to-peer energy trading platform using ESP32-S2, Node-Red and, MQTT protocol, Energy Rep., 7 (2021) 5733–5746. https://doi.org/10.1016/j.egyr.2021.08.190

[33] A. Jabbar and S. Dani, Investigating the link between transaction and computational costs in a blockchain environment, Int .J. Prod. Res., 58 (2020) 3423–3436. https://doi.org/10.1080/00207543.2020.1754487

[34] L. M. Bach, B. Mihaljevic, and M. Zagar, Comparative analysis of blockchain consensus algorithms, in 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2018, 1545–1550. https://doi.org/10.23919/MIPRO.2018.8400278

[35] D. Fullmer and A. S. Morse, Analysis of difficulty control in bitcoin and proof-of-work blockchains, in 2018 IEEE Conference on Decision and Control (CDC), 2018, 5988–5992. https://doi.org/10.1109/CDC.2018.8619082

[36] L. Hang, I. Ullah, and D.-H. Kim, A secure fish farm platform based on blockchain for agriculture data integrity, Comput. Electron. Agric.,170 (2020) 105251. https://doi.org/10.1016/j.compag.2020.105251