Journal of Theoretical and Applied Information Technology
15th September 2024. Vol.102. No. 17
© Little Lion Scientific

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

THE DEVELOPMENT OF A BLOCKCHAIN-BASED SYSTEM FOR ELECTRONIC VOTING

[1]EMAN-YASSER DARAGHMI, [2]AHMED HAMOUDI

[1]Department of Computer Science, Palestine Technical University, Tulkarm, Palestine

[2]Department of Graduate Studies, Palestine Technical University, Tulkarm, Palestine

E-mail: [1] e.daraghmi@ptuk.edu.ps, [2]hammoudi00@gmail.com

ABSTRACT

Elections and voting play a crucial role in the development of a democratic society, enabling the public to express their views and participate in the decision-making process. Voting methods have evolved from paper ballot systems to e-voting systems to preserve the integrity of votes, ensuring a secure, transparent, and verifiable process. Continuous efforts have been made to develop a secure e-voting system that eliminates fraud attempts and provides accurate voting results. In this paper, we propose the architecture of a blockchain-based e-voting system called VoteChain. Developed to support the existing voting system in the state of Palestine, VoteChain aims to provide secure e-voting with features such as auditability, verifiability, accuracy, privacy, flexibility, transparency, mobility, availability, convenience, data integrity, and distribution of authority. The work introduces a smart contract designed to meet the demands of e-voting, governing transactions, monitoring computations, enforcing acceptable usage policies, and managing data usage after transmission. The proposed system also adopts advanced cryptographic techniques to enhance security. VoteChain features a web-based interface to facilitate user interaction, providing protection against multiple or double voting to ensure the integrity of the election. Furthermore, VoteChain is designed with a user-friendly and easily accessible administrator interface for managing voters, constituencies, and candidates. It ensures equal participation rights for all voters, fostering fair and healthy competition among candidates while preserving voter anonymity.

Keywords: *Blockchain, Smart Contracts, Consensus, E-voting, Privacy, Security.*

1. INTRODUCTION

Elections and voting play a pivotal role in the development of a democratic society by allowing the public to express their views and participate in the decision-making process, ensuring equal rights and fair representation. To maintain the credibility of participants, the election and voting processes must be reliable and transparent [1], [2].

Over time, voting methods have evolved from traditional paper ballot systems to e-voting systems, aiming to preserve integrity and establish a secure, transparent [3,4], and verifiable system. Ongoing efforts focus on improving overall efficiency and resilience. Traditional voting, conducted at specific polling stations, incurs significant costs and time. Consequently, e-voting has emerged as a replacement, minimizing election costs and ensuring integrity by addressing privacy, security, and compliance requirements. E-voting leverages cryptographic techniques, enabling full features on common household devices and ensuring instant and anonymous vote counting.

For the widespread adoption of e-voting systems, adherence to benchmark parameters, including voter anonymity, vote integrity, and non-repudiation, is crucial. However, e-voting poses challenges, particularly in addressing concerns such as electoral fraud (e-Fraud) and voter manipulation.

Blockchain technology, with its strong cryptographic foundations, provides a potential solution [5], [6], [7], [8]. A blockchain-based e-voting system can mitigate cybercrimes, enhance verifiability, and ensure traceability, offering a cost-effective, convenient, and secure solution. Blockchain's immutability, decentralization, and distributed consensus through advanced cryptography enhance security beyond traditional record-keeping systems.

Several researchers, including [3], [9] , have proposed blockchain-based e-voting systems with features such as embedded smart contracts, blind signatures, and homomorphic encryption. Despite these efforts, further research is needed to fully understand, characterize, and evaluate the utility of blockchain-based e-voting systems.

This paper proposes the architecture of VoteChain, a blockchain-based e-voting system developed to support the existing voting system in Palestine. VoteChain aims to provide secure e-voting with auditability, verifiability, accuracy, privacy, security, flexibility, transparency, mobility, availability, convenience, data integrity, and distribution of authority.

The proposed smart contract [10] design meets the demands of e-voting, governing transactions, monitoring computations, enforcing acceptable usage policies, and managing data usage after transmission. Advanced cryptographic techniques further enhance security. VoteChain features a user-friendly web-based interface, providing protection against multiple or double voting to ensure election integrity. The system also includes an easily accessible administrator interface for managing voters, constituencies, and candidates while preserving voter anonymity and promoting fair competition.

2. RELATED WORK

For a long time, researchers have been working on developing secure and efficient e-voting protocols [11]. In 2019, Lai et al. [12] proposed a decentralized ETH-based e-voting system providing voters with a secure and anonymous voting experience. All voting requirements are embedded inside the ETH blockchain; thus, transparency is ensured in the voting process. The instructions are carried out in accordance with the integrated codes on the smart contract and Ethereum's network ensures transparency and security throughout the voting process. Each person interested in the election can access the related information openly through the Ethereum blockchain by moving data and calculations into an intelligent contract. The entire election is therefore decentralized and trustworthy.

In 2019, Patil et al. [13] proposes a blockchain-based voting system that uses smart contracts to provide secure and cost-effective election. It is based on an ETH private blockchain. The system allows users to send and receive transactions without being tied to a central server, by enable users to register in order to vote on thy proposed service, use the Recognition devices & valid ID number to verify whether or not the user is in the database eligible or not to vote. Then, the voter receives a unique hash address to cast a vote. Each hash has Ethers that he can cast votes once. The voter will visit a polling station during the day on which he is verified, and then cast a vote with an address he\she received and then log out. The voters also receive live voting status.

In 2019, Shahzad and Crowcroft [14] proposed a frame- work that can solve the problems of the voting process and provide a secure and tamper-resistant environment for the e-voting process. The blockchain can be adjusted to meet the varying requirements of the voting process.

In 2020, Mat et al. [15] proposed Feedback mechanisms to implement a blockchain voting system that can provide effective constraints on malicious voting behavior. The Wilson score is used as the initial support rate for each candidate.

In 2020, Prasetyadi et al. [16] proposed system will be built according to the principles of Indonesian voting. Each vote was verified using the Elliptic Curve Digital Signature Algorithm (ECDSA) instead of the RSA signature and its associated data was mined into blocks.

In 2021, Taş and Tanriöver [17] proposes a double-layer security model that can prevent fraud and minimize the effects of manipulation during the elections. In 2019, Awalu et al. [3] proposed system includes a multichain Blockchain network, arbitration server, distributed database, interactive, multi-device GUI, and application server. To promote anonymity, Digital Signature and Secure Hash Algorithm SHA were proposed, together with Proof of Work PoW. Election stages were compared to eVoting requirements. The suggested system met eVoting requirements such as transparency, privacy, scalability, receipt-freeness, security, integrity, correct- ness, auditable, etc. In 2018, Khan et al. [18] proposed system achieves end-to-end verifiability and complies with the key criteria for e-voting schemes. The article provides information on the proposed electronic voting system and how it will be implemented utilizing the Multichain platform. The paper also pro- vides a thorough analysis of the plan, demonstrating its value in achieving an end-to-end verified e-voting plan. In 2018, Yavuz et al. [19] tested a prototype e-voting application as an Ethereum smart contract utilizing Ethereum wallets and Solidity. An- droid could allow individuals without Ethereum wallets to vote. Ethereum blockchain will keep ballots and votes after an election. Users can vote via Android or via their Ethereum wallets, and every Ethereum node handles these transaction requests. This consensus makes e-voting transparent.

3. VOTECHAIN: SYSTEM ANALYSIS AND DESIGN

This section introduces the innovative Blockchain-based system, VoteChain, designed to fortify the existing voting system in the state of Palestine. The primary objectives of VoteChain encompass not only the prevention of fraud and vote manipulation but also the provision of secure e-voting, ensuring privacy, eligibility, convenience, receipt-freeness, and verifiability [20] [21] [22] [23] [24] [25] [26] [27] [28]. VoteChain is equipped with a user-friendly web-based interface, streamlining user interactions within the system. This interface serves as a crucial tool in safeguarding the integrity of the election process. It offers protection against multiple voting or double voting, a key feature essential for maintaining the fairness and trustworthiness of the electoral procedure. Additionally, VoteChain incorporates a user-friendly and easily accessible administrator interface. This interface empowers administrators to efficiently manage voters, constituencies, and candidates for constituencies. The design prioritizes ease of use, ensuring that administrators can navigate and oversee the system seamlessly.

VoteChain upholds the principles of equal rights, fostering universal participation among all voters. Its architecture is structured to develop a fair and healthy competition among candidates, simultaneously preserving the anonymity of voters. This ensures that the electoral process remains unbiased and inclusive.

Fig.1 illustrates the robust architecture of the VoteChain system. The architecture is meticulously crafted to encompass multiple layers of security and functionality, aligning with the system's overarching goals. Each component plays a crucial role in ensuring the system's resilience against fraudulent activities and manipulation.

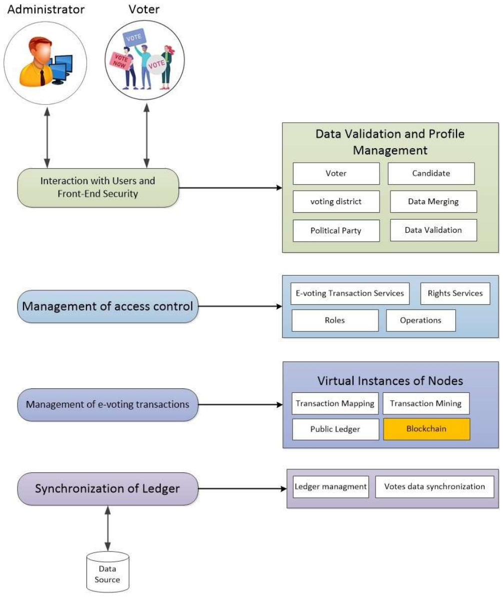

Administrator
Voter
VOTE
VOTE NOW
VOTE
Interaction with Users and Front-End Security
Data Validation and Profile Management
Voter
Candidate
voting district
Data Merging
Political Party
Data Validation
Management of access control
E-voting Transaction Services
Rights Services
Roles
Operations
Management of e-voting transactions
Virtual Instances of Nodes
Transaction Mapping
Transaction Mining
Public Ledger
Blockchain
Synchronization of Ledger
Ledger managment
Votes data synchronization
Data Source

Figure 1. Architecture for proposed e-voting system

3.1. VoteChain design: a comprehensive overview

The VoteChain design embodies a robust infrastructure tailored for real-world voting applications, prioritizing critical factors such as auditability, verifiability, accuracy, privacy, security, flexibility, transparency, mobility, availability, convenience, data integrity, and distribution of authority. To facilitate the seamless administration of voters, constituencies, and candidates, a user-friendly interface has been meticulously crafted for administrators. This interface not only simplifies administrative tasks related to the election process but also ensures that all voters enjoy equal participation rights. This inclusive approach fosters fair competition among candidates while preserving the vital element of voter anonymity.

Figure.1 illustrates the proposed architecture of VoteChain system. The User Interface and Front-End Security play a pivotal role in guiding voters through the voting process and assisting administrators with election-related tasks. This component serves dual functions: authenticating and authorizing users, voters, and administrators. Utilizing various security measures, from conventional "username and password" authentication to advanced "one-time password" (OTP) methods, the User Interface is the primary interaction point where user credentials are validated based on system-specific policies.

The Access Control Management Level acts as a facilitator for User Interaction and Front-End Security and E-Voting Transaction Management levels by providing essential services. These services encompass defining roles, managing access control, and handling voting transactions. Role definition and management specifically support access control tasks, while voting transaction definitions contribute to blockchain-based transaction mapping and mining.

At the core of the architecture lies the E-Voting Transaction Management Level. Here, the e-voting transaction, established at the Access Control Management Level, is mapped onto the transaction to be mined on the blockchain. Voter credentials obtained at the User Interaction and Front-End Security Level are incorporated into this mapped transaction. This step is crucial in generating the cryptographic hash and, ultimately, the transaction ID. Certification processes are anticipated to occur at the User Interaction and Front-End Security Level. The mining mechanism involves a multitude of virtual instances of nodes.

Ledger Synchronization Level: The Ledger Synchronization Level ensures synchronization between the local application and the multichain ledger. Votes are securely recorded on a blockchain ledger, utilizing cryptographic hashes to safeguard end-to-end communication. Voting results are computed through the smart contract and made public at the conclusion of the voting period. This multilayered approach ensures the integrity and security of the voting process within the VoteChain system.

3.2. E-voting blockchain definition

The design of the blockchain for electronic voting is rooted in Distributed Ledger Technology (DLT) [20] [10], [20], [21], [22], [23], [24], [25], [26], [27], [28]. Represented as a sequential chain of blocks, Figure. 2 illustrates the structure of the e-voting blockchain, starting with the foundational genesis block.

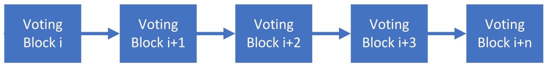


Figure 2. The Blockchain for E-Voting

Each block within this e-voting blockchain, as depicted in Figure. 3, encapsulates specific contents crucial to the integrity and security of the voting process. These contents include a voter's ID, the vote cast, the voter's signature, a timestamp, and a cryptographic digest (hash) of the previous block.

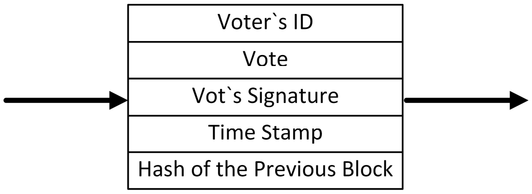


Figure 3. Voting Block

A voter ID is a unique identifier assigned randomly to an individual with the eligibility to vote. It identifies the voter exercising their right to vote for a preferred candidate. A vote is the act of casting a ballot in favor of the voter's preferred candidate. It represents the voter's choice in the election. Voter's signature is a mark made by the voter on the voting ballot, serving as a signature. It ensures the confidentiality of the vote by preventing others from discerning the voter's choice. The voter signs the hash of the vote with their private key, enabling subsequent verification of vote validity. A Timestamp records the exact submission time of the block. In cases where multiple blocks share the same timestamp, the one with the highest signature value takes precedence. A Digest (Hash) of the Previous Block is computed using the SHA-256 algorithm, this hash links the current block to the previous one in a tamper-resistant manner. This ensures the non-repudiation of transactions and protects against data tampering, making the e-voting blockchain secure and immutable.

VoteChain leverages DLT principles, employing a structured sequence of blocks to maintain the integrity, security, and transparency of the voting process. Each block encapsulates essential information, establishing a verifiable and tamper-resistant record of votes cast in the electronic voting system.

3.3. VoteChain: specifications and requirements

The foundational specifications of a generic e-voting system are delineated by Rura et al. [29]. This section delves into the critical criteria for e-voting, emphasizing how the proposed VoteChain system addresses these requirements.

✓ Privacy and Secrecy of a Voter's Ballot:

- Traditional Specification: Ensuring the privacy and secrecy of a voter's ballot is a fundamental requirement.
- VoteChain Enhancement: Voter anonymity is safeguarded through the cryptographic features of the blockchain. Upon voter registration in the VoteChain system, a unique hash is generated, serving as both the voter's identification and protection against misuse.

The hash, designed to resist collisions, adds an additional layer of complexity to traceability, enhancing the privacy of each vote.

- ✓ Eligibility – Permitting Only Registered Voters to Vote and Limiting Each Voter to One Vote:
  - Traditional Specification: Enforcing eligibility by permitting only registered voters to vote and restricting each voter to a single vote is crucial.
  - VoteChain Enhancement: The system mirrors the local election process by conducting registration and verification through national identities and additional personal data, such as phone numbers. Robust authentication, leveraging OTP technology, ensures that only registered voters access the system, preventing instances of double voting.
- ✓ Receipt Freedom – Voters Shouldn't Be Allowed to Show How They Voted to a Third Party:
  - Traditional Specification: Guaranteeing receipt freedom, preventing voters from revealing their votes, is imperative.
  - VoteChain Enhancement: The proposed method enables voters to cast their preferences, generating a cryptographic hash for each transaction. This hash is vital for verifiability, ensuring that a specific vote was counted. However, it deliberately prevents the extraction of vote details from the hash, maintaining the privacy and integrity of individual votes.
- ✓ Ease – Elections Must Be Simple for Voters, and All Eligible Citizens Must Be Able to Vote:
  - Traditional Specification: Ensuring simplicity for voters and universal accessibility is a standard requirement.
  - VoteChain Enhancement: VoteChain achieves this through an intuitive web-based interface, streamlining the voting process with minimal user interaction. The interconnected nature of the system allows users to seamlessly engage with the platform.
- ✓ Verifiability – Trust in the Voting Process:
  - Traditional Specification: Verifiability, the ability to trust the voting process, is essential.
  - VoteChain Enhancement: Post-voting, users receive a unique transaction identifier as a cryptographic hash, enabling them to verify whether their vote was counted. While this approach assures verifiability, it deliberately restricts individuals from discerning how they voted, a strategic measure to mitigate potential coercion or stress-related threats during the voting process.

The proposed VoteChain system not only aligns with the foundational specifications of a typical e-voting system but enhances key aspects to fortify privacy, security, and user confidence in the voting process.

4. VOTECHAIN IMPLEMENTATION

4.1 VoteChain: System Dependencies

Following points summarize the system dependencies.

- Node Package Manager (NPM) operates as a command-line utility and online repository, facilitating the release and management of open-source Node.js projects. This robust tool simplifies the installation and management of packages, offering access to a diverse array of applications and libraries within its extensive repository [30].
- The Truffle Framework empowers the creation of decentralized apps on the Ethereum network. This comprehensive framework includes a suite of tools for utilizing the Solidity programming language, facilitating the seamless development of smart contracts. Truffle supports the entire lifecycle of smart contracts, from testing to deployment on the blockchain. Additionally, it provides a dedicated environment for building client-side applications [31].
- Ganache, a local in-memory blockchain, is obtainable from the Truffle Framework website and easily installable. It furnishes 10 external accounts, each equipped with Ethereum addresses, on our local blockchain. Notably, each account is endowed with 100 bogus ether, ensuring a self-contained and convenient testing environment [32].
- Metamask, a browser extension, functions as an Ethereum wallet, offering users a secure platform for storing and conducting transactions on any Ethereum address. Its installation empowers seamless integration with Ethereum functionalities, enhancing user capabilities within the blockchain ecosystem [33].
- Development Environment (Visual Studio Code) serves as a versatile tool empowering developers to craft their customized development environment, often situated on a

remote server. This source code editor, rooted in the Node.js framework and sharing components with Azure DevOps, stands out by enabling users to seamlessly manage multiple directories, consolidating them into a singular, well-organized workspace [34].

- MongoDB, an open-source, cross-platform document-oriented database application, operates as a NoSQL database utilizing JavaScript Object Notation (JSON)-like documents with optional schemas. Developed by MongoDB Inc. and distributed under the Server Side Public License (SSPL), MongoDB offers a flexible and efficient approach to managing data [35].
- Twilio empowers developers to rapidly construct multi-channel solutions encompassing messaging, video, and email functionalities. Its versatile capabilities extend to seamless integration with various channels, enabling the creation of intelligent features and facilitating global scalability [36].

4.2. VoteChain: Smart Contracts and Implementation

VoteChain harnesses the Ethereum framework, an open-source blockchain-based platform for decentralized applications, coupled with cryptocurrency wallets like MetaMask. This synergy empowers both voters and candidates to seamlessly register and cast their ballots within the system. Oversight of the voting process is entrusted to the administrator of the Central Election Commission (CEC). Voters can engage in the ballot-casting process through either a full node, directly connected to the Ethereum Blockchain main chain, or a service node, facilitating accessibility through cloud services or wallets. Both node types connect to Ethereum, allowing voters to access the associated Decentralized Application (DApp) and contribute transactions to Ethereum's block chain.

The Casper technique, serving as the consensus mechanism, ensures the verification and validation of submitted transactions before undergoing processing through smart contracts [37].

Smart contracts form the backbone of the proposed VoteChain system, encompassing functionalities such as voter and candidate registration, vote casting and counting, and result declaration and announcement. These contracts, defined by the Solidity programming language, outline the rules that nodes must adhere to when executing operations, such as casting a ballot. Authentication, verification, and validation, integral to the consensus process, are seamlessly integrated into these smart contracts.

The accompanying Decentralized Application (DApp) boasts an intuitive Graphical User Interface (GUI) that facilitates user navigation through the voting processes. For authorized voters to submit a transaction on Ethereum's main chain, their credentials are meticulously vetted in an off-chain voters' database stored on the CEC server. This verification step is a prerequisite before any transaction is broadcast.

Transactions are treated as part of a private blockchain, ensuring they remain inaccessible to the public. With the voter's database managed by the CEC state unit and stored in a distributed database, concerns regarding accessibility and network traffic are effectively addressed within this innovative model.

Implementation relies on the MongoDB Atlas Cloud Database, a cloud-hosted database specifically chosen for storing voter credentials and related information submitted by the CEC. Furthermore, voters establish a private account on the Ethereum blockchain framework and the MetaMask wallet, serving as both a cryptocurrency wallet and a gateway to blockchain-based applications (DApp). This multi-layered approach enhances the security, accessibility, and efficiency of the VoteChain system.

4.2.1. VoteChain: Voting Process

Figures 4 and 5 visually outline a secure e-voting session orchestrated through VoteChain. The initial step in this voting journey involves user login to the website or Decentralized Application (DApp). Leveraging the voter credentials stored in the CEC database, user authentication is fortified through the use of a One-Time Password (OTP) delivered via SMS. Crucially, the outcome of a successful voter registration is safeguarded within the CEC database (off-chain), distinct from Ethereum's main chain (on-chain). This meticulous process serves to uphold the reliability and authenticity of voters throughout the voting procedure.

It is imperative to note that each voter's private account is established on the Ethereum framework and a service node, akin to a bitcoin wallet (MetaMask). Ganache, a personal Ethereum blockchain, facilitates testing and provides a view of on-chain transactions and blocks. Meanwhile, MetaMask functions as a service node and cryptocurrency wallet, enabling the payment of the requisite Ether (ETH) gas fee to cast a vote.

Upon the formation of Ganache and MetaMask accounts, voters receive a public key, private key, and MetaMask username. Importantly, the voter's Ganache account credentials (public and private keys) are imported and linked with the MetaMask account, serving as the foundation for ballot casting and future elections.

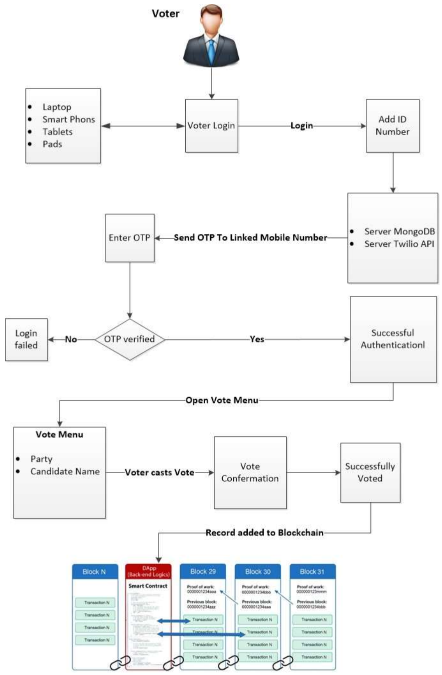


Figure. 4: Secure e-voting Session.

To cast a vote using DApp, the voter logs into the CEC website and selects "Vote" from the DApp menu. Following the confirmation and verification of voter credentials, the voter submits their private key to cast a vote. Concurrently, MetaMask credentials are provided by either the voter or CEC authority to log transactions and blocks on Ganache. Subsequently, this block is mined and seamlessly integrated into Ethereum's immutable main chain, marking the successful execution of transactions and block mining on the main chain.

Voter credentials undergo validation, ensuring the casting of an immutable vote in accordance with smart contract protocols. This meticulous process is designed to safeguard the privacy and security of both the ballot and the cast vote.

During the voting phase, the involvement of a third party is entirely unnecessary, distinguishing this process from traditional client-server systems. Moreover, voters have the flexibility to submit their votes by logging into the dedicated voters' link on the CEC website, providing an additional layer of accessibility to the voting process.

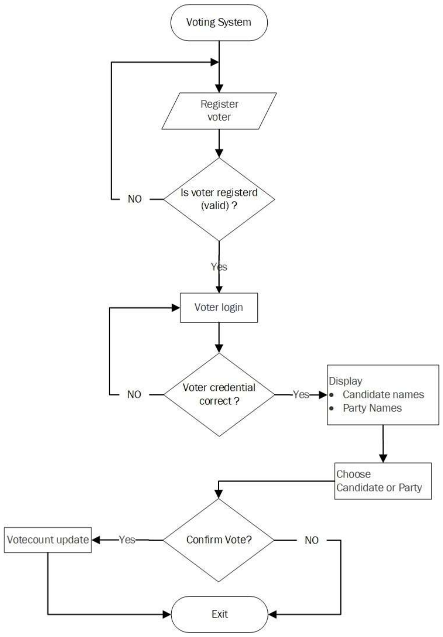


Figure. 5: Secure e-voting Flowchart

5. EXPERIMENTAL RESULTS AND DISCUSSION

This section shows how to setup and run the proposed VoteChain system.

5.1. Configuring Linux for VoteChain system use

4.1.1. Installing the necessary software

Table. 1 summarizes the essential environment and software requirements for running the VoteChain system.

Table 1. VoteChain: Environment and software

Components	Description
Ubuntu	Enterprise Open Source and Linux
NodeJs	JavaScript runtime for server-side development
Truffle	Development environment, testing framework, and asset pipeline for Ethereum
Metamask	Ethereum wallet and browser extension for managing your blockchain identity

5.1.2 Installing the dependencies

To install project dependencies, the following command has to be executed in the project folder using a terminal window:

```
# npm install
```

This command installs the necessary dependencies.

5.1.3. Blockchain and Metamask setup

1- Truffle compile: Execute the following command in the main folder to compile the smart contract:

```
# truffle compile
```

The truffle compile command will generate a.json file in the build directory of the project that corresponds to the smart contract and describes its specific details. An error message will be displayed if there is a syntax error. As a result, it compiles a smart contract and produces JSON files. A smart contract has to be compiled every time a change is made.

2- Deployment Contract: The built-in private chain of the truffle app allows us to test a contract on the local computer. It can only be used for testing, and it cannot interact with other blockchains.

The command below will start the Ganache truffle console. It will show various information, such as its chain network, accounts, and Mnemonic. and can only use these private keys to access these ac- counts, Ganache provides 10 accounts by default.

```
# truffle develop
```

This generates a JSON file in the build directory, corresponding to the smart contract.

3. Installing and Using Metamask on Google Chrome.

4. Configuring env variables and creating a .env file in the main folder.

5. Copying and pasting the 0th account, 0th private key, and mnemonic from the terminal into the .env file. The mnemonic should be framed in quotation marks.

5.1.4. Configuring Data Base (MongoDB)

- Creating a MongoDB database and setting its URL in the DB URI in .env file.
- Changing the ADMIN EMAIL setting to the preferred email address in the.env file.. • Modifying the .env file by entering a password into the ADMIN PASSWORD field.
- Locating the bcrypt hash of the password you selected using the BCBYPT ONLINE tool (keep rounds as 10)
- Building an Administrators collection inside of the MongoDB database.
- Creating a new document with the two properties, and droping it into the admins collection manually.
- Building a second (stats) collection within the MongoDB database.
- Adding a document (manually) to the stats collection, then adding the two properties to the doc and changing the data type from String to Int32.
- Copying the document id from the above doc and pasting it into STATS DOC ID in .env.

5.1.5. API (Twilio)

Creating an account with Twilio Application Programming Interface (API). Finding the three variables that Twilio uses, then adding the three variables that the environment uses.

5.1.6. Finalizing setup tasks

Creating a .env file in the client’s directory.

5.2. Starting and running the project

The frontend will be launched in a web browser and prompting to signing in to Metamask. In Metamask, selecting Localhost 7545 as the network.

5.3. Test case: Voting process

The graphical representation in Fig.10 outlines the intricacies of the voting process, delineating tasks managed by both the administrator and the voters. The pivotal stages involve meticulous data handling and procedural responsibilities. The Central Elections Commission (CEC) meticulously compiles essential information of eligible voters, encompassing details such as name, identity number, mobile phone number, and electoral center number. This comprehensive dataset is then diligently uploaded into the MongoDB database, serving as the foundational repository for the upcoming electoral process. Moreover, pre-Election Preparations occur in anticipation of the imminent voter registration stage, signifying the commencement of the holistic electoral process. By ensuring the accuracy and completeness of voter details in the database, the CEC lays the groundwork for a seamless and credible election. On the other hand, the administrator oversees critical procedures, ensuring the integrity of the entire process. Verification processes are conducted to confirm the authenticity and accuracy of voter information. Database Preparation:

5.3.1.Voter Registration:

Voters actively engage in the registration process, providing essential details to be verified against the prepared database. The CEC verifies voter details against the database, maintaining the security and reliability of the registration process.

This strategic division of responsibilities ensures that the foundation for a fair and transparent election is established. The CEC's proactive measures and the collaborative efforts of administrators and voters contribute to the overall success of the electoral process.

Step 1: The main screen of the VoteChain system is shown in figure 6.

After the voter clicking on Ballot box button, VoteChain system Obtaining voter identity, an Ethereum wallet address, and requesting an OTP verification.

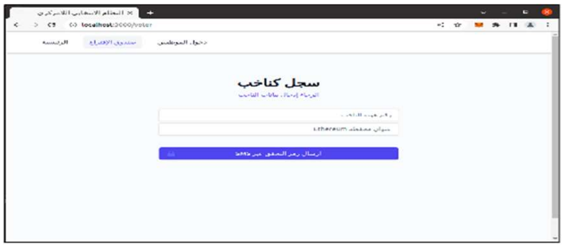


Figure 6. Voter registration

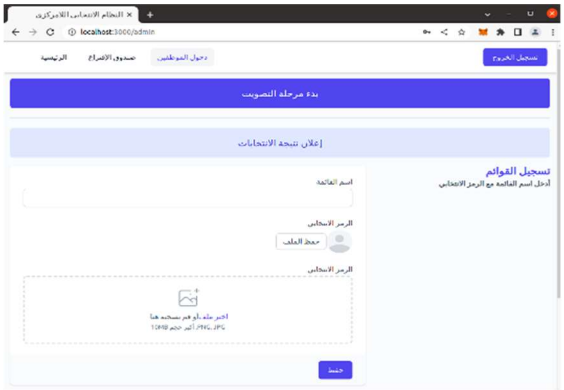


Figure 7. Fill party details

Step 2: The message ”Voter already registered” will be displayed if the per- son has previously registered to vote. If the voter is not registered, an OTP will be sent to the registered phone number, validate the OTP, and then set the voter’s registration status to ”registered” in the database.

After that, addVoter function will be triggered from Election.sol smart contract and the result saved on chain.

5.3.2. Creating Party (Administrator)

Step 1: Admin logs in. The admin’s username and password will be checked to make sure they are typed correctly and that they have permission to access the admin screen, (The username and password saved off chain).

The login validation process employs robust security measures to ensure the integrity of the authentication system. Here's an enhanced description of the steps involved:

1. User Input Retrieval: When an administrator attempts to log in, the system retrieves the email address and password entered into the login form.
2. Password Encryption: The system utilizes the bcrypt node module, a secure hashing algorithm, to encrypt the plain text password provided by the administrator. This encryption process enhances security by transforming the password into a hashed representation.
3. Retrieval of Stored Password: The system retrieves the encrypted password stored in the MongoDB database associated with the provided email address.
4. Comparison of Encryptions: A critical step involves comparing the encrypted password generated from the entered plain text with the stored encrypted password in the database.
5. Validation Outcome: If the comparison results in a match, indicating that the entered password corresponds to the stored encrypted password, the system grants access to the administrator. In the event of a mismatch, a notification is displayed, signaling that the login information is incorrect. The administrator is then informed that the login attempt was unsuccessful. This rigorous process ensures that the system's login mechanism is fortified against unauthorized access attempts. The bcrypt encryption, coupled with careful password matching, forms a robust defense against potential security threats, providing a secure environment for administrator authentication.

Step 2: Fill party details (Party name,Party Logo), refer to figure 7.
Step 3: Click "Create Party," triggering the createPoliticalParty function from the smart contract. The function accepts two parameters representing the political party name and the link for the logo, which is uploaded during the political party addition process. The function checks if the current

voting phase equals 1; if so, it throws an error indicating the registration phase is already over. If the registration phase is still ongoing, the function creates a new PoliticalParty instance and pushes it to the collection that holds the created political parties. After that, the function triggers the smart contract function called PoliticalPartyCreated using the emit keyword to add the political party details to the blockchain after the transaction is signed off using the Metamask wallet.
Step 4: The admin is prompted to sign the transaction in the Metamask wallet in order to save the party information.

5.3.3. Start Voting

Figure. 8 shows how the voting process works.

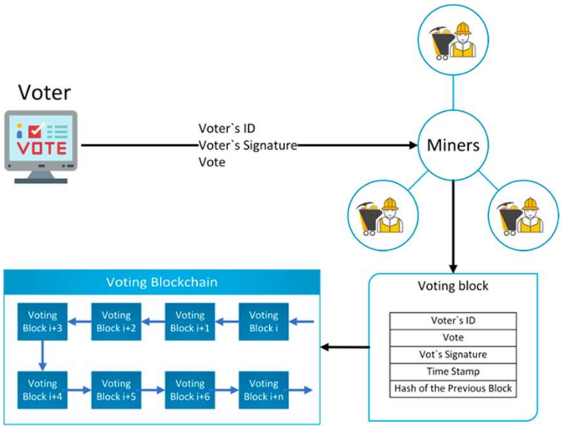


Figure 8. Voting procedures

5.3.3.1. Procedures for Administrators

Step 1: Admin logs in: Verify the admin's username and password to verify sure they are entered properly and that they have authorization to access the admin screen (The username and password saved off chain).

Step 2: The Voting Process Is Started After the Admin Clicks the "Start Voting" Button in the Admin Panel see Figure 9.

Step 3: Voting period is activated when the admin clicks the button (Start Voting). Clicking the button will trigger the smart contract function called startVoting. Once the function is triggered, the voting period will be set to last for 5 minutes and the current phase will transition into the voting phase (Phase 2).

5.3.3.2. Procedures for Voting Process for Voters

Step 1- Voter Authentication:

- The voter begins by entering their Ethereum wallet ID into the VoteChain system.
- The system then retrieves the registered parties and presents them to the voter for selection.

Step 2- Party Selection:
Step3- Blockchain Interaction:

- Upon pressing the "Vote" button, the smart contract's vote function is activated, initiating the recording of the vote on the blockchain.
- The system recognizes the voter based on their Ethereum wallet ID, ensuring the legitimacy of the voting process. The system checks whether the voter is registered for the current election cycle. It confirms that the voter has not cast their vote previously in the ongoing election.

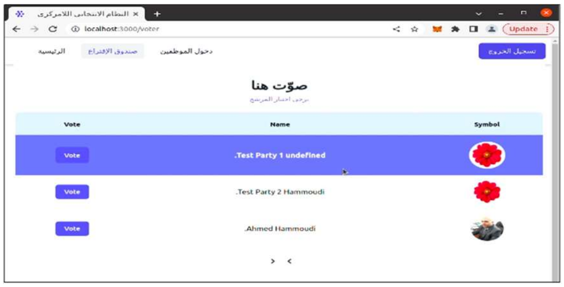
Step4 - Voter Eligibility:

- The system further assesses the voter's eligibility based on the designated election center and the geographical area where the elections are taking place.
- For instance, a voter from City A participating in City B's elections would be restricted, allowing only residents of City B to cast their votes.

Step5-Vote Recording:

- Upon successful completion of the eligibility checks, the system increments the voter count for subsequent result calculations.
- The vote function is then triggered, utilizing the emit keyword in the smart contract, initiating the addition of the vote to the selected party.
- The entire process is securely recorded on the blockchain, ensuring transparency and accuracy.
- The transaction is finalized and signed off using the Metamask wallet.

This meticulous process guarantees the integrity of each vote, preventing multiple votes from a single user and ensuring that only eligible voters participate based on their geographical and registration details.



صوّت هنا

يرجى اختيار المرشح

Vote	Name	Symbol
Vote	.Test Party 1 undefined	
Vote	.Test Party 2 Hammoudi	
Vote	.Ahmed Hammoudi	

FIGURE 9. Start Voting

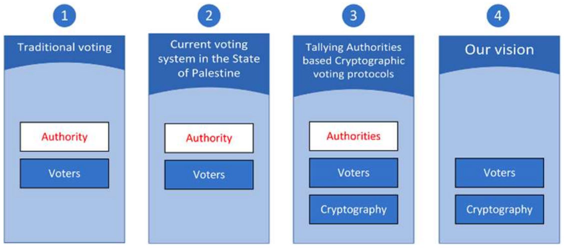


FIGURE 10. A vision of suppressing tallying authorities

5.3.3.3. Declaring Results (Tallying Results)

In the context of voting systems, the involvement of a reliable authority is critical for ensuring the accuracy of election results. Traditional paper-based voting methods present inherent challenges, as voters relinquish control of their ballots once placed in the ballot box. To safeguard the integrity of paper ballots throughout the stages of collection, transit, and counting, a meticulous chain of custody is deemed necessary. Despite such precautions, instances of lost or inaccurately counted paper ballots have been documented in the past.

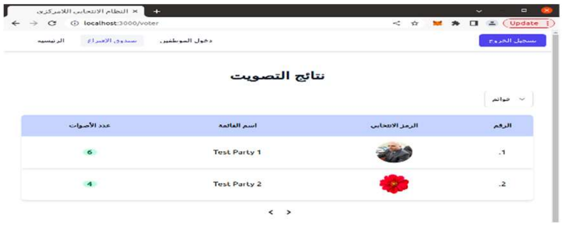
In contrast, the advent of e-voting introduces a paradigm shift towards a "self-enforcing" system. Unlike traditional paper-based procedures, where reliance on tallying authorities is imperative, e-voting systems operate in a manner that is inherently authorities-free. The integration of digital technologies empowers these "self-enforcing e-voting" systems, rendering them not only feasible but also essential for the future of elections.

Figure 10 visually underscores this transformative perspective, emphasizing the pivotal role of self-enforcing e-voting technologies in shaping the integrity and reliability of electoral processes in the years to come.

The voting results computation process could be summarized as:

- Administrator Login: In the initial step, the administrator logs into the system.
- Compute Results: Following the login, the administrator initiates the computation of voting results by clicking the designated "Compute Results" button. The system evaluates and presents the results to the administrator if the allocated voting time of 5 minutes has elapsed. However, if the voting session is still ongoing, a notification is displayed, informing the administrator that voting remains open.

- Result Computation Process: The computational process involves the execution of the computeResult function from the Election.sol smart contract. The function assesses whether the voting phase has concluded; if not, it generates an error message indicating the ongoing voting phase. If the voting phase has ended, the function compiles a list of election centers into an array. Subsequently, it iterates through each election center individually. For each center, the system employs the election center code (associated with the postal code of its physical location) to access the collection storing the vote counts of candidates registered in that specific center. The system identifies the candidate with the highest number of votes within each election center, thereby declaring them the winner. If the victorious candidate is affiliated with a political party, the blockchain increments the seat count for that party by one. Following this determination, the function advances the system to phase 3, signifying the conclusion of the voting process, and officially declares the results (see Figure 11)



نتائج التصويت

الرقم	الرمز الانتخابي	اسم القائمة	عدد الأصوات
.1		Test Party 1	6
.2		Test Party 2	4

Figure 11. Declare Results

6. CONCLUSIONS

In this paper, we introduce a blockchain-based e-voting system leveraging smart contracts. This innovative system facilitates secure and cost-effective elections while safeguarding voter privacy. Our demonstration on an Ethereum private blockchain highlights how blockchain technology provides a unique opportunity to overcome the challenges associated with traditional e-voting systems. This not only guarantees the security and integrity of elections but also enhances transparency. The inherent transparency of blockchain technology simplifies election audits and analysis, making these processes more accessible. The outlined characteristics of a voting system stem from decentralized networks, offering the potential to enhance democratic procedures, especially in direct election systems. Utilizing blockchain as the foundational technology for e-voting opens avenues for increased transparency and independent auditing. This thesis explores the possibilities presented by blockchain technology and its role in shaping the future of e-voting systems. We emphasize ensuring the immutability of the blockchain, allowing any member of the public to independently verify its integrity.

ACKNOWLEDGMENT

We would like to thank Palestine Technical University- Kadoorie for their support.

REFERENCES

[1] P. E. C.- Central-Elections, "The Electoral System for Local Elections." [Online]. Available: https://www.elections.ps/tabid/318/language/en-US/Default.aspx

[2] S. Collard and E. Fabre, "Electronic voting in the French legislative elections of 2012," *Des. Dev. Use Secure Electron. Voting Syst.*, pp. 176–198, Mar. 2014, doi: 10.4018/978-1-4666-5820-2.ch009.

[3] I. L. Awalu, P. H. Kook, and J. S. Lim, "Development of a distributed blockchain evoting system," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jul. 2019, pp. 207–216. doi: 10.1145/3345035.3345080.

[4] N. Guide and A. Gauriansurkar, "Electronic Voting," *Int. J. Res. Eng. Sci. IJRES ISSN*, vol. 10, pp. 21–26, 2022, [Online]. Available: www.ijres.org

[5] Z. Alsaed *et al.*, "Role of Blockchain Technology in Combating COVID-19 Crisis," *Appl. Sci.*, vol. 11, no. 24, p. 12063, Dec. 2021, doi: 10.3390/app112412063.

[6] E.-Y. Daraghmi, M. Abu Helou, and Y.-A. Daraghmi, "A Blockchain-Based Editorial Management System," *Secur. Commun. Netw.*, vol. 2021, pp. 1–17, May 2021, doi: 10.1155/2021/9927640.

[7] E. Y. Daraghmi, Y. A. Daraghmi, and S. M. Yuan, "MedChain: A design of blockchain-based system for medical records access and permissions management," *Appl. Sci. Switz.*, vol. 9, 2019, [Online]. Available: https://doi.org/10.3390/APP9224966

[8] Daraghmi, Daraghmi, and Yuan, "UniChain: A Design of Blockchain-Based System for Electronic Academic Records Access and Permissions Management," *Appl. Sci.*, vol. 9, no. 22, p. 4966, Nov. 2019, doi: 10.3390/app9224966.

[9] W. Zhang *et al.*, "A Privacy-Preserving Voting Protocol on Blockchain," *IEEE Int. Conf. Cloud Comput. CLOUD*, vol. 2018-July, no. April, pp. 401–408, 2018, doi: 10.1109/CLOUD.2018.00057.

[10] E. Daraghmi, S. Jayousi, Y. Daraghmi, R. Daraghmi, and H. Fouchal, "Smart Contracts for Managing the Agricultural Supply Chain: A Practical Case Study," *IEEE Access*, pp. 1–1, 2024, doi: 10.1109/ACCESS.2024.3439412.

[11] S. Agrawal, J. Neu, E. N. Tas, and D. Zindros, "Proofs of Proof-of-Stake with Sublinear Complexity," Sep. 2022, [Online]. Available: http://arxiv.org/abs/2209.08673

[12] W. J. Lai, Y. C. Hsieh, C. W. Hsueh, and J. L. Wu, "DATE: A Decentralized, Anonymous, and Transparent E-voting System," *Proc. 2018 1st IEEE Int. Conf. Hot Inf.-Centric Netw. HotICN 2018*, no. HotICN, pp. 24–29, 2019, doi: 10.1109/HOTICN.2018.8605994.

[13] H. Patil, P. Ladkat, A. Jituri, R. Desai, and Dr. S. Shinde, "Blockchain Based E-Voting System," *SSRN Electron. J.*, 2019, doi: 10.2139/ssrn.3422954.

[14] B. Shahzad and J. Crowcroft, "Trustworthy Electronic Voting Using Adjusted Blockchain Technology," *IEEE Access*, vol. 7, pp. 24477–24488, 2019, doi: 10.1109/ACCESS.2019.2895670.

[15] X. Ma, J. Zhou, X. Yang, and G. Liu, "A blockchain voting system based on the feedback mechanism and wilson score," *Inf. Switz.*, vol. 11, no. 12, pp. 1–13, 2020, doi: 10.3390/info11120552.

[16] G. C. Prasetyadi, A. B. Mutiara, and R. Refianti, "Blockchain-based electronic voting system with special ballot and block structures that complies with indonesian principle of voting," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 1, pp. 164–170, 2020, doi: 10.14569/ijacsa.2020.0110121.

[17] R. Taş and Ö. Ö. Tanriöver, "A Manipulation Prevention Model for Blockchain-Based E-Voting Systems," *Secur. Commun. Netw.*, vol. 2021, 2021, doi: 10.1155/2021/6673691.

[18] K. M. Khan, J. Arshad, and M. M. Khan, "Secure digital voting system based on blockchain technology," *Int. J. Electron. Gov. Res.*, vol. 14, no. 1, pp. 53–62, Jan. 2018, doi: 10.4018/IJEGR.2018010103.

[19] E. Yavuz, A. K. Koç, U. C. Çabuk, and G. Dalkılıç, "Towards secure e-voting using ethereum blockchain," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, 2018, pp. 1–7. doi: 10.1109/ISDFS.2018.8355340.

[20] Y. Salem and E. Daraghmi, "GDPR-BLOCKCHAIN COMPLIANCE FOR PERSONAL DATA," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 24, pp. 5867–5877, 2021.

[21] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, "UniChain: A Design of Blockchain-Based System for Electronic Academic Records Access and Permissions Management," *Appl. Sci.*, vol. 9, no. 22, 2019, doi: 10.3390/app9224966.

[22] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, "MedChain: A Design of Blockchain-Based System for Medical Records Access and Permissions Management," *IEEE Access*, vol. 7, pp. 164595–164613, 2019, doi: 10.1109/ACCESS.2019.2952942.

[23] E. Y. Daraghmi and Y. S. Ming, "Using graph theory to re-verify the small world theory in an online social network word," in *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*, Bali Indonesia: ACM, Dec. 2012, pp. 407–410. doi: 10.1145/2428736.2428811.

[24] Z. Alsaed *et al.*, "Role of Blockchain Technology in Combating COVID-19 Crisis," *Appl. Sci.*, vol. 11, no. 24, 2021, doi: 10.3390/app112412063.

[25] E.-Y. Daraghmi, M.-C. Wu, and S.-M. Yuan, "A Multilayer Data Processing and Aggregating Fog-Based Framework for Latency-Sensitive IoT Services," *Appl. Sci.*, vol. 11, no. 4, p. 1374, Feb. 2021, doi: 10.3390/app11041374.

[26] E.-Y. Daraghmi, M. Abu Helou, and Y.-A. Daraghmi, "A Blockchain-Based Editorial Management System," *Secur. Commun. Netw.*, vol. 2021, pp. 1–17, May 2021, doi: 10.1155/2021/9927640.

[27] E. Daraghmi, C.-P. Zhang, and S.-M. Yuan, "Enhancing Saga Pattern for Distributed Transactions within a Microservices Architecture," *Appl. Sci.*, vol. 12, no. 12, p. 6242, Jun. 2022, doi: 10.3390/app12126242.

[28] E. Daraghmi, Z. Qaroush, M. Hamdi, and O. Cheikhrouhou, "Forensic Operations for Recognizing SQLite Content (FORC): An Automated Forensic Tool for Efficient SQLite Evidence Extraction on Android Devices," *Appl. Sci.*, vol. 13, no. 19, p. 10736, Sep. 2023, doi: 10.3390/app131910736.

[29] L. Rura, B. Issac, and M. K. Haldar, "Implementation and evaluation of steganography based online voting system," *Int. J. Electron. Gov. Res.*, vol. 12, no. 3, pp. 71–93, Jul. 2016, doi: 10.4018/IJEGR.2016070105.

[30] Node.js, "What is npm? | Node.js." 2011. [Online]. Available: https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/

[31] Trufflesuite, "Truffle Suite - Truffle Suite." [Online]. Available: https://trufflesuite.com/

[32] T. Suite, "Truffle Suite - Truffle Suite." [Online]. Available: https://trufflesuite.com/ganache/

[33] Metamask.io, "MetaMask - chrome web store." [Online]. Available: https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn?hl=en

[34] Microsoft, "Visual Studio Code - Code Editing. Redefined." [Online]. Available: https://code.visualstudio.com/?wt.mc_id=DX_841432

[35] Mongodb, "Database. Deploy a multi-cloud database." [Online]. Available: https://www.mongodb.com/

[36] Twilio, "Twilio Customer Engagement Platform." [Online]. Available: https://www.twilio.com/

[37] E. Bandara *et al.*, "Casper: a blockchain-based system for efficient and secure customer credential verification," *J. Bank. Financ. Technol.*, vol. 6, no. 1, pp. 43–62, Jun. 2022, doi: 10.1007/s42786-021-00036-3.

[38] P. Zhang, D. C. Schmidt, J. White, and G. Lenz, "Blockchain Technology Use Cases in Healthcare," *Adv. Comput.*, vol. 111, pp. 1–41, Jan. 2018, doi: 10.1016/bs.adcom.2018.03.006.

[39] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," *Natl. Inst. Stand. Technol.*, Oct. 2018, [Online]. Available: http://dx.doi.org/10.6028/nist.ir.8202

[40] T. M. Buchsbaum, "E-Voting: International Developments and Lessons Learnt," 2004, [Online]. Available: https://subs.emis.de/LNI/Proceedings/Proceedings47/Proceeding.GI.47-4.pdf

[41] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A Survey on Ethereum Systems Security: Vulnerabilities, Attacks, and Defenses," *ACM Comput. Surv.*, vol. 53, no. 3, Jun. 2020, doi: 10.1145/3391195.

[42] A. Endurthi and A. Khare, "Two-Tiered Consensus Mechanism Based on Proof of Work and Proof of Stake," *2022 9th Int. Conf. Comput. Sustain. Glob. Dev. INDIACom*, pp. 349–353, 2022, doi: 10.23919/INDIACom54597.2022.9763215.

[43] M. Hajian Berenjestanaki, H. R. Barzegar, N. El Ioini, and C. Pahl, "Blockchain-Based E-Voting Systems: A Technology Review," *Electronics*, vol. 13, no. 1, p. 17, Dec. 2023, doi: 10.3390/electronics13010017.

[44] J. Liu, T. Han, M. Tan, B. Tang, W. Hu, and Y. Yu, "A Publicly Verifiable E-Voting System Based on Biometrics," *Cryptography*, vol. 7, no. 4, p. 62, Nov. 2023, doi: 10.3390/cryptography7040062.

[45] A. J. Roberto, L. D. Mattson, P. A. Von Feldt, and X. Zhou, "'The Only Thing We Have to Fear Is Fear Itself': Predicting College Students' Voting Behavior Using the Extended Parallel Process Model," *Soc. Sci.*, vol. 12, no. 11, p. 628, Nov. 2023, doi: 10.3390/socsci12110628.