International Journal of Frontier Technology and Engineering (IJFTE)
Vol. 03, No. 01, December 2024, pp. 1-18
ISSN: 2986-9064

Blockchain-Based E-Voting System: A Decentralized Approach on the Ethereum Private Network

Yan Watequlis Syaifudin[1], Silvia Prada Aprilia[1], Salies Apriliyanto[2], Dinda Rizqiyatul Himmah[3], Indrazno Siradjuddin[4], Mohammad Sinal[5], and Alfiandi Aulia Rahmadani[4]

[1]*Department of Information Technology, Politeknik Negeri Malang, Malang, Indonesia*
[2]*PT. Infonika Parasa, Surabaya, Indonesia*
[3]*Programme of Private International Law, University of Groningen, Groningen, Netherlands*
[4]*Department of Electrical Engineering, Politeknik Negeri Malang, Malang, Indonesia*
[5]*Unit of General Course, Politeknik Negeri Malang, Malang, Indonesia*

Corresponding Author: Yan Watequlis Syaifudin, qulis@polinema.ac.id

Received Date: 02-08-2024 Revised Date: 05-10-2024 Accepted Date: 21-11-2024

Abstract

Blockchain technology significantly enhances the security, transparency, and integrity of electronic voting systems by decentralizing vote recording, enabling public verification, and ensuring voter anonymity through cryptographic techniques. This decentralized ledger guarantees that once a vote is cast, it is securely documented and cannot be modified without network consensus, safeguarding the election's credibility. Smart contracts automate processes such as voter identity verification and vote tallying, reducing the chances of fraud and errors. Built on the private Go Ethereum (Geth) network, this study implements an e-voting system that features validator nodes for transaction verification, user-friendly mobile applications for voter interaction, and comprehensive smart contract capabilities that define the voting process rules. The mobile application guides users from the launch of the application and authenticating their identity to the selection of candidates and the secure recording of their votes, ensuring seamless connectivity to the blockchain and incorporating essential verification steps. Furthermore, the implementation process includes meticulous setup of Ethereum nodes, smart contract development, and thorough testing for functionality, performance, and security. The implementation of this blockchain-based e-voting system involves setting up an Ethereum private network with nodes and smart contracts to manage the secure voting process, focusing on security, transparency, and immutability. The performance of the system was evaluated through comprehensive tests that covered system functionality, validater node integrity, access control, and data transparency. Testing confirmed the application's reliability, efficiently handling transaction loads, and blocking unauthorized access. While initial blockchain synchronization caused slight delays, the system operated smoothly under higher traffic, ensuring data integrity and security. Although positive user feedback highlighted confidence in the system's transparency and reliability, challenges persist, particularly in public familiarity with blockchain technologies, necessitating simplified user interfaces and robust educational campaigns for broader adoption. Technical challenges also remain, such as scalability issues typical in blockchain networks such as Ethereum, and require accessible design and infrastructure, especially in less developed areas, to ensure widespread and effective use.

Keywords : election, e-voting, blockchain, mobile application, security, smart contract

1 Introduction

Blockchain technology greatly improves the security, transparency, and integrity of electronic voting systems by distributing the vote recording, allowing them to be publicly confirmed, and preserving voter

anonymity through cryptographic techniques[1]. This decentralized ledger technology guarantees that once a vote is submitted, it is securely documented and cannot be modified without the agreement of network participants, thus safeguarding the credibility of the election procedure[2]. Smart contracts streamline essential procedures, such as verifying voter identity and tallying votes, hence reducing the likelihood of fraudulent activities and mistakes made by individuals. A decentralized e-voting system comprises validator nodes responsible for verifying transactions, smart contracts that automate the election process, and user interfaces that let voters securely cast their ballots[3]. This system utilizes a private Go Ethereum network to strike a balance between maintaining control over the voting environment and capitalizing on the inherent security and transparency aspects of blockchain technology. This ultimately promotes trust among stakeholders and guarantees a fair and dependable electoral result[4].

The mobile application's voting process is meticulously designed to be user-friendly and secure, guiding voters through each step from launching the app and authenticating their identity to selecting candidates and securely recording votes on the blockchain[5]. Upon opening the application, users connect to the *Go Ethereum (Geth)* private network[6] and are prompted to log in with a unique identifier and password verified against a smart contract. Once authenticated, voters can view a list of candidates with relevant information, select their preferred candidates, and go through verification steps to ensure that they have not voted previously. The application then encrypts the vote data and signs it using the user's private key before transmitting the transaction to the network. Validator nodes authenticate this transaction[7], which is recorded on the blockchain, providing confirmation of the vote and ensuring the integrity of the election process while maintaining voter anonymity through advanced cryptographic techniques.

This study presents the implementation of a blockchain-based e-voting system. It starts with establishing an Ethereum private network that involves the setup of nodes running Ethereum clients[8], the development and deployment of smart contracts using Solidity to manage a secure and transparent voting process[9], and facilitating user interactions through transactions. The process begins with configuring the nodes that maintain copies of the blockchain and verify transactions. Smart contracts play a pivotal role in defining the rules and logic of the voting system[10], ensuring aspects such as security, transparency, and immutability. The key components of these contracts include structures for voters and candidates, as well as functions for adding, updating, or deleting voter and candidate information, casting votes, and retrieving historical vote data.

Furthermore, the mobile-based platform implementation process divides into several critical sub-sections[11], including technical aspects such as smart contract development, blockchain integration, and user interface design. The application is built using *TypeScript* to ensure compatibility with various mobile operating systems and features an intuitive user interface designed for ease of use. To develop the Go Ethereum network, key steps include installing *Geth*, creating node accounts, configuring the Genesis file, and initializing nodes to start the network. The mobile application facilitates secure and transparent voting processes by connecting to the Geth node, handling transactions, monitoring blockchain events, and retrieving necessary blockchain data.

The evaluation encompasses four primary areas: testing scenarios, system functionality, performance, and security and transparency, aimed at assessing the system's reliability, efficiency, and robustness. The testing scenarios verify the integrity of validator nodes, ensure effective blocking of unauthorized access, and maintain data transparency for authorized users even when validator nodes are down. System functionality tests, such as launching the application, credential verification, and accurate vote display, confirm the core features work seamlessly across different devices. Performance tests demonstrate that the system effectively handles user authentication, voting processes, and data storage, maintaining smooth operation despite minor delays during initialization. Finally, rigorous checks for security and transparency verify the immutability of voting data, user privacy, and robust access controls, highlighting the system's capability to balance security and transparency effectively. User feedback indicates high satisfaction with the app's usability and speed, reflecting increased confidence in the e-voting process, and the overall evaluation asserts that the blockchain-based system offers a comprehensive solution to contemporary electoral challenges.

The rest of the paper is organized as follows: Section 2 discusses several studies related to the development of blockchain-based e-voting systems and Ethereum platform. Section 3 describes how blockchain technology improves the security, transparency, and integrity of electronic voting systems by decentralizing vote recording. Section 3 describes how blockchain technology improves the security, transparency, and integrity of electronic voting systems by decentralizing vote recording. Section 4 explains the voting process using the mobile application designed to be user-friendly and secure. Section 5 presents the implementation of an Ethereum private network that involves setting up nodes that run Ethereum clients. Section 6 details the implementation process of the mobile-based platform in the research conducted. Section 7 presents the evaluation of this

blockchain-based e-voting system that covers four main areas. Finally, Section 9 concludes this study with presentation of future works.

2 Related Works

This section discusses several important studies that have made significant contributions to the development of blockchain-based e-voting systems and Ethereum platform.

2.1 Studies on E-Voting Systems

Research by Susanto[12] highlights the use of the Solidity programming language to interact with the Ethereum blockchain in an e-voting system. This study demonstrates that smart contracts can generate unique codes for every new election, thereby preventing the manipulation of election results. The results of this study show the great potential of blockchain technology to improve the security and integrity of election systems.

Song et al.[13] developed a prototype voting system using Ethereum blockchain technology. The trial of this system showed that users found it easier to operate and the system performed well in terms of the completeness of the information presented. However, the study also identified several drawbacks, including a transaction execution time of approximately 30 seconds, which could affect the overall efficiency of the system.

2.2 Ethereum Platform

Ethereum is a decentralized blockchain platform that allows developers to build and deploy smart contracts and decentralized applications (dApps)[14]. Proposed by Vitalik Buterin in late 2013, Ethereum went live on July 30, 2015[8], following a development period that began with a public crowd sale in 2014. Ethereum features the Ethereum Virtual Machine (EVM), a Turing-complete environment that facilitates the development and execution of dApps. The platform operates using its native cryptocurrency, Ether (ETH), to pay for transaction fees and computational services. Ethereum transitioned from a proof-of-work (PoW) to a proof-of-stake (PoS) consensus mechanism with Ethereum 2.0, improving scalability, energy efficiency, and security. It also plays a crucial role in the space of decentralized finance (DeFi), allowing a variety of financial services without intermediaries, as well as popularizing non-fungible tokens (NFTs)[15].

Go Ethereum[6], commonly known as Geth, is one of the most popular implementations of an Ethereum node, written in the Go programming language. It allows users to run their own instance of the Ethereum blockchain, enabling participation in the network by validating transactions and blocks[16]. Geth serves various purposes such as deploying smart contracts, managing accounts, and mining Ether. In addition, it provides a command-line interface for interacting with the Ethereum network, making it a versatile tool for developers and enthusiasts looking to engage with Ethereum. Geth supports various operational modes, such as the full node operation, which stores the entire blockchain, and the light node operation, which only downloads block headers and requires less storage.

Proof of Authority (PoA)[17] is a consensus mechanism used in blockchain networks that operates through a small number of designated nodes, called validators, to validate transactions and create new blocks, as shown in Figure 1. This approach differs from systems like proof of work (PoW) or proof of stake (PoS) which require significant computational power or staked cryptocurrency[18]. Instead, PoA depends on the verified identities and reputations of its validators, making these personal attributes crucial for achieving consensus.

2.3 Blockchain Technology in Voting

Several studies[19, 20] explore a web-based e-voting application that uses multichain to manage one or more blockchains. The blockchain technology in this study helps store transparent voting data, maintain voter confidentiality, and ensure that vote data cannot be changed, duplicated, or deleted. The results of this study show that blockchain can provide a reliable solution to the problems often faced in traditional e-voting systems.

Hjalmarsson et al.[21] implemented an e-voting system with blockchain technology using Go-Ethereum Proof-of-Authority (PoA), as well as this study[22]. This study demonstrates the speed of transactions, albeit with performance limitations and higher costs compared to centralized systems. However, this study reinforces the notion that blockchain technology can enhance transparency and security in the electoral process.

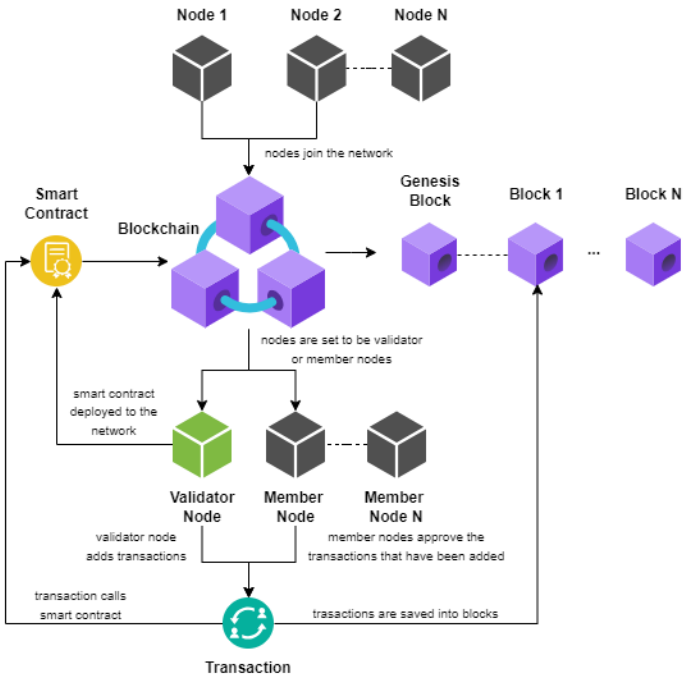

Figure 1: PoA Consensus Workflow on Private Blockchain Go Ethereum

This research provides a solid foundation for further development of blockchain-based e-voting systems. By understanding the advantages and disadvantages of existing approaches, this research aims to develop better and more efficient solutions for decentralized election processes.

3 Decentralized E-voting

This section describes how blockchain technology improves the security, transparency and integrity of electronic voting systems by decentralizing vote recording, ensuring public verification, maintaining voter anonymity through cryptographic methods, and automating processes to minimize fraud and human error.

3.1 Utilization of Blockchain

A potential remedy to improve the security, transparency, and integrity of electronic voting systems is blockchain technology[23]. Blockchain is essentially a distributed[24], decentralized ledger that keeps track of transactions across several computers. Because previous entries cannot be changed or removed without the network's members agreeing, it is almost impossible to do so. This implies that every vote cast during electronic voting can be recorded in a safe and open way for public verification[25]. Election integrity is maintained, while anonymity is maintained through the use of cryptographic techniques that guarantee that voter identities are secure. Voter authentication and counting are two examples of voting process automation that smart contracts in blockchain applications can automate, lowering the possibility of fraud or human mistakes.

Furthermore, the integration of blockchain technology into electronic voting effectively addresses various challenges in trust and accountability[21]. A blockchain-based voting system differs from traditional systems in that it distributes control among multiple nodes in a network[26], making it resistant to tampering and cyberattacks, unlike centralized systems, where a single authority can manipulate results. The decentralized aspect of this system fosters transparency, as all parties involved can independently check outcomes, while the unchangeable record of the blockchain enables straightforward audits of the election process. Moreover, the ability of blockchain to deliver instantaneous conclusions boosts the overall effectiveness of elections, reducing the time required to finalize the results. The use of blockchain technology into e-voting systems is a major

step forward in the endeavor to provide secure, transparent, and reliable electoral procedures[27].

3.2 Decentralized E-voting System Architecture

The decentralized e-voting system architecture uses a blockchain network to record and verify votes[3]. The system consists of several key components, including validator nodes, smart contracts, and a user interface. Validator nodes are responsible for verifying transactions and maintaining the integrity of the blockchain. Smart contracts are used to automate the election process and ensure that election rules are followed. The user interface allows voters to cast their votes securely and easily[28].

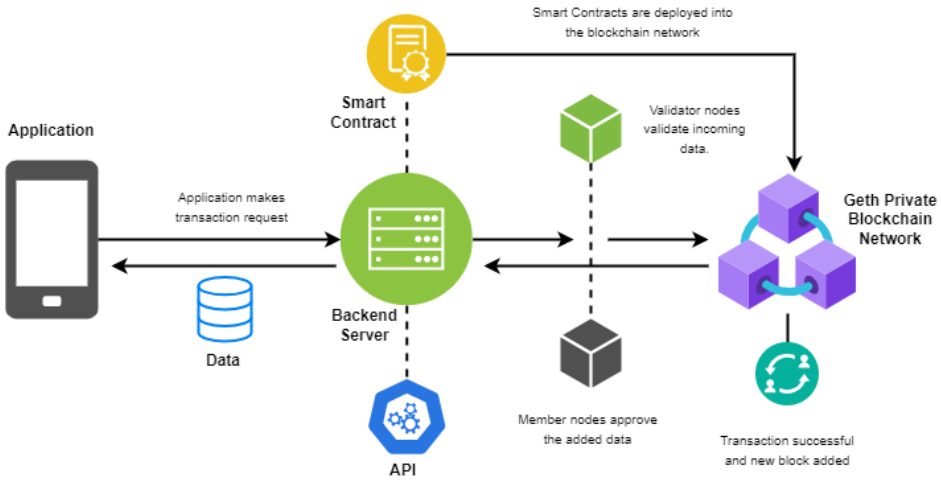

Figure 2: System Architecture for Evoting Using Private Blockchain Go Ethereum

The application described in this paper is specifically built on a private Go Ethereum network[29], leveraging the robust and flexible infrastructure provided by the Ethereum ecosystem. Using a private network, we can maintain greater control over the voting environment while still benefiting from the security and transparency inherent in blockchain technology. The following subsections will explore various aspects of this implementation, including the system architecture, the role of smart contracts[30], and the intricacies of the voting process within this private Ethereum-based framework[31].

3.3 Election Process in Decentralized E-voting

A decentralized e-voting system encompasses various stakeholders and components that collaborate to provide a secure, transparent, and dependable election process. The election process commences with the election administrator, who initiates the election by establishing its parameters, including the roster of candidates and eligible voters, as well as choosing the duration of the voting period. By employing blockchain technology or a comparable distributed ledger system, the administrator ensures that the election infrastructure is decentralized, thereby mitigating the possibility of manipulation by any individual institution (Ebenez 2014). Consensus processes and cryptographic security uphold the integrity of the system, enabling all parties involved, including voters and candidates, to function within a reliable setting.

To participate in this decentralized system, voters and candidates engage directly with the blockchain-based platform. Voters verify their identity using safe mechanisms[32], such as cryptographic keys, to access the voting system while maintaining anonymity. Once the voters cast their votes, the blockchain permanently records these votes, rendering them unchangeable. Validator nodes have a vital function in this context. They are responsible for confirming transactions, specifically vote casting, and ensuring the integrity of the blockchain. Validator nodes collaborate to validate and establish a unanimous agreement on the condition of the ledger, guaranteeing the accurate recording of votes and the integrity of the election process. Registered and approved candidates may transparently monitor the election process, thanks to the blockchain's public nature, which provides a verifiable record of all transactions. The decentralized architecture enhances both security and transparency, while also promoting trust among users, thereby guaranteeing a fair election result.

3.4 Security and Privacy in E-voting

The utmost importance of security in e-voting systems is to ensure the fairness and accuracy of elections[33]. The main security goals encompass authentication, integrity, and availability. Authentication systems are essential for verifying the identity of voters, so guaranteeing that only eligible individuals can cast a vote. Methods such as multi-factor authentication can improve the effectiveness of this procedure[34]. Integrity entails protecting the voting data from any unauthorized changes or interference, maybe via sophisticated cryptographic techniques or blockchain technology to ensure the accuracy of the records. Ensuring availability is critical because it protects the system from disruptions or denial-of-service attacks that could impede voting. An effective e-voting system must incorporate safeguards to protect against diverse cyber threats, ensuring the security of the entire voting process.

In terms of privacy, electronic voting systems must guarantee the confidentiality of the voter[35]. This entails protecting voters' confidentiality to prevent any possibility of tracing the votes back to the individuals who cast them. Privacy-preserving methods, like as encryption, are crucial for ensuring the secure and secret transfer and storage of votes. In addition, it is crucial to strike a balance between transparency and auditability while taking privacy concerns into account to maintain public trust in election results. Systems should be designed to provide independent audits and verifications while ensuring the protection of individual voter privacy. The careful equilibrium between strong security measures and stringent privacy protocols is crucial to cultivating voter confidence and guaranteeing the integrity and confidentiality of the political process.

4 Election Procedure

This section explains the voting process using the mobile application that is designed to be user-friendly and secure, guiding voters through steps from app launch and authentication to selecting candidates and securely recording votes on the blockchain.

4.1 General Procedure

The voting process is designed to be user-friendly while maintaining the highest standards of security and integrity[36]. This chapter outlines the step-by-step procedure that a voter follows when participating in an election using our mobile application, as shown in Figure 3. From opening the app to final recording of votes on the blockchain, each stage is carefully crafted to ensure a smooth user experience and maintain the sanctity of the voting process, as shown in Figure 4. The following sections detail each phase of the voting journey:

4.2 Open the App

The e-voting process begins when a user launches the mobile app on their device. The app, built using React Native and TypeScript for cross-platform compatibility, immediately establishes a connection to the Go Ethereum (Geth) private network. Once launched, the app syncs with the blockchain to ensure it is working with up-to-date information. The user is then presented with a Login Screen to access the voting interface.

4.3 Get Credentials

To access the voting interface, users must first authenticate themselves. The app displays a login screen where users enter their unique identifier and password. These credentials are then verified against the information stored in a smart contract deployed on the Ethereum network. This smart contract, implemented using Solidity and deployed via Hardhat, processes the login request and determines whether to grant access. If authentication is successful, the system displays the home page, which displays the candidates running in the election, the number of votes each candidate has received, and historical data from the ongoing election.

4.4 Vote Candidates

Once authenticated, the application fetches the current list of candidates from the smart contract and presents it to the user on the Voting Screen. This screen displays relevant information for each candidate, such as the candidate's name and Vision & Mission. The interface offers a clear mechanism for voters to choose their preferred candidate, featuring a button that displays the candidate's name and ID for the vote feature. After making a selection, the user is presented with a confirmation dialog to ensure their choice is deliberate.

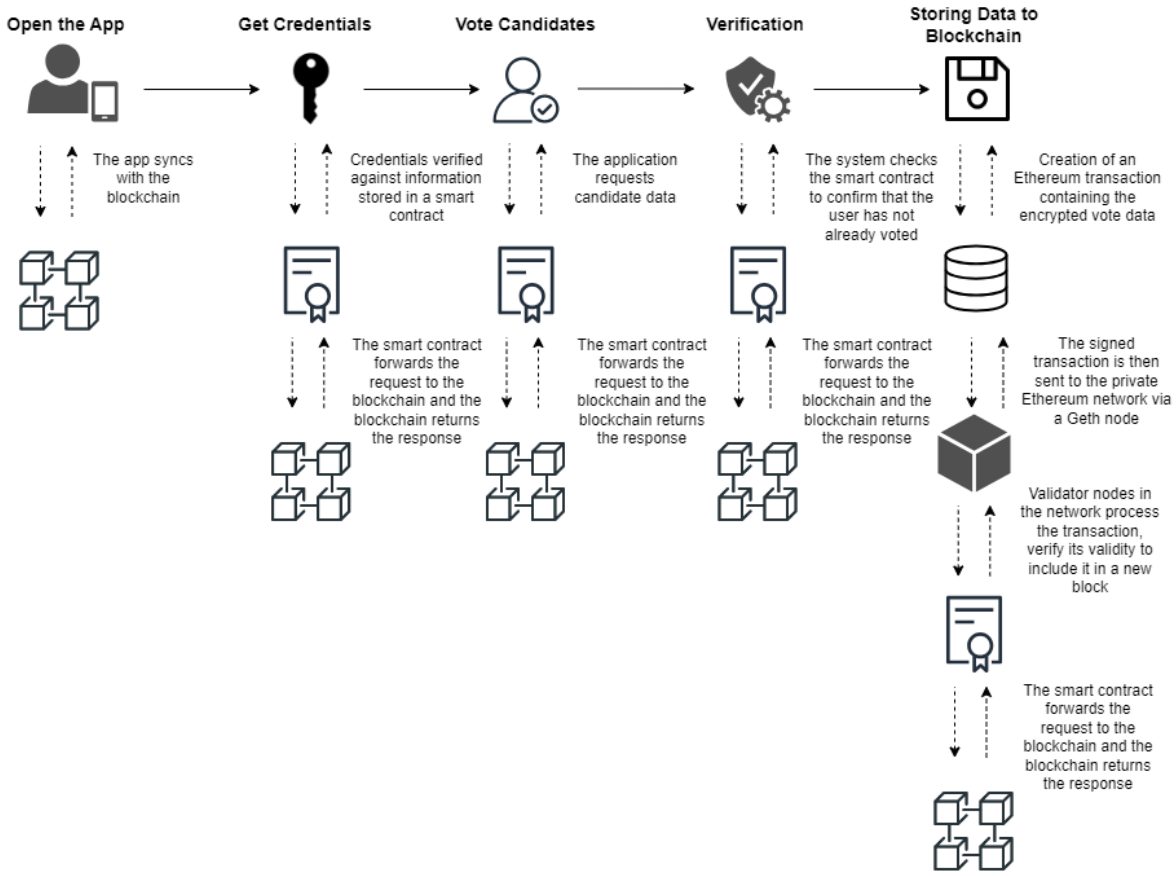

Figure 3: E-voting Procedure using Private Blockchain Go Ethereum Network

4.5 Verification

Before completing the vote, the system performs several verification steps. First, the system checks the smart contract to confirm that the user has not already voted in this election, maintaining the principle of "one person, one vote"[37]. As an additional security measure, the user is required to re-enter their password. After receiving final confirmation from the user, the application prepares the voting transaction, which includes the encrypted vote data and the user's blockchain address.

4.6 Storing Data to Blockchain

The process of storing a vote on the blockchain starts by generating an Ethereum transaction that includes encrypted vote data. This data is then signed using the user's securely maintained private key. A *Geth* node sends the signed transaction to the private Ethereum network, where validator nodes authenticate and confirm its legitimacy before incorporating it into a brand-new block. When the smart contract receives the transaction, it starts the voting function's execution, which modifies the internal state of the contract to include the new vote. This is done by increasing the total vote count and indicating that the user has voted. The intelligent agreement emits an occurrence to indicate the successful registration of the vote, enabling instantaneous updates and tracking of past events.

Once the transaction is completed, the application remains in a state of anticipation until it receives confirmation that the transaction has been successfully added to the block and safely stored on the blockchain. Upon receipt of this confirmation, the user is promptly notified, so ensuring the formal casting of their vote while maintaining their privacy. To make sure that voting is safe, clear, and easy to check, the whole process uses the Ethereum blockchain's security features, such as the Proof of Authority (PoA) consensus mechanism[38]. Smart contracts provide the automated enforcement of voting rules and provide an immutable

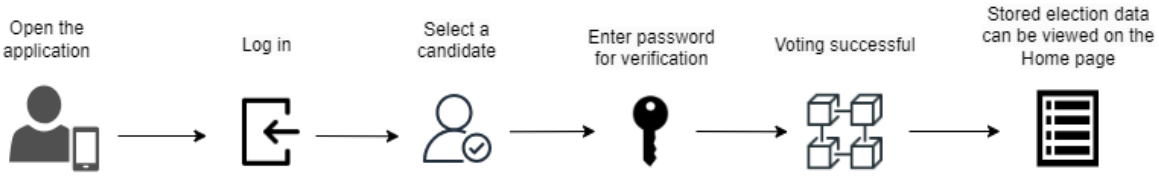

Figure 4: E-voting Procedure Workflow

record of all votes, ensuring the creation of a secure and reliable e-voting system.

5 Implementation of Ethereum Private Network

This section presents the implementation of an Ethereum private network that involves setting up nodes running Ethereum clients, developing and deploying smart contracts using Solidity to manage the secure and transparent voting process, and facilitating user interactions with these contracts through transactions.

5.1 Network Setup

The implementation begins with the setup of the Ethereum network. This involves configuring nodes that will participate in the network. Each node runs an Ethereum client which maintains a copy of the entire blockchain and verifies transactions. Popular clients include Geth and Parity[39, 40].

5.2 Smart Contract Development

Smart contracts are developed using Solidity[41], Ethereum's contract-oriented programming language. These contracts define the rules and logic of the application. In the context of this research, smart contracts handle the voting process, ensuring security, transparency, and immutability. The smart contract design for the blockchain-based e-voting system is crucial in ensuring the security, transparency, and integrity of the voting process. The contract's structure and functionality are outlined in the accompanying diagram.

5.3 Core Components

The core components of the smart contract include:

- **Voter**: This struct represents the voter and stores their unique identifier, name, email, and voting status.
- **Candidate**: This struct represents the candidate and stores their unique identifier, name, vision, and mission.
- **VoterHistory**: This struct stores the voter's voting history, including the timestamp and the candidate they voted for.
- **VoteCountHistory**: This struct tracks the vote count for each candidate over time.

5.4 Key Functions

The key functions within the smart contract are:

- `addVoter(uint256, string, string, string)`: Adds a new voter to the system.
- `updateVoter(uint256, string, string, string)`: Updates an existing voter's information.
- `deleteVoter(uint256)`: Removes a voter from the system.
- `addCandidate(uint256, string, string, string)`: Adds a new candidate to the election.
- `updateCandidate(uint256, string, string, string)`: Updates an existing candidate's information.
- `deleteCandidate(uint256)`: Removes a candidate from the election.
- `vote(uint256, uint256)`: Allows a voter to cast their vote for a specific candidate.

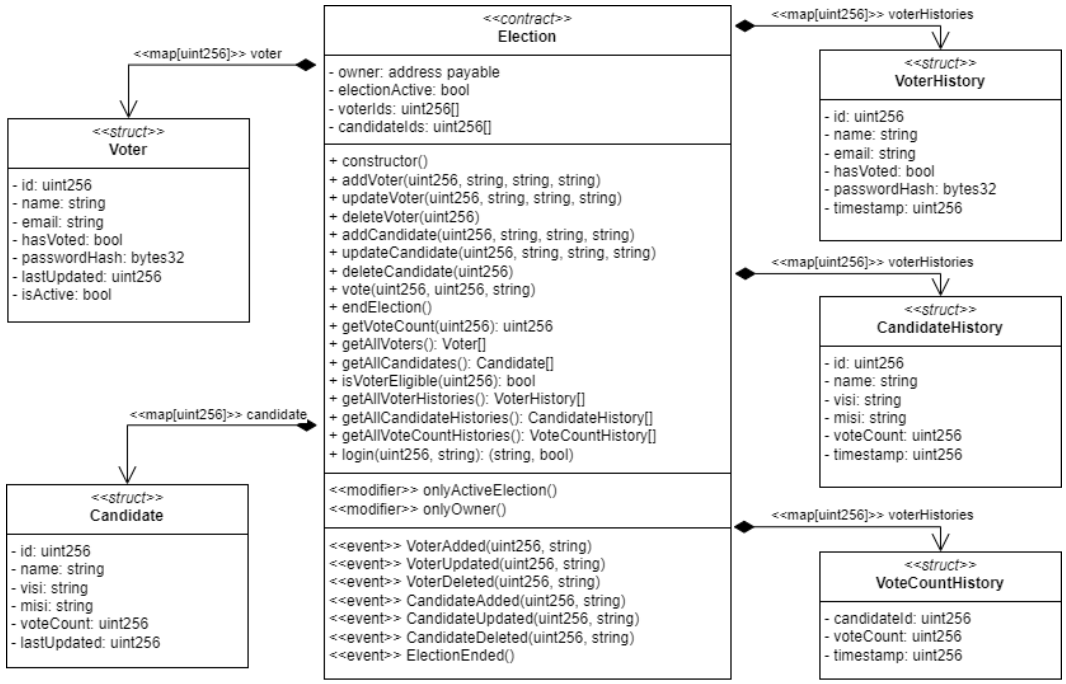

Figure 5: Smart Contract Design

- `getVoteCount(uint256)`: Returns the current vote count for a candidate.
- `getVoterHistory(uint256)`: Retrieves the voting history of a specific voter.
- `getVoteCounts()`: Returns the vote count history for all candidates.

These functions, along with the associated data structures, ensure the secure and transparent management of the voting process on the Ethereum blockchain.

5.5 Deployment to Ethereum Blockchain

Once developed, the smart contracts are deployed to the Ethereum blockchain[42]. Deployment involves broadcasting the contract's bytecode to the network. The contract is then included in a block and added to the blockchain. Deployment can be done using tools such as Truffle or Remix.

5.6 Interacting with Smart Contracts

After deployment, the smart contracts are interacted with through transactions. Users can send transactions to execute the functions defined in the contracts. For the e-voting system, this includes casting votes and tallying results.

6 Implementation of Mobile-based Platform

This section details the implementation process of the mobile-based platform in the research conducted. The implementation is segmented into various sub-sections, each describing a critical aspect of the system.

6.1 Platform Overview

The implementation of the system follows the design phase, focusing on developing the mobile application integrated with a private blockchain using Go Ethereum (Geth). Implementation involves several technical aspects, such as smart contract development, private blockchain integration, and user interface design. The program is developed using TypeScript, which is compatible with various mobile operating systems and integrates with the blockchain network through Geth.

6.2 Mobile Application Development

The development of mobile applications involves creating an intuitive user interface and integrating it with the blockchain network. The application is built using *React Native* and *TypeScript*, ensuring compatibility across different mobile platforms. Integration with *Geth* allows for secure and transparent voting processes.

6.2.1 Smart Contract Deployment

Smart contracts are deployed using the Hardhat RPC on a local network port. The following code snippet demonstrates the deployment process:

```
const hre = require("hardhat");

async function main() {
    const Voting = await hre.ethers.getContractFactory("Voting");
    const voting_ = await Voting.deploy();

    await voting_.deployed();

    console.log(
        `Contract Address: ${voting_.address}`
    );
}

main().catch((error) => {
    console.error(error);
    process.exitCode = 1;
});
```

Figure 6: Deploy Smart Contract Using Hardhat

6.2.2 User Interface Development

The user interface of the blockchain-based e-voting system is designed to be highly user-friendly, facilitating seamless interaction between voters and administrators. The login screen features a clean and intuitive layout where users can easily authenticate by entering their unique identifier and password, which are verified against the information stored in the smart contract. Serving as the central hub for the application, the home screen combines the functionalities of the result and history screens, providing users with current vote counts for each candidate while maintaining a transparent record of all votes cast, thus enabling easy access to complete voting histories and enhancing the overall user experience. The voting screen further streamlines the process by displaying a list of candidates with their respective names, IDs, visions, and missions, allowing users to select their preferred candidate confidently before confirming their choice through a dialog prompt. In general, the development of the user interface prioritizes creating a seamless and intuitive experience, ensuring the accessibility and usability of the e-voting system for all participants.

6.2.3 Integration with Geth

The mobile application is integrated with the Go Ethereum (Geth) private network to establish a secure and transparent connection to the blockchain. This integration includes:

- **Connecting to the Geth Node**: The application establishes a connection with the Geth node, allowing it to interact with the blockchain network and access the necessary data for the voting process.
- **Handling Transactions**: The application manages the creation, signing, and submission of Ethereum transactions to the Geth node, ensuring that votes are securely recorded on the blockchain.
- **Monitoring Blockchain Events**: The application listens for and processes relevant events emitted by the smart contract, such as the successful recording of a vote, to provide real-time updates and feedback to the users.

- **Retrieving Blockchain Data**: The application retrieves data from the blockchain, including voter information, candidate details, and voting history, to display the necessary information to users.

7 Evaluation

The evaluation of this blockchain-based e-voting system covers four main areas: testing scenarios, system functionality, performance, and security and transparency. This evaluation aims to assess the reliability, efficiency, and robustness of the system.

7.1 Testing Scenarios

The test scenario begins with blockchain testing, which involves checking the integrity of validator nodes, preventing unauthorized access, and data transparency. Testing shows that when a validator node is closed, attempts to add data to the blockchain are unsuccessful, confirming the important role of validators in maintaining network security. The system successfully blocks all attempts to access the network using unregistered private keys and member node private keys, demonstrating an effective access control mechanism. Data transparency testing confirms that authorized users can view voting data using member node private keys even when the validator node is down, ensuring transparency while maintaining security.

Smart contract testing focuses on verifying deployment, voter management, and candidate management. This test confirms the correct ownership and election status of the contract after deployment. The processes for registering, updating, and deleting voter data, as well as adding, updating, and deleting candidate data, all function as intended. System testing includes user authentication, voting simulation, and report counting and reporting of results. The login process successfully distinguishes between authorized and unauthorized users, the voting simulation works as intended with double voting prevention, and the vote counting and the display of results are verified to be accurate and timely.

7.2 System Functionality

Functionality testing focused on ensuring that all core features of the e-voting system worked as intended, following the election procedure outlined earlier.

7.2.1 Open the App

Verification of the ability of the application to launch successfully on various mobile devices and operating systems was a critical first step. This process also confirmed the establishment of a proper connection to the Go Ethereum (Geth) private network. Additional tests included evaluating the app's responsiveness and initial load times, as well as ensuring proper synchronization with the blockchain to maintain up-to-date information.

7.2.2 Get Credentials

Comprehensive testing was performed on the user authentication process, ensuring that it operated correctly. The usability and clarity of the login screen UI were evaluated, alongside the verification of credential checks against the smart contract on the blockchain. The handling of errors was tested for incorrect login attempts and the secure storage and transmission of user credentials was also verified.

7.2.3 Vote Candidates

The voting interface was meticulously evaluated to ensure the correct and complete display of candidate information, including names, IDs, visions, and missions. The proper functioning of the candidate selection mechanism was verified, ensuring that only one selection was possible. The confirmation dialog was tested to accurately present the selected candidate, and the responsiveness and accessibility of the interface were also assessed on different screen sizes.

7.2.4 Verification

The pre-vote verification process was thoroughly tested to ensure that the system correctly identified whether a user had already voted in a particular election. The password re-entry requirement was tested for proper functioning, including error handling for incorrect entries. The system's ability to accurately prepare

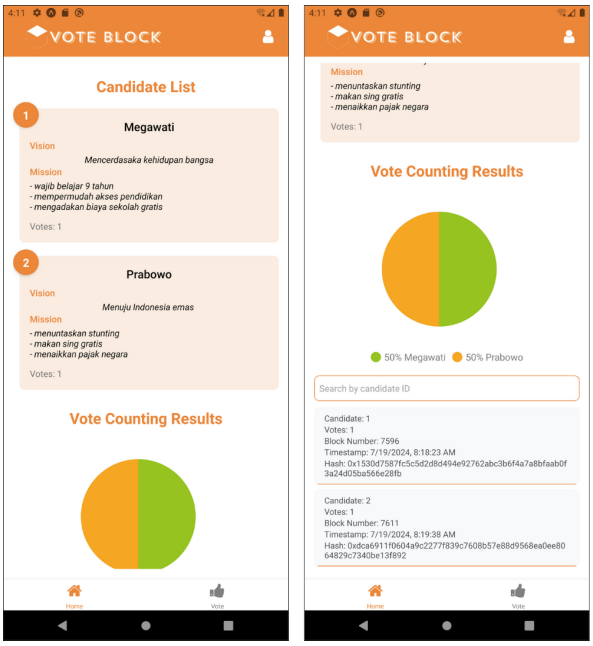

Figure 7: Home Screen

the voting transaction, ensuring that all necessary data was included, was also verified. Additionally, the correct association of the user's blockchain address with the vote was confirmed.

7.2.5 History Data

The system's ability to correctly maintain and display voting history data was evaluated. Voter privacy was a key concern and its preservation in history records was verified. The process of retrieving historical data from the blockchain was tested and the clarity and completeness of the history data display interface was thoroughly assessed.

7.3 Functionality Test Outcomes

The functionality of the e-voting application was thoroughly tested to ensure a secure, seamless, and user-friendly voting experience. Each critical phase of the voting procedure was assessed, with results summarized as follows.

7.3.1 Application Launch

The application successfully launched across all tested devices, establishing a stable connection to the Go Ethereum (Geth) private network. This ensured real-time syncing with blockchain data, providing users with the latest information as they initiated the voting process.

7.3.2 User Authentication

Authentication tests confirmed the security and effectiveness of user verification. Valid credentials were securely matched against smart contract records, ensuring that only eligible voters could proceed to the voting interface. Incorrect attempts were properly blocked.

7.3.3 Candidate Selection

Post-authentication, users navigated smoothly through the candidate selection phase. The voting screen accurately displayed candidate details, and the confirmation dialog functioned correctly, preventing accidental submissions and verifying voter selections.

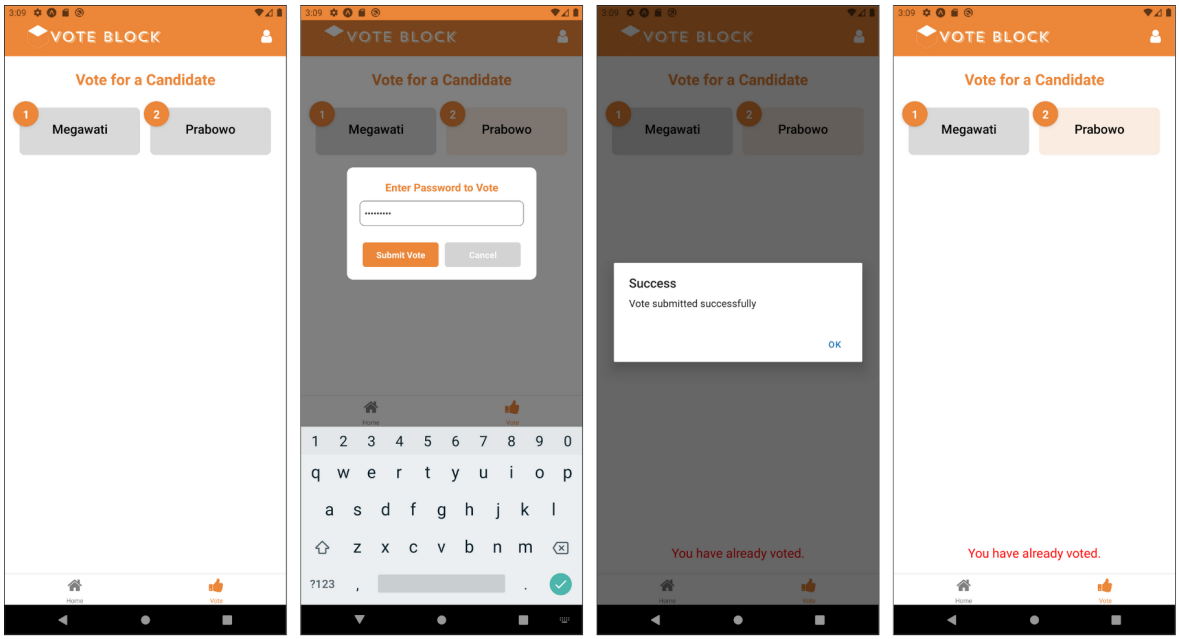

Figure 8: Vote Screen

7.3.4 Vote Verification

The system's verification checks effectively upheld the "one person, one vote" principle. Users re-entered their passwords for confirmation, which successfully ensured the integrity of each vote prior to blockchain submission. Security measures were robust and satisfactory.

7.3.5 Blockchain Recording

Votes were securely transmitted and stored on the blockchain following successful verification. The application managed Ethereum transactions efficiently, confirming validity via Geth nodes. Users received prompt notifications upon the successful recording of their votes.

7.4 Performance

Performance testing demonstrated that the blockchain-based e-voting system operates by its intended functions. All components of the system, including user authentication, voting process, and result tagging, operated as expected. The blockchain network proved capable of storing voting data securely and consistently. Although there was a slight delay when the system was first initiated, primarily due to the blockchain network initialization process and initial synchronization between nodes, this did not interfere with the overall functionality of the system. After initialization, the system ran smoothly, capable of handling the voting process and data storage without significant constraints.

It's important to note that while there was a slight delay when starting the system, this is a common characteristic of blockchain networks and does not affect the integrity or security of the stored data. The system's performance remained stable even when handling higher transaction loads during peak voting periods.

7.5 Security and Transparency

The security and transparency tests are critical to ensuring the integrity and openness of the election process, as explained in Table 1. Rigorous verification confirmed that voting data stored on the blockchain remain immutable and voter privacy is maintained by ensuring that individual votes cannot be traced back to a specific voter. The access control and authentication mechanisms proved to be effective in preventing

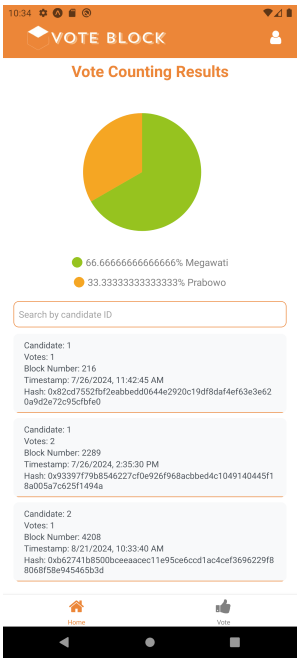

Figure 9: Data Stored in Blockchain Displayed on Home Page

unauthorized access. A detailed audit of the smart contract code identified and addressed potential vulnerabilities. Data transparency testing confirmed that voting data could be accessed and viewed using member node private keys, even when validator nodes were down.

7.6 User Feedback

To measure the effectiveness and ease of use of the e-voting system, a survey has been conducted among 10 participants who used the app during a mock election. The feedback collected provided valuable information on the user experience and the perceived reliability of the system. Most of the users (80%) found the interface of the app intuitive and easy to navigate. They particularly appreciated the clear instructions provided at each step of the voting process. The ability to view candidate information within the app was highlighted as a useful feature by 70% of participants. Regarding the voting process itself, 90% of users cited the voter history data provided after voting as a major factor in building this trust. However, some users (10%) expressed initial concerns about the privacy of their votes, although these concerns largely diminished after explaining the security measures of the blockchain in the system.

Most of the users (70%) reported greater confidence in the election process when using the blockchain-based system, citing the transparency and immutability of the blockchain as key factors. Many users (60%) also appreciated the ability to verify their votes after the election, although some (20%) found this process a bit cumbersome and suggested simplifying the verification steps. The constructive feedback included suggestions for additional features, such as notifications for election updates (requested by 40% of the users) and a more detailed explanation of how blockchain technology ensures the security of the vote (requested by 30% of users).

Overall, user feedback was largely positive, with 90% of the participants stating that they would prefer the blockchain-based electronic voting system over traditional methods in future elections. This feedback not only validated the user-centric design of the system but also provided valuable direction for future feature improvements and enhancements. The results of this comprehensive evaluation, including positive user feedback, demonstrate that the implemented blockchain-based e-voting system effectively addresses the key challenges of security, transparency, and efficiency in the electronic voting process, while providing a user-friendly experience. The system successfully maintains data integrity and voter privacy while ensuring transparency, providing a robust and well-received solution to modern election needs.

Table 1: Security and Transparency Testing of Blockchain Network

Test Case	**Testing Scenario**		
	Input	***Expected Result***	***Result***
Network Security Testing	Shutting down validator node and attempting to add data	An error occurs, and data cannot be added	Complies
	Replacing private key with an unregistered private key on the network	An error occurs, and data cannot be added	Complies
	Replacing private key with a member node's private key	An error occurs, and data cannot be added	Complies
Network Transparency Testing	Replacing private key with a member node's private key	Data can be displayed	Complies
	Shutting down validator node and attempting to access data with a member node's private key	Data can be displayed	Complies

[a]Testing is conducted under controlled conditions.

8 Discussion on Evaluation Results

This section discusses on evaluation results with the obstacles in system implementation.

8.1 Summary of Evaluation Results

The system functionality tests of the e-voting application focused on validating each component of the election process, ensuring that it adhered to the high standards of user-friendliness and security. The initial step involved confirming that the application could launch seamlessly on various devices, ensuring cross-platform compatibility and connectivity to the Go Ethereum (Geth) private network. This stage was crucial, as it established a foundation for the application to consistently relay up-to-date blockchain data. Once the application was launched, the authentication system was scrutinized to guarantee secure access. The user interface was evaluated for clarity and ease of use, ensuring that users could log in efficiently using verified credentials. This step also involved testing error responses to incorrect credentials, a critical aspect of maintaining system integrity.

Further testing of the voting process verified the comprehensive display and selection of candidates. Each component of the interface was evaluated for responsiveness across different devices, ensuring accessibility. The system's verification protocols were examined to enforce the "one person, one vote" principle, checking for duplicate submissions, and confirming the correct handling of user input before votes were securely recorded on the blockchain. The tests confirmed that the votes were transmitted and stored securely, leveraging the immutability of the blockchain to validate each transaction. Additionally, the system's capability to maintain historical data while preserving voter privacy was confirmed, allowing for secure data retrieval and accurate data visibility post-election. These tests ensured that the system fulfilled its intended functions, providing a seamless and transparent voting experience that upheld the integrity and confidentiality of all participants' votes.

8.2 Obstacles in System Implementation

One of the significant obstacles related to system limitations in implementing a mobile voting application using blockchain technology is the inherent complexity and technical demands of blockchain itself. The procedure involves intricate steps such as encryption, transaction validation, and blockchain synchronization,

which can strain both the mobile devices and the back-end infrastructure. In particular, during peak voting times, the system could face performance bottlenecks due to limited network bandwidth and node processing capabilities. Additionally, the blockchain's immutability, a strength in ensuring vote integrity, also means that any bugs or errors in deployed smart contracts could be difficult to rectify without redeploying a new contract, potentially disrupting the voting process.

Scalability is another pressing concern when deploying a blockchain-based voting system. As more users participate in the system, the volume of transactions grows significantly, which can lead to congestion and increased latency in processing transactions. Blockchain networks like Ethereum are notorious for scalability issues, where transaction speeds can slow down and costs increase as the number of users scales up. Implementing a private Ethereum network mitigates some of this by controlling network traffic, but this approach can still face scalability hurdles, particularly in large-scale elections or elections with unexpected surges in voting activity. Solutions such as optimizing consensus algorithms and off-chain transactions may offer some relief, but implementing these on a scale remains a technical challenge.

Finally, user adoption poses its own set of challenges. The general public might have reservations about using blockchain-based systems due to a lack of familiarity and understanding of the technology. Users may be hesitant to engage with a voting method they perceive as complex or nonintuitive, which could hinder widespread adoption. Moreover, ensuring accessibility across diverse demographic groups, including those less tech-savvy, requires thoughtful design and thorough user education. Implementation challenges also include ensuring that all potential voters have access to the necessary technology, such as smartphones and reliable Internet connections, particularly in rural or underdeveloped areas. To overcome these barriers, robust educational campaigns and user-friendly design are crucial to ensuring that the system appeals to and is usable by the widest possible audience.

9 Conclusion

The integration of blockchain technology into electronic voting systems offers substantial improvements in security, transparency, and trustworthiness, but implementing such systems presents both technical and user-centric challenges. The development of the described e-voting application focused heavily on maintaining user-friendliness while enhancing security. Rigorous testing confirmed the application's ability to launch smoothly on various devices, ensuring consistent performance across platforms. The app's connection to the Go Ethereum (Geth) private network was critical to providing real-time blockchain data access, essential for maintaining the transparency of the voting process. The successful authentication of users reinforced access security, with the user interface designed to help users efficiently log in and navigate the voting procedure, correctly handling incorrect login attempts to maintain system integrity.

Further evaluations of the voting interface emphasized accessibility and clarity, confirming that users could view and select candidates with ease. The verification protocols tested ensured the enforcement of the one-person-one-vote principle, eliminating duplicate submissions, and verifying the correct recording of votes on the blockchain. This system confirmed the security and immutability of vote storage, allowing for reliable post-election data retrieval without compromising voter privacy. These extensive tests upheld the system's capacity to deliver a seamless and transparent voting experience, emphasizing data integrity and user confidentiality, which are critical components of modern electronic voting systems.

Despite the achievements in system implementation and its successful rigorous testing, several obstacles were identified. Technical complexities, such as blockchain synchronization and inherent computational demands, posed challenges, particularly during high transaction loads. The immutability of blockchain complicates error correction once smart contracts are deployed, requiring meticulous planning and testing before implementation. Scalability remains a significant hurdle, as transaction volumes increase with user participation, potentially leading to network congestion and latency. Managing these challenges requires advanced solutions, such as optimizing consensus protocols and incorporating off-chain transactions to effectively handle workload distribution. Additionally, fostering user adoption presents challenges due to a lack of public familiarity with blockchain technology, necessitating comprehensive educational efforts and intuitive design to bridge the gap. These solutions must consider accessibility to ensure inclusivity between diverse populations, addressing potential technological limitations, particularly in underserved areas. Robust educational initiatives and user-friendly system designs are essential to enhance engagement and ensure broad acceptance of such advanced voting technologies.

Moving forward, future work should focus on expanding the system's capabilities and enhancing user engagement. This could involve integrating additional features such as real-time election updates, simplified vote verification procedures, and improved user feedback mechanisms to further streamline the voting experience.

References

[1] S. Makani, R. Pittala, E. Alsayed, M. Aloqaily, and Y. Jararweh, "A survey of blockchain applications in sustainable and smart cities," *Cluster Computing*, vol. 25, 2022.

[2] Q. Tang, H. Shang, Y. Kong, and S. Sun, "Promoting the digital transformation of the payment industry based on blockchain technology," *International Journal of Education and Humanities*, vol. 11, 2023.

[3] J. H. Hsiao, R. Tso, C. M. Chen, and M. E. Wu, "Decentralized e-voting systems based on the blockchain technology," in *Lecture Notes in Electrical Engineering*, vol. 474, 2018.

[4] J. El-Gburi, G. Srivastava, and S. Mohan, "Secure voting system for elections," *International Journal of Computer Aided Engineering and Technology*, vol. 16, 2022.

[5] A. Singh, A. Ganesh, R. R. Patil, S. Kumar, R. Rani, and S. K. Pippal, "Secure voting website using ethereum and smart contracts," *Applied System Innovation*, vol. 6, 2023.

[6] Getch.Ethereum, "Home — go-ethereum," 2022.

[7] A. Panigrahi, A. K. Nayak, and R. Paul, "Smart contract assisted blockchain based public key infrastructure system," *Transactions on Emerging Telecommunications Technologies*, vol. 34, 2023.

[8] CoinMarketCap, "What is ethereum (eth)?" *CoinMarketCap*, 2021.

[9] S. Myung and J. H. Lee, "Ethereum smart contract-based automated power trading algorithm in a microgrid environment," *Journal of Supercomputing*, vol. 76, 2020.

[10] G. Schmitt, A. Mladenow, C. Strauss, and M. Schaffliauser-Linzatti, "Smart contracts and internet of things: A qualitative content analysis using the technology-organization-environment framework to identify key-determinants," in *Procedia Computer Science*, vol. 160, 2019.

[11] Y. W. Syaifudin, Y. Yunhasnawa, Y. Pramitarini, A. Setiawan, E. Rohadi, and P. Y. Saputra, "A proposed framework of campus-oriented online text messaging system," *International Journal of Interactive Mobile Technologies*, vol. 14, 2020.

[12] A. Susanto, "Smart contract blockchain pada e-voting," *Jurnal Informatika Upgris*, vol. 5, 2019.

[13] J. G. Song, S. J. Moon, and J. W. Jang, "A scalable implementation of anonymous voting over ethereum blockchain," *Sensors*, vol. 21, 2021.

[14] G. A. Oliva, A. E. Hassan, and Z. M. J. Jiang, "An exploratory study of smart contracts in the ethereum blockchain platform," *Empirical Software Engineering*, vol. 25, 2020.

[15] M. Ali and S. Bagui, "Introduction to nfts: The future of digital collectibles," *International Journal of Advanced Computer Science and Applications*, vol. 12, 2021.

[16] B. B. A. Christyono, M. Widjaja, and A. Wicaksana, "Go-ethereum for electronic voting system using clique as proof-of-authority," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, 2021.

[17] B. Academy, "Proof of authority explained," *Binance Academy*, 2020.

[18] X. Qiu, Z. Qin, W. Wan, J. Zhang, J. Guo, S. Zhang, and J. Xia, "A dynamic reputation-based consensus mechanism for blockchain," *Computers, Materials and Continua*, vol. 73, 2022.

[19] G. Rathee, R. Iqbal, O. Waqar, and A. K. Bashir, "On the design and implementation of a blockchain enabled e-voting application within iot-oriented smart cities," *IEEE Access*, vol. 9, 2021.

[20] P. A. Sankhe, "Implementation of decentralized blockchain e-voting," *International Scientific Journal of Engineering and Management*, vol. 02, 2023.

[21] F. P. Hjalmarsson, G. K. Hreioarsson, M. Hamdaqa, and G. Hjalmtysson, "Blockchain-based e-voting system," in *IEEE International Conference on Cloud Computing, CLOUD*, vol. 2018-July, 2018.

[22] M. H. Berenjestanaki, H. R. Barzegar, N. E. Ioini, and C. Pahl, "Blockchain-based e-voting systems: A technology review," *Electronics (Switzerland)*, vol. 13, 2024.

[23] B. M. B. Pereira, J. M. Torres, P. M. Sobral, R. S. Moreira, C. P. de Almeida Soares, and I. Pereira, "Blockchain-based electronic voting: A secure and transparent solution," *Cryptography*, vol. 7, 2023.

[24] C. G. Schmidt and S. M. Wagner, "Blockchain and supply chain relations: A transaction cost theory perspective," *Journal of Purchasing and Supply Management*, vol. 25, 2019.

[25] G. Revathy, K. B. Raj, A. Kumar, S. Adibatti, P. Dahiya, and T. M. Latha, "Investigation of e-voting system using face recognition using convolutional neural network (cnn)," *Theoretical Computer Science*, vol. 925, 2022.

[26] A. Shukla, D. P. Mishra, A. Pattnaik, and S. R. Salkuti, "Analysis and design on acceptance of blockchain based e-voting system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, 2024.

[27] M. S. Farooq, U. Iftikhar, and A. Khelifi, "A framework to make voting system transparent using blockchain technology," *IEEE Access*, vol. 10, 2022.

[28] C. Vaidya, C. Kirnapure, J. Rithe, D. Sonkusare, P. Khade, and K. Kharche, "An approach towards decentralized e-voting," in *International Conference on Emerging Trends in Engineering and Technology, ICETET*, vol. 2023-April, 2023.

[29] B. Kaynak, S. Kaynak, and Özer Uygun, "Cloud manufacturing architecture based on public blockchain technology," *IEEE Access*, vol. 8, 2020.

[30] P. Bellavista, C. Esposito, L. Foschini, C. Giannelli, N. Mazzocca, and R. Montanari, "Interoperable blockchains for highly-integrated supply chains in collaborative manufacturing," *Sensors*, vol. 21, 2021.

[31] J. Polge, J. Robert, and Y. L. Traon, "Permissioned blockchain frameworks in the industry: A comparison," *ICT Express*, vol. 7, 2021.

[32] A. Alshehri, M. Baza, G. Srivastava, W. Rajeh, M. Alrowaily, and M. Almusali, "Privacy-preserving e-voting system supporting score voting using blockchain," *Applied Sciences (Switzerland)*, vol. 13, 2023.

[33] N. A. Asad, M. T. Elahi, A. A. Hasan, and M. A. Yousuf, "Permission-based blockchain with proof of authority for secured healthcare data sharing," in *2020 2nd International Conference on Advanced Information and Communication Technology, ICAICT 2020*, 2020.

[34] V. George and M. P. Sebastian, "A secure and efficient scheme for remote poll station voting," *International Journal of Electronic Government Research*, vol. 9, 2013.

[35] F. Z. Chentouf and S. Bouchkaren, "Security and privacy in smart city: a secure e-voting system based on blockchain," *International Journal of Electrical and Computer Engineering*, vol. 13, 2023.

[36] M. V. Vladucu, Z. Dong, J. Medina, and R. Rojas-Cessa, "E-voting meets blockchain: A survey," *IEEE Access*, vol. 11, 2023.

[37] A. Bengtson, "One person, one vote and the importance of baseline," *Inquiry (United Kingdom)*, 2022.

[38] L. Chacko, P. Rajan, V. Anilkumar, and V. Pathari, "Proof-of-variable-authority: A blockchain consensus mechanism for securing iot networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14424 LNCS, 2023.

[39] S. Rouhani and R. Deters, "Performance analysis of ethereum transactions in private blockchain," in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, vol. 2017-November, 2017.

[40] G. Estevam, L. M. Palma, L. R. Silva, J. E. Martina, and M. Vigil, "Accurate and decentralized timestamping using smart contracts on the ethereum blockchain," *Information Processing and Management*, vol. 58, 2021.

[41] G. Zheng, L. Gao, L. Huang, and J. Guan, *Ethereum Smart Contract Development in Solidity*, 2020.

[42] A. Pinna, S. Ibba, G. Baralla, R. Tonelli, and M. Marchesi, "A massive analysis of ethereum smart contracts empirical study and code metrics," *IEEE Access*, vol. 7, 2019.