



**VIT<sup>®</sup>**

**Vellore Institute of Technology**

(Deemed to be University under section 3 of UGC Act, 1956)

**CSI3003**

**Blockchain Technologies**

**Project Title:**

**Implementing Immutable Ledgers In Decentralized Voting System**

**Submitted To:**

**Prof. Selvi M**

**Team Members:**

**Sadiya Rasool (20MID0190)**

## INDEX:

1. [ABSTRACT](#)
2. [INTRODUCTION](#)
3. [LITERATURE SURVEY](#)
4. [PROPOSED WORK](#)
5. [ALGORITHM](#)
6. [CODE](#)
7. [RESULTS](#)
8. [CONCLUSION](#)
9. [REFERENCES](#)

## ABSTRACT

This paper addresses the critical issues of mistrust and lack of transparency in traditional and digital voting systems. It proposes a solution that leverages blockchain technology to create a secure, transparent, and reliable voting system. By highlighting the flaws in existing voting systems and emphasizing the need for a solution that ensures **fairness, integrity, and trust** in the voting process.

The proposed framework presents a high-level architecture of the voting system, involving key stakeholders such as voters, Voting Management System (VMS), Identifying Authorities (IA), and Authentication Authorities (AA). Voters can participate in the voting process through a **decentralized application** (dAPP) accessible via smartphones or web browsers. The system verifies the eligibility of voters through their computerized National ID and allows them to cast their votes securely.

The **use of blockchain** technology in the proposed framework ensures **transparency, security, and traceability** in the voting process. Each vote is recorded on the blockchain, making it **immutable** and **tamper-proof**. Smart contracts are employed to securely execute transactions and establish a connection between users and the network.

The paper also discusses the security aspects of the blockchain-based voting system, including encryption techniques, prevention of 51% attacks, and the use of cryptographic hash functions. It highlights the scalability and performance of the proposed system, demonstrating its feasibility for large-scale implementation.

## INTRODUCTION

The traditional voting system has faced widespread mistrust and challenges due to a lack of transparency and security. To address these issues, this paper proposes a framework that leverages blockchain technology to create a transparent and reliable voting system. The framework aims to build trust between voters and election authorities by providing maximum transparency and integrity in the voting process.

The flaws in traditional and digital voting systems emphasize the need for a solution that ensures fairness and reduces the risk of manipulation.

It introduces blockchain technology as a potential solution, highlighting its ability to provide transparency, security, and traceability in transactions. The proposed framework presents a **high-level architecture of the voting system**, involving key stakeholders such as voters, Voting Management System (VMS), Identifying Authorities (IA), and Authentication Authorities (AA).

The framework employs a combination of features of blockchain technology to guarantee the integrity of the electoral process. This includes the verification of **eligibility** through the use of a **computerized National ID**, as well as the secure recording of voting transactions on the blockchain. To prevent double voting, each voter is allocated a **Voting Coin (VC)**. Additionally, smart contracts are employed to facilitate secure transactions and connect users to the network. This system uses **encryption** and **cryptographic hash functions** to protect the voters' privacy and prevent unauthorized access. The framework focuses on 51% attacks to ensure the system is resilient against malicious actors.

**Performance analysis** of the proposed system is performed by simulating real-life scenarios with the help of **Remix**, a browser based blockchain tool. Real-life scenarios are used to measure the system's response time as well as its scalability. The evaluation shows that the system is able to efficiently process large numbers of transactions, which makes it **suitable for large-scale implementation**.

The proposed framework offers improved latency management, simplified voter verification, and enhanced transparency compared to existing solutions by addressing the challenges of complex computation and vulnerability of voter identities. Hence, It offers a solution to the flaws and lack of trust in traditional and digital voting systems.

## LITERATURE SURVEY

1. Alvi, Syada Tasmia, et al. "DVTChain: A Blockchain-Based Decentralized Mechanism to Ensure the Security of Digital Voting System Voting System." Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, 1 July 2022.

### **DVTChain: A Blockchain-Based Decentralized Mechanism for Digital Voting Security**

Traditional paper balloting methods have limitations, such as errors and potential abuse. Digital voting systems face issues related to privacy, security, and verifiability. This literature review explores the research paper by Syada Tasmia Alvi, Mohammed Nasir Uddin, L. Islam et al.

#### **Overview of the Research Paper:**

The research paper focuses on the development of DVTChain, a blockchain-based digital voting system that ensures anonymity, privacy, verifiability, integrity, security, and fairness in the voting process. The proposed system utilizes smart contracts on the Ethereum blockchain to perform various operations, including voter verification, voting correctness, and protection against fraudulent activities.

#### **Key Contributions and Mechanism of DVTChain:**

1. **Registration Phase:** The first phase of the voting system involves two parts - voter registration and candidate registration. The voter's information is stored as a hash in the blockchain, ensuring anonymity and privacy.
2. **Voting Setup Phase:** This phase sets up the voting process by creating smart contracts for voters and candidates. The voter authentication and vote casting process are performed directly between the voter and the blockchain.
3. **Voting Phase:** During this phase, voters authenticate themselves and cast their votes using a vote coin. The casted votes are encrypted using a public key and stored in the blockchain until the end of the election.
4. **Result Phase:** After the voting period ends, the encrypted votes are decrypted by the election commission using a private key. The voting contract then sends vote coins to the chosen candidate's public key for counting, ensuring the integrity of the voting process.

#### **Comparison with Existing Systems:**

DVTChain is compared with other existing blockchain-based e-voting systems. These systems have limitations such as weak voter authentication, lack of privacy, and vulnerability to attacks.

The proposed DVTChain system has been implemented and evaluated using the Ethereum blockchain platform. Performance of the system has been analyzed based on security and gas costs. The research paper discusses the implementation details, including the use of smart contracts, encryption techniques, and the publishing of voting results.

2. Yu, B. et al. (2018). Platform-Independent Secure Blockchain-Based Voting System. In: Chen, L., Manulis, M., Schneider, S. (eds) Information Security. ISC 2018. Lecture Notes in Computer Science(), vol 11060. Springer, Cham.

### **A Platform-Independent Secure Blockchain-Based Voting System**

(Proposed in 2018 by B yu et al.)

#### **Overview of the Research Paper:**

The proposed eVoting system in this paper offers key features-

**Decentralization:** It eliminates the need for a central authority in vote tallying and result publishing, using blockchain for security and transparency.

**Platform Independence and Security:** It's platform-agnostic and employs cryptographic methods (like Paillier encryption, proof-of-knowledge, and linkable ring signatures) for robust security.

**Scalability:** The system handles large-scale voting efficiently with optimized ring signature algorithms, maintaining security even with one million voters.

The system involves four key entities:

**Smart Contract Administrator:** Responsible for deploying and terminating the smart contract on the blockchain platform. This administrator's role is crucial for setting up the voting process.

**Voting Administrator:** Organizes the vote by configuring voting parameters and initiating the tallying and result publishing phase. The system uses SLRS to prevent administrators from linking ballots to specific voters, enhancing privacy.

**Smart Contract:** Performs essential functions, including storing encrypted ballots, verifying their validity, counting encrypted votes, verifying voting results' correctness, and publishing the results. It also provides a platform for voters to verify the voting process.

**Voters:** Individuals with the right to cast their votes. They must register in the voting system before participating.

The correctness of the system is ensured through public verifiability provided by the smart contract and cryptographic schemes' proof of knowledge. Any inconsistencies in transaction execution trigger an error and lead to transaction rejection.

3. Yi, Haibo. "Securing E-Voting Based on Blockchain in P2P Network." EURASIP Journal on Wireless Communications and Networking, vol. 2019, no. 1, 28 May 2019

### **Title: Securing e-voting based on blockchain in P2P network**

Authors: Haibo Yi

Journal: EURASIP Journal on Wireless Communications and Networking (2019)

#### **Introduction:**

The research paper discusses the design and implementation of a blockchain-based e-voting system for secure and practical electronic voting. It highlights the need for a more secure and practical e-voting system and presents techniques to exploit blockchain technology to improve

the security and anonymity of the voting process. A synchronized model of voting records is introduced based on distributed ledger technology (DLT) to prevent forgery of votes. It also proposes a user credential model based on elliptic curve cryptography (ECC) for authentication and non-repudiation. Additionally, a withdrawal model is presented that allows voters to change their vote before a preset deadline.

### **Overview of the Research Paper:**

#### Use of Blockchain Technology:

The paper explains the concept of a blockchain as a growing list of blocks linked through cryptographic hashes and discusses the synchronization of ledgers among multiple nodes and the use of community validation.

#### Proposed Blockchain-based E-voting Scheme:

A synchronized model of voting records based on DLT would prevent vote forgery.  
A user credential model based on ECC would facilitate authentication and non-repudiation.  
A withdrawal model that allows voters to change their vote before a preset deadline is made.  
The voting process and the generation of new blocks in the blockchain can be illustrated.

#### Working and Implementation:

1. **Synchronized Model of Voting Records:** The model utilizes distributed ledger technology (DLT) to create a synchronized model of voting records. This prevents forgery of votes by ensuring that all votes are recorded and verified across multiple nodes in the blockchain network.
2. **User Credential Model:** The model incorporates a user credential model based on elliptic curve cryptography (ECC) to provide authentication and non-repudiation. Each voter is issued a unique identity (ID) and a list of candidates..
3. **Withdrawal Model:** The model allows voters to change their vote before a preset deadline. Similar to the voting process, the withdrawal process is facilitated through the blockchain network.
4. **Voting Process:** Voters cast their votes by submitting their vote, timestamp, and signature to the network. Miners, elected randomly, verify the votes and generate new blocks in the blockchain. All votes in the blockchain are cryptographically linked block by block.
5. **Mining and Generation of Voting Blocks:** New blocks are generated by users in the peer-to-peer (P2P) network based on a proof-of-work (PoW) algorithm. When a new vote is submitted and verified, a miner generates a new block with the vote information for the network.

4. Denis González, Camilo, et al. "Electronic Voting System Using an Enterprise Blockchain." Applied Sciences, vol. 12, no. 2, 1 Jan. 2022, p. 531

#### **Title: Electronic Voting System Using an Enterprise Blockchain**

Authors: Camilo Gonzalez, Daniel Frias, Alexi Masso, Omar Rojas, Guillermo Sosa-Gomez

#### **Introduction:**

Electronic voting systems have gained prominence as a crucial component of modern democracies. They serve as cryptographic, decentralized registries distributed among various entities, including corporations, institutions, and partnerships. Each block contains essential information such as a unique alphanumeric code (HASH), digitally signed through public-key cryptography, and a chronological arrangement of transactions.

### **Overview of the Research Paper:**

Blockchain technology is characterized by decentralization, consensus algorithms, cryptographic security, and immutability. It relies on a network of distributed peer nodes for communication and consensus. Each node maintains an identical copy of the data, ensuring transparency and accountability. Transactions are securely stored in blocks using cryptographic techniques, which enables easy tracking of any transaction.

### **Hyperledger Fabric 2.2 (HF):**

It is an open-source enterprise-grade distributed ledger technology employed in this study for developing an electronic voting system. It operates on authorized blockchain networks, emphasizing consensus protocols, Certification Authorities (CA), peer-to-peer communication, and a distributed database ensuring data integrity. HF supports smart contracts written in general-purpose languages, allowing for varying levels of privacy and visibility among network members.

**Assets:** Represented as key-value pairs, assets in Hyperledger Fabric can range from tangible (real estate, hardware) to intangible (intellectual property), and undergo state changes recorded as transactions in the ledger.

**Chaincode or Smart Contract:** These are executable codes written in a language unaffected by natural language ambiguities. They automate the verification of compliance with agreements and are immutable parts of the network.

**Identities:** In an authorized network like Hyperledger Fabric, participants must authenticate their identity through a verifiable Public Key Infrastructure (PKI), facilitated by a Certification Authority (CA).

**Membership Service Provider (MSP):** An MSP verifies participant identities and assigns privileges, transforming identity into role within the network.

**Wallet:** A wallet contains user identities, determining access rights to network resources in combination with an MSP.

### **HF for Electronic Voting:**

Its unique architecture allows for scalability, a crucial factor in the performance of blockchain-based solutions. Recent advancements have demonstrated Hyperledger Fabric's ability to handle up to 20,000 transactions per second, making it a compelling choice for electronic voting systems.

5. Alvi, Syada Tasmia, et al. "Digital Voting: A Blockchain-Based E-Voting System Using Biohash and Smart Contract." IEEE Xplore, 1 Aug. 2020

**Title: A Blockchain-based E-Voting System using Biohash and Smart Contract**

Authors: Syada Tasmia Alvi, Mohammed Nasir Uddin, and Linta Islam

### **Overview of the Research Paper:**

Each voter receives a distinct Voter ID (VID), and the data about each voter is kept in the blockchain as a hash value. Voters can use their VID to validate their vote on the blockchain. By employing public keys as their identity and storing their data as hashes, the system ensures the anonymity and privacy of voters. Their electoral processes. A smart contract-based blockchain-based electronic voting system has been presented as a solution to this problem. The system attempts to give voters security, privacy, anonymity, and integrity. For data integrity and anonymity, Merkle tree and fingerprint hash are used. Scalability, verifiability, decentralisation, singularity, and authentication are some benefits of the suggested system. Comparing the proposed system to others now in use reveals that it performs better in terms of offering different security and privacy characteristics.

**Decentralisation:** The voting method intends to do away with the requirement for a centralised authority and distribute the voting process among numerous nodes by using blockchain technology, which is a decentralised ledger. As there is less chance of vote manipulation or tampering, this decentralisation improves the voting system's security, transparency, and integrity.

**Scalability:** Markle tree and fingerprint hash usage also contributes to maintaining the reliability and effectiveness of the system. In general, the suggested system seeks to offer a scalable answer for electronic voting.

**Biohash:** The proposed system uses Biohash, a biometric-based authentication mechanism, to ensure the integrity and security of the voting process.

**Authentication:** The proposed system ensures the integrity, anonymity, privacy, and security of voters. To access the voting system, voters must send their identity information to smart contracts to verify their eligibility. Ensure that only authorized individuals or processes are allowed to access company IT resources

**Voting Administrator:** The voting procedure is secure, honest, and private thanks to the smart contract. It also contributes to choosing a miner to increase transaction and vote counting speed.

**Merkle tree and fingerprint hash:** The proposed system uses a Merkle tree and fingerprint hash to achieve data integrity and anonymity.



6. S. T. Alvi, L. Islam, T. Y. Rashme and M. N. Uddin, "BSEVOTING: A Conceptual Framework to Develop Electronic Voting System using Sidechain," 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Semarang, Indonesia, 2021, pp. 10-15, doi: 10.23919/EECSI53397.2021.9624282.

**Title: BSEVOTING is a conceptual framework for developing an electronic voting system using sidechain technology**

The proposed BSEVOTING system, which uses sidechain technology, is said to be meant to be an effective and scalable electronic voting system, according to the document's conclusion. By offering qualities like eligibility, anonymity, integrity, privacy, security, fairness, and receipt-freeness, it seeks to alleviate the shortcomings of conventional blockchains. By employing sidechains to isolate the storing of vote data for each candidate, the system uses less storage. The adoption of Ethereum as a blockchain platform and its scalability issues are also mentioned in the document. The suggested system will be implemented and further discussed in future work.

**Decentralisation:**Blockchain enables a peer-to-peer consensus network in which vote data is recorded and validated by numerous network nodes, doing away with the requirement for a central authority. E-voting systems can provide security, transparency, and integrity while lowering the possibility of vote tampering or manipulation by utilising decentralisation.

**Scalability:**Scalability is addressed in the proposed BSEVOTING system by using sidechain technology. By shifting transactions to a parallel chain, sidechains let the primary blockchain run more efficiently and reduce its workload. This strategy ensures effective data processing and storage, improving the voting system's overall scalability.

**Smart Contract:** Smart contracts are used to manage many parts of the voting process in the context of the planned BSEVOTING system. By automating and enforcing the rules and processes laid forth in them, these smart contracts guarantee the correctness and integrity of the voting process.

**Integrity and legitimacy:** The framework aims to ensure the integrity and legitimacy of the entire voting system by using a clear, stable, and tamper-proof voting process.

**Sidechain technology:** which is a separate blockchain network that connects to another blockchain, called a parent blockchain or mainnet, via a two-way peg.

7. Li C, Xiao J, Dai X, Jin H. AMVchain: authority management mechanism on blockchain-based voting systems. Peer Peer Netw Appl. 2021;14(5):2801-2812. doi: 10.1007/s12083-021-01100-x. Epub 2021 Mar 11. PMID: 33723494; PMCID: PMC7947943.

**Title: AMVchain: authority management mechanism on blockchain-based voting systems**

Is published in year 2021 by Chenchen Li , Jiang Xiao , Xiaohai Dai ,Hai Jin.

**Overview of the Research Paper:**

The AMVchain electronic voting system, which overcomes the drawbacks and difficulties of current blockchain-based voting systems, is presented as the document's conclusion. As a consensus algorithm, it uses PBFT and consortium blockchain, and proxy nodes cut down on the number of nodes involved. The performance of the ring signature during the voting process and the system's tally time are highlighted in the document. Results from experiments show that AMVchain can efficiently process voters' requests in situations when voter requests are reasonably numerous.

**Smart Contract Administrator:** The Smart Contract Administrator ensures that the smart contracts execute according to pre-determined rules and are stored securely on the blockchain.

**Voting Administrator:** They may have the authority to set voting system parameters, such as the start and end times of the voting, and deploy the necessary chaincode on the blockchain. Have access to the dashboard that displays the blockchain network and voting information.

**Scalability:**Meta-analysis on scalable blockchain-based electronic voting systems found that blockchain technology has the potential to improve the organization of large-scale elections in countries without access to modern technologies.

**Three-tier access control framework:** The system uses a three-tier access control framework to ensure the privacy of voters and candidates. The framework includes a registration authority, a voting authority, and a counting authority.

**Efficiency:** The proposed system is designed to be efficient and effective in handling the voting process.

**Consortium blockchain:** The proposed system uses a consortium blockchain to ensure the transparency and immutability of the voting process.

8. Khan, Kashif Mehboob, et al. "Investigating Performance Constraints for Blockchain Based Secure E-Voting System." Future Generation Computer Systems, vol. 105, Apr. 2020, pp. 13–26, <https://doi.org/10.1016/j.future.2019.11.005>.

**Title: Investigating Performance Constraints for Blockchain-Based Secure E-Voting System**

By Kashif Mehboob Khan, Junaid Arshad, Muhammad Mubashir Khan

**Overview of the Research Paper:**

The document discusses a literature survey conducted on the performance constraints of blockchain-based e-voting systems. The survey focuses on scalability and performance factors

such as block size, block generation rate, and transaction speed. The research includes experimentation with both permissioned and permissionless blockchain settings to evaluate the impact of these factors on the efficiency and scalability of the e-voting model. The findings aim to provide insights for achieving scalable blockchain-based e-voting solutions and contribute to the research community's understanding of the challenges and potential trade-offs in this domain.

**Performance constraints:** The paper explores the performance constraints of blockchain-based secure e-voting systems, including block generation rate, transaction speed, and scalability.

**Decentralization, transparency, and tamper-proof:** The proposed system is designed to be decentralized, transparent, and tamper-proof, ensuring the integrity and legitimacy of the entire voting system.

**Novel blockchain-based e-voting system:** The paper presents a novel blockchain-based e-voting system that investigates the capabilities of blockchain technology to achieve e-voting for permissioned and permissionless voting models.

**Voters:** Discusses the use of blockchain transactions to transfer voter tokens from valid voter addresses to valid candidate addresses, ensuring verifiability and maintaining an auditable and tamper-proof list of voting records.

**Scalability:** The document discusses various parameters that impact the scalability of blockchain-based e-voting systems, such as block size, block generation rate, transaction speed, and network constraints.

9. Abuidris, Y., Kumar, R., Yang, T. and Onginjo, J. (2021), Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding. ETRI Journal, 43: 357-370. <https://doi.org/10.4218/etrij.2019-0362>

**Title: Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding**

**Overview of the Research Paper:**

The document includes previous work related to e-voting systems based on blockchain. It mentions various studies that have proposed protocols for blockchain-based e-voting systems using different consensus methods such as Proof of Work (PoW) and Proof of Stake (PoS). The limitations of these protocols in terms of scalability and security are also discussed. The document then introduces the proposed hybrid consensus model (PSC-Bchain) combined with sharding to address these limitations and enhance the security and scalability of the e-voting system.

**Scalability:** The system aims to improve scalability by using sharding.

**Decentralization:** The proposed system is designed to be ensuring the integrity and legitimacy of the entire voting system.

**Sharding:** The system uses sharding, which is a technique that divides the blockchain network into smaller, more manageable parts called shards, to improve scalability.

**Hybrid consensus model:** The proposed system uses a hybrid consensus model (PSC-Bchain) composed of Proof of Credibility and Proof of Stake that work mutually to address the security challenges of traditional blockchain-based e-voting systems

**Voters:** The blockchain contract ensures transparency and integrity, allowing voters to track and verify the voting process

**Smart Contract:** Smart contracts in the proposed e-voting system are deployed in the top layer of the blockchain, specifically on the Ethereum blockchain.

10. Pawlak, Michał, et al. "Towards the Intelligent Agents for Blockchain E-Voting System." *Procedia Computer Science*, vol. 141, 2018, pp. 239–246, <https://doi.org/10.1016/j.procs.2018.10.177>.

**Title: Towards the intelligent agents for blockchain e-voting system**

Authors: Michał Pawlak, Aneta Poniszewska-Marańda, and Natalia Kryvinska published in 2018.

**Overview of the Research Paper:**

The survey covers topics such as remote electronic voting, electronic voting schemes, verifiable internet voting, and the use of blockchain technology in voting systems. It also mentions specific examples of countries like Estonia and South Korea that have implemented blockchain technology in their voting procedures. The document introduces the concept of the Auditable Blockchain Voting System (ABVS), which integrates e-voting with blockchain technology to create a secure and verifiable voting system. It discusses the use of intelligent agents and a multi-agent system in ABVS to enhance security and efficiency.

**System concepts:** The proposed system uses intelligent agents and multi-agent system concepts to address the lack of transparency and auditability in traditional e-voting systems

**Blockchain technology:** The proposed system uses blockchain technology to ensure transparency, tamper-proofing, and security

**Decentralization:** This decentralization ensures that the agents cannot be modified outside the nodes, making it easier to detect any attempts to tamper with the voting system.

**Transparency and auditability:** The proposed system aims to ensure transparency and auditability in the e-voting process.

**Security:** The proposed system is designed to be secure, ensuring the integrity and legitimacy of the entire voting system.

**Smart Contract:** smart contracts can be utilized to send intelligent agents in the form of transactions between the voting applications and trusted nodes. This allows for the secure and automated interaction between different components of the voting system, enhancing transparency and efficiency Voters

11. Shujaa, Mohamed & Ulddin, Ahmed & muhsen, Ahmed. (2023). Secure blockchain e-voting system using speck cipher. 10.1063/5.0154775.

**Title:Secure blockchain e-voting system using speck cipher**

proposed system uses Raspberry Pi

**Overview of the Research Paper:**

The "Secure blockchain e-voting system using speck cipher" is a proposed system that aims to provide a secure and efficient e-voting system using blockchain technology and the Speck cipher algorithm . a small and low-cost computer, to implement the blockchain network and the Speck cipher algorithm to ensure the security of the voting process . The system is designed to be lightweight and efficient, making it suitable for large-scale e-voting system

**Speck cipher algorithm:** The proposed system uses the Speck cipher algorithm to ensure the security of the voting process

**Transparency:** The proposed model aims to ensure the security and transparency of the voting process.

**Accuracy :** The proposed model aims to ensure the accuracy and trustworthiness of the election results.

**decentralized:**Highlights the importance of to ensure transparency and trust in the voting process.

**Scalability:**The challenges of organizing large-scale elections in countries without access to modern technologies and how blockchain technology can be a solution to these issues.

**Verifiability:** The proposed model can allow voters to verify their casted vote, ensuring verifiability.

**AES encryption algorithm and SHA-256:** The proposed system can use the AES encryption algorithm and SHA-256 along with blockchain to ensure the security of the voting process.

12. Y. Li et al., "A Blockchain-Based Self-Tallying Voting Protocol in Decentralized IoT," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 1, pp. 119-130, 1 Jan.-Feb. 2022, doi: 10.1109/TDSC.2020.2979856.

**Title: A Blockchain-based Self-tallying Voting Scheme in Decentralized IoT**

**Authors:** Yannan Li, Willy Susilo, Guomin Yang, Yong Yu, Dongxi Liu and Mohsen Guizaniø

**Overview of the Research Paper:**

The presented self-tallying voting system utilizes blockchain technology and consists of three main phases: Pre-vote, Vote, and After-vote.

**Pre-vote Phase:**

Setup: Voters register and obtain private-public key pairs. Each voter generates a random private key and computes the corresponding public key. They also create a zero-knowledge proof to prove the validity of their key pairs, and both the public keys and proofs are published on the blockchain.

Commit: Before casting their votes, voters choose a random value and create a commitment to their vote along with a zero-knowledge proof. The commitment ensures fairness in the system. If a voter quits after making a commitment, their ballot can be revealed with the cooperation of all other voters.

**Vote Phase:**

Vote: Voters encrypt their votes and create zero-knowledge proofs to prove the correctness of their votes. These encrypted votes are then published on the blockchain.

**After-vote Phase:**

Tally: To compute the final voting results, all the encrypted votes are collected from the blockchain, and the tally is performed publicly by anyone.

Recover: In case the last voter does not follow the rules and refuses to vote, other voters can recover the ballot using the corresponding commitment and the help of all the other voters.

**Dealing with Abortive Issues:**

To handle abortive issues where voters may quit the voting process, the system uses commitments and zero-knowledge proofs. If the last voter quits after making a commitment, their ballot can be revealed with the cooperation of all other voters. This ensures that simply removing votes won't lead to a different result.

**Dealing with Adaptive Issues:**

To address adaptive issues where the last voter has an advantage in accessing the results early,

the system employs time-locked encryption. Votes are encrypted with a witness encryption, and the witness can be produced by the blockchain after a certain time period, which acts as a computational reference clock. Decryption of votes is only allowed after the deadline has passed, ensuring that all voters and observers can tally the votes and obtain the result simultaneously.

13. Halderman, J.A., Teague, V. (2015). The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election. In: Haenni, R., Koenig, R., Wikström, D. (eds) E-Voting and Identity. Vote-ID 2015. Lecture Notes in Computer Science(), vol 9269. Springer, Cham.

**Title: The New South Wales iVote System: Security Failures and Verifications Flaws in a Live Online Election.**

Authors: J. A. Halderman, and V. Teague

**Overview of the Research Paper:**

In 2015, during the New South Wales State election, around 280,000 eligible citizens used the iVote system. It was developed by ScytI and to cast a vote, citizens had to go through four steps:

1. **Registration:** Voters had to register and were given an 8-digit iVote ID. They also had to choose a 6-digit PIN.
2. **Voting:** They logged in using their iVote ID and PIN, cast their vote, and received a 12-digit receipt number. The vote was encrypted on the voting device, sent to the voting server, and forwarded to a separate verification service.
3. **Verification (Optional):** They could choose to use an interactive voice response (IVR) system to verify the vote. They entered the iVote ID, PIN, and receipt number to hear their vote read back to me. This service was available until the close of polls.
4. **Receipt Query (Optional):** They also had the option to check if the vote, based on the receipt number, was included in the final count through an online receipt service. No login was required, and this service remained active even after polls closed.

However, there were significant security issues with the system. The server used for iVote had poor SSL configuration, receiving an F rating on security tests. It also used insecure Diffie-Hellman parameters, allowed weak 512-bit export cipher suites that were vulnerable to attacks, and had a vulnerability to the POODLE attack. These security problems could potentially have allowed a malicious attacker to insert vote-stealing code into the iVote application, compromising the integrity of the election.

14. Ayed, Ahmed Ben. "A Conceptual Secure Blockchain Based Electronic Voting System." International Journal of Network Security & Its Applications 9 (2017): 01-09.

## **Title: A Conceptual Secure Blockchain Based Electronic Voting System**

Authors: Ben Ayed, Ahmed

### **Overview of the Research Paper:**

A Conceptual Secure Blockchain-Based Electronic Voting System was proposed by Ahmed Ben Ayed in 2017 to address significant shortcomings in existing electronic voting systems, particularly in terms of security and identity verification. This innovative system incorporates four core requirements to ensure a reliable and trustworthy electronic voting process:

#### **1. Authentication:**

- Only registered voters are eligible to cast their votes.
- Eliminates the need for a registration process by verifying voters' identities against a pre-verified database.
- Ensures that each voter can only cast one vote.

#### **2. Anonymity:**

- Guarantees voter anonymity throughout and after the election.
- Prevents any links between voters' identities and their ballots.

#### **3. Accuracy:**

- Ensures the accuracy of votes.
- Every vote is counted, and the system prevents changes, duplications, or removal of votes.

#### **4. Verifiability:**

- The system is designed to be transparent and auditable.
- Allows for independent verification to ensure all votes are counted correctly.

The system utilizes blockchain technology as a fundamental component:

- Blockchain Foundation:
  - The first transaction added to the blockchain represents the candidate and serves as the foundation block.
  - Subsequent votes for the candidate are recorded on top of this foundation block.
  - Protest votes are also permitted to express dissatisfaction with candidates or the political system.
- Security Measures:
  - To enhance security, each block contains the previous voter's information, making it easier to detect compromises or tampering.
  - The blockchain is decentralized, with nodes in each district, preventing a single point of failure.
  - The blockchain serves as the platform for the actual voting process, where votes are sent to nodes for addition.

Ahmed Ben Ayed's proposed e-Voting system offers a robust solution to electronic voting challenges, offering secure, anonymous, and verifiable voting while eliminating the need for a separate registration process. However, the system's practical implementation and security measures would require rigorous evaluation to ensure its effectiveness and trustworthiness in real-world scenarios.



## 15. Blockchain-Based E-Voting System

### **Title: Blockchain Based E-Voting System**

Authors: Kukwase, Praful, et al.

### **Overview of the Research Paper:**

This paper proposes a blockchain-based system for organizing and conducting elections. The system utilizes a chain of blocks where each block represents a collection of information, and mining is used to collect and organize this information. Each block is identified using a scientific hash, and they are linked together to form a sequence starting with the "Genesis Block." The data is connected using a linked list structure.

The methodology for this system involves:

**Organizing Elections:** Election administrators use a decentralized app (dApp) to set up elections. A smart contract is used to establish a list of eligible candidates and voting districts. Smart contracts for each ballot are deployed on the blockchain, and permission is granted to district nodes to interact with the appropriate smart contract for voting.

**Registration of Voters:** Election administrators define a deterministic list of eligible voters, requiring the creation of a government identity verification service. Each eligible voter is provided with a unique digital ID and a PIN number, and a wallet is created for them for each election they are eligible for. The system ensures anonymity by using NIZKP (Non-Interactive Zero-Knowledge Proof).

**Transaction of a Vote:** Voters interact with smart contracts on the blockchain specific to their voting district. The smart contract communicates with the Ethereum blockchain and records the vote as a transaction if a consensus is reached among the nodes in the district. Voters must provide identification to authenticate their vote.

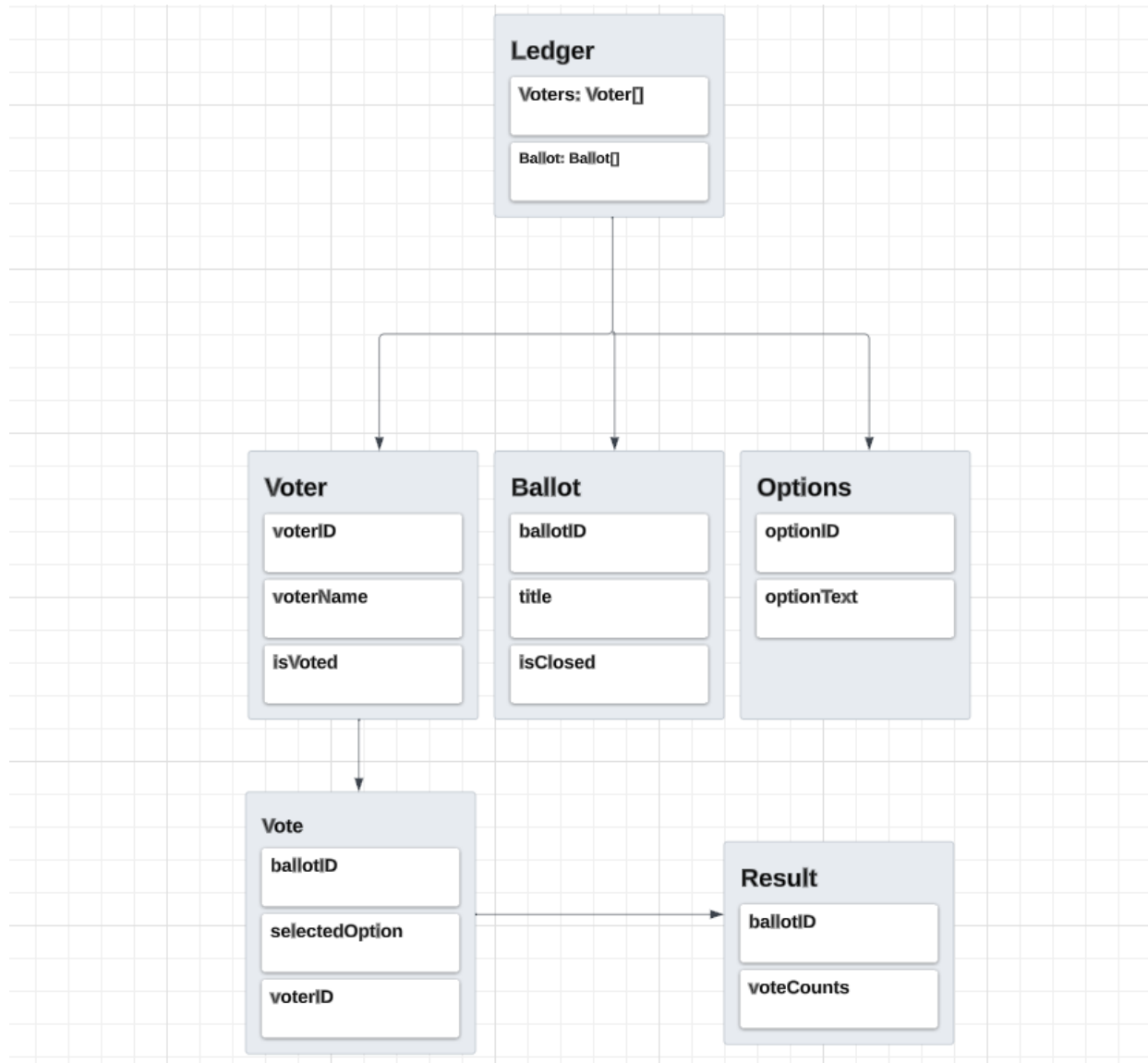
**Compiling the Results:** Smart contracts tally election results in real-time. Each ballot has its own smart contract and storage, and when an election is completed, the final result is announced publicly.

**Vote Verification:** Each voter receives a transaction ID for their vote. They can verify their vote by presenting their transaction ID to a government official, who can use the blockchain explorer to locate the corresponding transaction on the blockchain, ensuring transparency and accountability.

In summary, the proposed system leverages blockchain technology to create a secure and transparent election process, from voter registration to result compilation and vote verification, while maintaining voter anonymity and ensuring the integrity of the electoral process.

## PROPOSED WORK

### UML Diagram For Proposed Work:



Our proposed work on the "Ballot" smart contract prioritizes the development of a decentralized voting system that embodies simplicity, transparency, and security. Our proposed work is guided by several key principles, showcasing our commitment to creating a reliable and trustworthy solution for recording votes on the blockchain.

**Achieving Clarity and Readability:**

We aim to prioritize code readability by implementing clear variable names, meaningful function names, and concise yet comprehensive comments. Our goal is to enhance accessibility for developers, fostering collaboration and minimizing the risk of errors or misunderstandings.

**Focusing on Modularity and Reusability:**

Our design places a strong emphasis on modularity to improve maintainability and facilitate future enhancements or reuse of specific components. We intend to organize the code into distinct functions and modular components, utilizing structures like structs and modifiers for efficiency and flexibility in managing different aspects of the voting process.

**Implementing a Security-First Mindset:**

We are committed to integrating a security-first mindset throughout the design and implementation of the smart contract. Our approach involves adhering to common security patterns and best practices, including input validation checks, controlled access using modifiers, and explicit conditions to prevent unauthorized actions.

**Ensuring Transparency and Accessibility:**

Our focus is on making information about voters, choices, and the ballot's state publicly accessible through state variables and public functions. Leveraging the inherent transparency of the blockchain, we aim to build trust among stakeholders by providing visibility into the entire voting process.

**Establishing an Immutable Ledger:**

We plan to leverage the immutability of the blockchain to create a tamper-resistant and verifiable voting record. Our objective is to ensure that once a vote is cast, it becomes an indelible part of the ledger, upholding the integrity of the voting process and instilling confidence in the final results.

**Facilitating Progressive State Transition:**

Ensuring a controlled and well-defined progression of states is paramount to the reliability of the voting system. We are designing functions to transition the state only when specific conditions are met, contributing to the systematic flow of the voting process.

**Prioritizing Gas Efficiency:**

Our focus is on prioritizing gas efficiency to optimize the cost-effectiveness of transactions on the blockchain. We aim to carefully consider operations that minimize gas consumption and use storage variables judiciously to mitigate excessive gas costs, ensuring economic viability.

## ALGORITHM

### **Step 1. Contract Initialization:**

The initiation of the contract involves providing essential details such as the official's address, official's name, proposal description, and an array of choices through the constructor. To maintain the integrity of the voting structure, the constructor mandates the provision of exactly three choices during initialization.

### **Step 2. Adding Voters:**

In the "Created" state, the official, as the sole authorized entity, can add voters to the system using the addVoter function. Each voter is uniquely identified by their address and is associated with a name. This phase ensures that voter registration occurs before the voting process begins, providing an organized and controlled start to the election.

### **Step 3. Starting the Vote:**

Upon completing the voter registration phase, the official triggers the commencement of the voting process using the startVote function. This action transitions the contract from the "Created" state to the "Voting" state, signaling the beginning of the voting period.

### **Step 4. Casting Votes:**

During the "Voting" state, eligible voters can cast their votes using the doVote function. Voters specify their choices, and the contract validates the eligibility of the voter (ensuring they haven't voted before) and the validity of the chosen option. The contract meticulously keeps track of the total number of votes and increments the count for each specific choice.

### **Step 5. Ending the Vote:**

Once the voting period concludes, the official concludes the process by invoking the endVote function. This action transitions the contract from the "Voting" state to the "Ended" state, signaling the completion of the voting process.

### **Step 6. Querying Results:**

Post the "Ended" state, anyone, including the official, can query the results using functions like getChoiceDescription and getVoteCounts. These functions provide valuable insights into the choices made by voters, allowing for transparent verification of the results and fostering trust in the decentralized voting system.

## CODE

### Smart Contract Code:

```
// SPDX-License-Identifier: MIT
pragma solidity >= 0.7.0 <0.9.0;

contract Ballot {
    // VARIABLES
    struct Vote {
        address voterAddress;
        uint8 choice; // Use uint8 for representing choices (0 for
Chinese, 1 for Italian, 2 for Spanish)
    }

    struct Voter {
        string voterName;
        bool voted;
    }

    uint[] private choiceCounts; // Array to store the counts for each
choice
    uint public totalVoter = 0;
    uint public totalVote = 0;

    address public ballotOfficialAddress;
    string public ballotOfficialName;
    string public proposal;
    string[] public choices; // Array to store the choices

    mapping(address => Voter) public voterRegister;
    mapping(uint => Vote) private votes;

    enum State { Created, Voting, Ended }
    State public state;

    // MODIFIERS
    modifier condition(bool _condition) {
        require(_condition, "Condition not met");
    }
```

```

        _;
    }

    modifier onlyOfficial() {
        require(msg.sender == ballotOfficialAddress, "Not authorized");
        _;
    }

    modifier inState(State _state) {
        require(state == _state, "Invalid state");
        _;
    }

    // FUNCTION
    constructor(
        string memory _ballotOfficialName,
        string memory _proposal,
        string[] memory _choices
    ) {
        require(_choices.length == 3, "Must provide exactly three
choices");
        ballotOfficialAddress = msg.sender;
        ballotOfficialName = _ballotOfficialName;
        proposal = _proposal;
        state = State.Created;
        choiceCounts = new uint[](3); // Initialize the array with three
elements
        choices = _choices;
    }

    function addVoter(
        address _voterAddress,
        string memory _voterName
    ) public
        inState(State.Created)
        onlyOfficial
    {
        Voter memory v;
        v.voterName = _voterName;
        v.voted = false;
    }

```

```

        voterRegister[_voterAddress] = v;
        totalVoter++;
    }

    function startVote()
        public
        inState(State.Created)
        onlyOfficial
    {
        state = State.Voting;
    }

    function doVote(uint8 _choice)
        public
        inState(State.Voting)
        returns (bool voted)
    {
        require(_choice < 3, "Invalid choice"); // Ensure the choice is
within the valid range

        bool isFound = false;
        if(bytes(voterRegister[msg.sender].voterName).length != 0
            && !voterRegister[msg.sender].voted)
        {
            voterRegister[msg.sender].voted = true;
            Vote memory v;
            v.voterAddress = msg.sender;
            v.choice = _choice;
            choiceCounts[_choice]++;
            votes[totalVote] = v;
            totalVote++;
            isFound = true;
        }
        return isFound;
    }

    function endVote()
        public
        inState(State.Voting)
        onlyOfficial

```

```
{
    state = State.Ended;
    // You can now access the counts for each choice using
    choiceCounts[0], choiceCounts[1], choiceCounts[2]
}

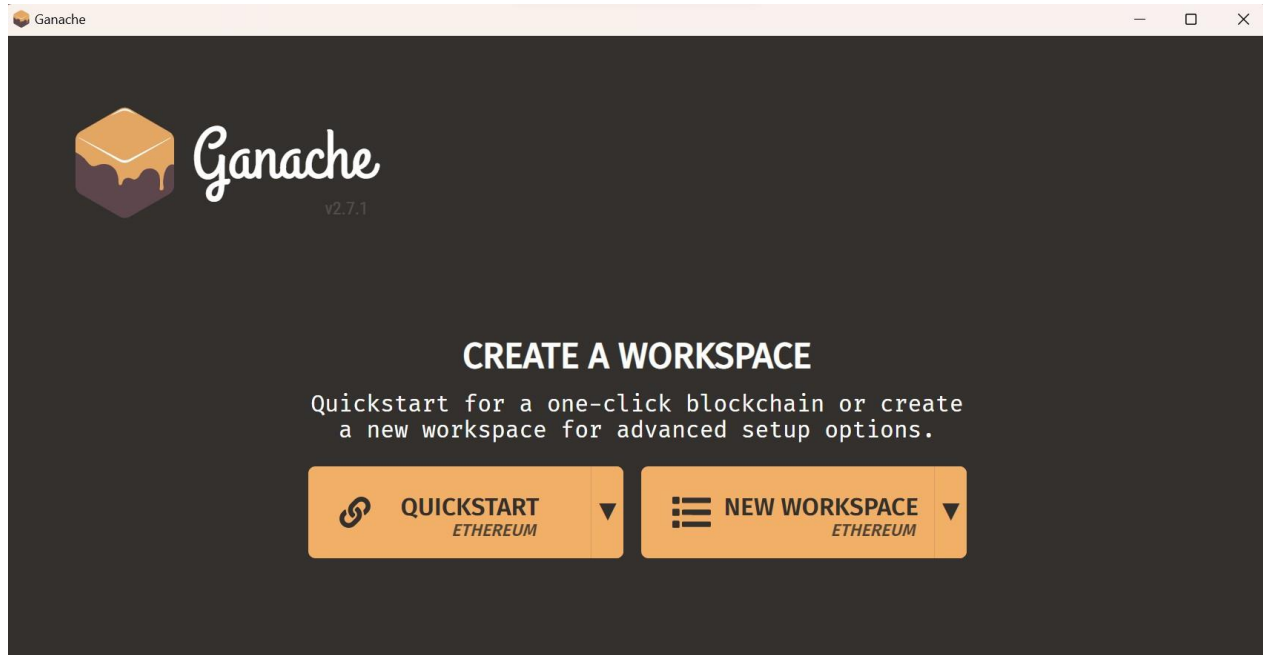
function getChoiceDescription(uint8 _choice) public view returns
(string memory) {
    require(_choice < 3, "Invalid choice");
    return choices[_choice];
}

function getVoteCounts() public view returns (uint, uint, uint) {
    require(state == State.Ended, "Voting has not ended yet");
    return (choiceCounts[0], choiceCounts[1], choiceCounts[2]);
}
}
```

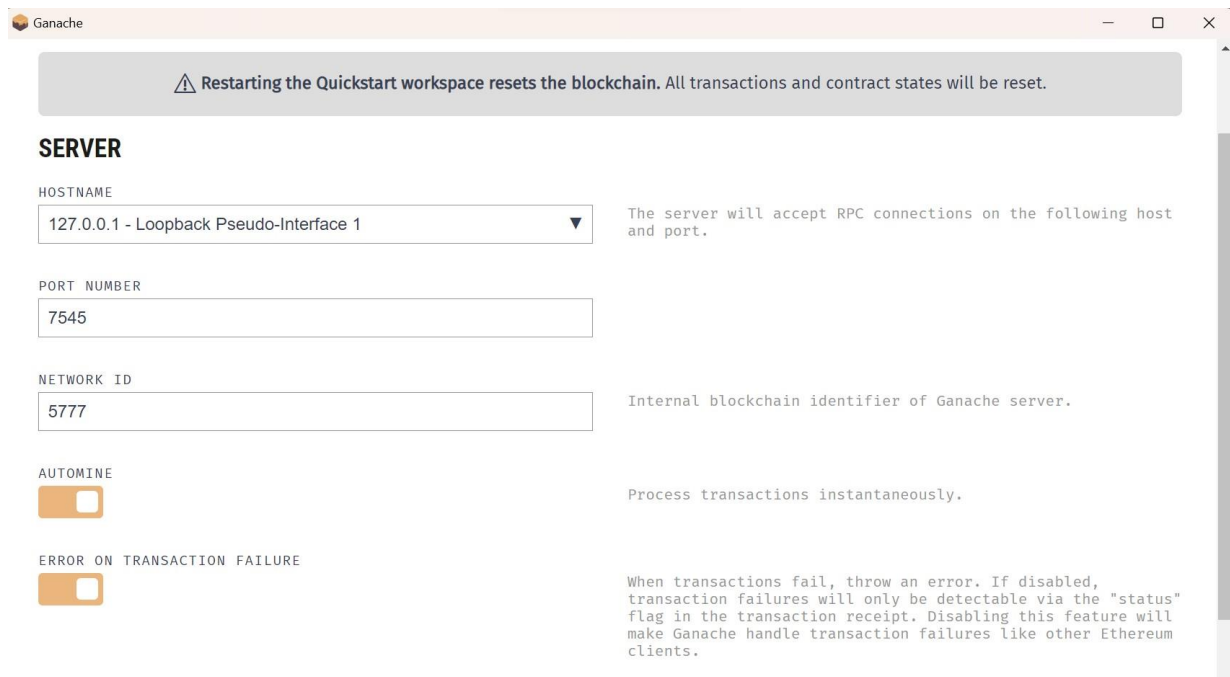


## RESULTS

### Opening Ganache:



### Configuring Local Blockchain:



Ganache

WORKSPACE SERVER ACCOUNTS & KEYS CHAIN ADVANCED ABOUT

CANCEL RESTART

⚠ Restarting the Quickstart workspace resets the blockchain. All transactions and contract states will be reset.

## ACCOUNTS & KEYS

ACCOUNT DEFAULT BALANCE

100

The starting balance for accounts, in Ether.

TOTAL ACCOUNTS TO GENERATE

10

Total number of Accounts to create and pre-fund.

AUTOGENERATE HD MNEMONIC

☐

Turn on to automatically generate a new mnemonic and account addresses on each run.

rug twenty filter scale cash require tenant intact conduct man inherit m

Enter the Mnemonic you wish to use.

note: this mnemonic is not secure; don't use it on a public blockchain.

LOCK ACCOUNTS

☐

If enabled, accounts will be locked on startup.

Ganache

WORKSPACE SERVER ACCOUNTS & KEYS CHAIN ADVANCED ABOUT

CANCEL SAVE AND RESTART

⚠ Restarting the Quickstart workspace resets the blockchain. All transactions and contract states will be reset.

## GAS

GAS LIMIT

6721975

Maximum amount of gas available to each block and transaction. Leave blank for default.

GAS PRICE

20000000000

The price of each unit of gas, in WEI. Leave blank for default.

## HARDFORK

HARDFORK

Istanbul

The hardfork to use. Default is Merge.

## Starting Local Blockchain:

The screenshot shows the Ganache application window. At the top, there's a navigation bar with icons for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below this is a status bar with various settings like CURRENT BLOCK, GAS PRICE, GAS LIMIT, HARDFORK, NETWORK ID, RPC SERVER, MINING STATUS, and WORKSPACE. The main area displays the MNEMONIC and HD PATH. Below that, a table lists five accounts with their addresses, balances, transaction counts, and indices.

ADDRESS	BALANCE	TX COUNT	INDEX
0x9e2c3f68576D55C5dc677Ff6EA305E409D77746a	100.00 ETH	0	0
0x36F4b2EE922f27c93CA2619c309B34dc24D0092B	100.00 ETH	0	1
0x51015c7304E6f40744D69f100FAC901b54296E70	100.00 ETH	0	2
0x7Ca9e0c9733C4A94D6B382707A4D4e3F76e143bd	100.00 ETH	0	3
0x52320febCEcc61815EE23c4C1EA207Bc3a120bD3	100.00 ETH	0	4

## Compiling Smart Contract:

The screenshot shows the Solidity Compiler interface. It includes a sidebar with icons for file management, search, and other functions. The main area displays the compiler version (0.8.0+commit.c7dfd78e) and options like Auto compile and Hide warnings. A progress bar indicates the compilation of 'votnext.sol'. Below the progress bar, there are buttons for 'Publish on Ipfs', 'Publish on Swarm', and 'Compilation Details'.

COMPILER 0.8.0+commit.c7dfd78e

Auto compile ☐ Hide warnings ☐

Advanced Configurations

Compile votnext.sol

Ctrl+S to compile contracts/votnext.sol

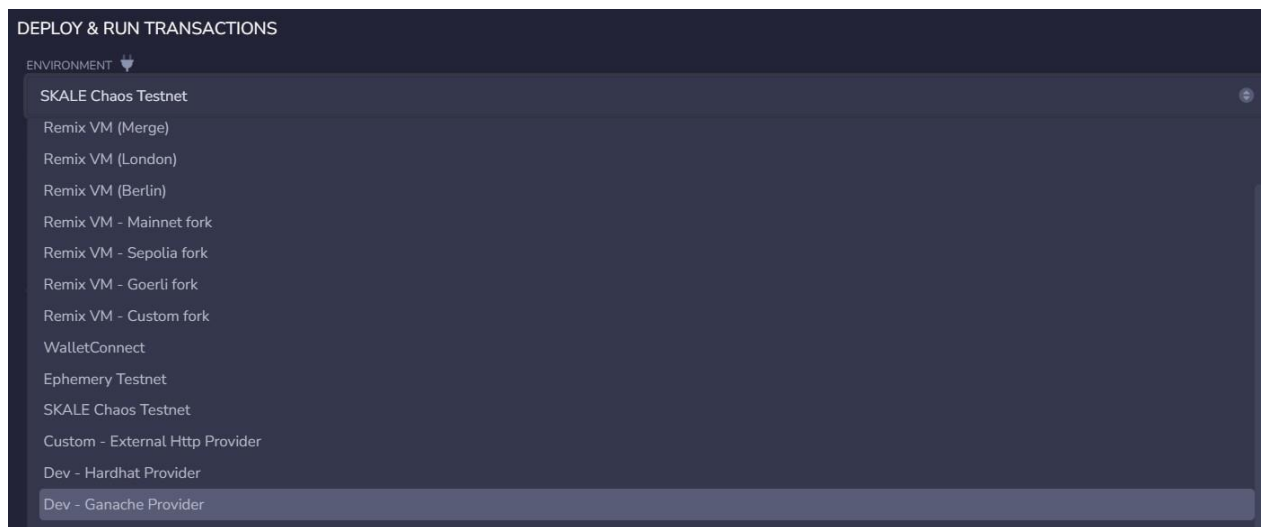
CONTRACT Ballot (votnext.sol)

Publish on Ipfs

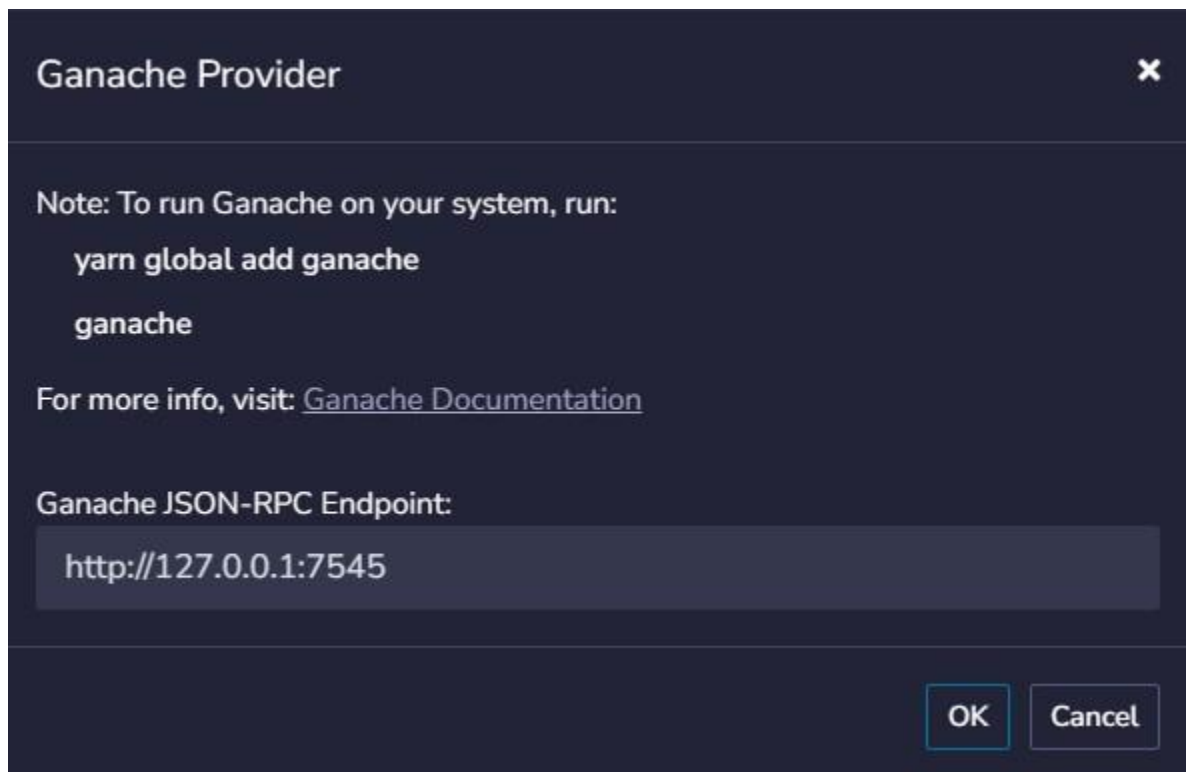
Publish on Swarm

Compilation Details


## Connecting To Local Ganache Blockchain:





## Enter Ganache RPC End Point:



## Configure Gas Limit & Select Admin Account:

ACCOUNT 


0x9e2...7746a (100 ether)  

GAS LIMIT


6721975

VALUE

0


Wei 

CONTRACT

Ballot - contracts/votenext.sol 

evm version: istanbul

## Enter Parameters:

DEPLOY 

\_BALLOTOFFICIALNAME:



"Election"

\_PROPOSAL:

"Which Party Do You Select"


\_CHOICES:

["BJP", "Congress", "AAP"]


 Calldata  Parameters 

transact

## Block Transaction:




[block:1 txIndex:-] from: 0x9e2...7746a  
to: Ballot.(constructor) value: 0 wei  
data: 0x608...00000 logs: 0 hash: 0x6a6...0f645

Debug 


status

0x1 Transaction mined and execution succeed


transaction hash

0x215067972042b029e6d49802af165f34ac8320cfd5941534bc4095  
edf5ba71ca 


block hash

0x6a612ed13d6e4b571157f51d9fe1e4bba580e826e94c9332ff5897  
fabf40f645 


block number

1 


contract address

0xae9a2075432456ce3833ab128825de06b9b8927e 

from

0x9e2c3f68576d55c5dc677ff6ea305e409d77746a 

to

Ballot.(constructor) 

## Deployed Contract:

▼ BALLOT AT 0XAE9...8927E (BLOCKCHAIN)

Balance: 0. ETH

addVoter

address \_voterAddress, string \_voterName

▼

doVote

uint8 \_choice

▼

endVote

startVote

ballotOfficialA...

ballotOfficialN...

choices

uint256

▼

getChoiceDesc...

uint8 \_choice

▼

getVoteCounts

proposal

state

totalVote

## Adding Verified Voter Blocks:

transact to Ballot.addVoter pending ...

✓

[block:2 txIndex:-] from: 0x9e2...7746a  
to: Ballot.addVoter(address,string) 0xae9...8927e value: 0 wei data: 0xd9e...00000  
logs: 0 hash: 0x1f9...f337e

Debug ▼

transact to Ballot.addVoter pending ...

✓

[block:3 txIndex:-] from: 0x9e2...7746a  
to: Ballot.addVoter(address,string) 0xae9...8927e value: 0 wei data: 0xd9e...00000  
logs: 0 hash: 0x54b...2cc9d

Debug ▼

transact to Ballot.addVoter pending ...

✓

[block:4 txIndex:-] from: 0x9e2...7746a  
to: Ballot.addVoter(address,string) 0xae9...8927e value: 0 wei data: 0xd9e...00000  
logs: 0 hash: 0x826...34045

Debug ▼

transact to Ballot.addVoter pending ...

✓

[block:5 txIndex:-] from: 0x9e2...7746a  
to: Ballot.addVoter(address,string) 0xae9...8927e value: 0 wei data: 0xd9e...00000  
logs: 0 hash: 0x9d5...fea35

Debug ▼

## Getting Descriptions Of Party Options:

CALL

[call] from: 0x9e2c3f68576D55C5dc677Ff6EA305E409D77746a  
to: Ballot.getChoiceDescription(uint8) data: 0xd12...00000

Debug

from

0x9e2c3f68576D55C5dc677Ff6EA305E409D77746a

to

Ballot.getChoiceDescription(uint8)

0xae9A2075432456ce3833ab128825dE06b9b8927E

input

0xd12...00000

decoded input

{  
    "uint8 \_choice": 0  
}

decoded output

{  
    "0": "string: BJP"  
}

logs

[]

call to Ballot.getChoiceDescription

## Verifying Voter Identity:

voterRegister

:

0x36F4b2EE922f27c93CA2619c309B34dc24D0092B"

Calldata


Parameters

call

0: string: voterName Rehan








1: bool: voted false

## Starting Vote Process:



**[block:6 txIndex:-]** from: 0x9e2...7746a  
to: Ballot.startVote() 0xae9...8927e value: 0 wei  
data: 0x4c0...a6af0 logs: 0 hash: 0x334...d4f85

Debug ^



status	0x1 Transaction mined and execution succeed
transaction hash	0xb65678f8e85265718e3bab394465e568e32d13c55131bacf0f350ee5d12682e1 
block hash	0x334579b3cea7ef2ba023bd2a22dec5f0e0fe63e1fa2d36acfc739bf1ea4d4f85 
block number	6 
from	0x9e2c3f68576d55c5dc677ff6ea305e409d77746a 
to	Ballot.startVote() 0xae9a2075432456ce3833ab128825de06b9b8927e 
gas	gas 
transaction cost	43905 gas 

## Voting From 3 Different People:


doVote ^

\_choice:

0

 Calldata  Parameters

transact





**[block:7 txIndex:-]** from: 0x36f...0092b  
to: Ballot.doVote(uint8) 0xae9...8927e value: 0 wei  
data: 0xe9b...00000 logs: 0 hash: 0x409...02cb8

Debug v

doVote ^

\_choice:

0

 Calldata  Parameters

transact



[block:8 txIndex:-] from: 0x510...96e70

to: Ballot.doVote(uint8) 0xae9...8927e value: 0 wei

data: 0xe9b...0000 logs: 0 hash: 0x9e7...c3120

Debug

▼

doVote

\_choice:

2

Calldata

Parameters

transact

[block:11 txIndex:-] from: 0x7ca...143bd

to: Ballot.doVote(uint8) 0xae9...8927e value: 0 wei

data: 0xe9b...00002 logs: 0 hash: 0x87a...ef3aa

Debug

▼

# Voting Process Ended:

transact to Ballot.endVote pending ...

[block:12 txIndex:-] from: 0x9e2...7746a

to: Ballot.endVote() 0xae9...8927e value: 0 wei data: 0xb92...23946

logs: 0 hash: 0xd2d...a099a

Debug

^

status

0x1 Transaction mined and execution succeed

transaction hash

0x27c6a37e3dfc785e0c09c6c890affe6120223ac36f847c2a3971817ff8a8f21b

block hash

0xd2d3eb9180867cd0f2f08de00bfd7465fdb1716ff3344ae5e282db66901a099a

block number

12

from

0x9e2c3f68576d55c5dc677ff6ea305e409d77746a

to

Ballot.endVote() 0xae9a2075432456ce3833ab128825de06b9b8927e

## Get Vote Counts:

**getVoteCounts**

**getVoteCounts - call**

0: uint256: 2

1: uint256: 0

2: uint256: 1

CALL **[call]** from: 0x9e2c3f68576D55C5dc677Ff6EA305E409D77746a to: Ballot.getVoteCounts() data: 0xffc...97b20 Debug

**from** 0x9e2c3f68576D55C5dc677Ff6EA305E409D77746a

**to** Ballot.getVoteCounts() 0xae9A2075432456ce3833ab128825dE06b9b8927E

**input** 0xffc...97b20

**decoded input** {}

**decoded output**

```
{
  "0": "uint256: 2",
  "1": "uint256: 0",
  "2": "uint256: 1"
}
```

## Getting App States (App State 2 means voting has ended) & Total Voters:

**state**

0: uint8: 2

**totalVote**

0: uint256: 3

Checking Vote Status of Voter:

voterRegister

:

"0x36F4b2EE922f27c93CA2619c309B34dc24D0092B"

Calldata

Parameters

call


0: string: voterName Rehan

1: bool: voted true

Blockchain Transaction History:

<div><div>TX HASH</div><div>0x27c6a37e3dfc785e0c09c6c890affe6120223ac36f847c2a3971817ff8a8f21b</div></div>	<div>CONTRACT</div> <div>CALL</div>
<div><div>FROM ADDRESS</div><div>0x9e2c3f68576D55C5dc677Ff6EA305E409D77746a</div></div> <div><div>TO CONTRACT ADDRESS</div><div>0xae9A2075432456ce3833ab128825dE06b9b8927E</div></div> <div><div>GAS USED</div><div>28948</div></div> <div><div>VALUE</div><div>0</div></div>	
<div><div>TX HASH</div><div>0x3c919a541b157d52a5f07fae05ec7ff400cbfb9c21aac1cdc07018531691f456</div></div>	<div>CONTRACT</div> <div>CALL</div>
<div><div>FROM ADDRESS</div><div>0x7Ca9e0c9733C4A94D6B382707A4D4e3F76e143bd</div></div> <div><div>TO CONTRACT ADDRESS</div><div>0xae9A2075432456ce3833ab128825dE06b9b8927E</div></div> <div><div>GAS USED</div><div>97272</div></div> <div><div>VALUE</div><div>0</div></div>	
<div><div>TX HASH</div><div>0xa7cdf88ed8c2a3999c07c2da18b89b26b3d23cd59181a3949337b353c4965013</div></div>	<div>CONTRACT</div> <div>CALL</div>
<div><div>FROM ADDRESS</div><div>0x51015c7304E6f40744D69f100FAC901b54296E70</div></div> <div><div>TO CONTRACT ADDRESS</div><div>0xae9A2075432456ce3833ab128825dE06b9b8927E</div></div> <div><div>GAS USED</div><div>24809</div></div> <div><div>VALUE</div><div>0</div></div>	
<div><div>TX HASH</div><div>0x199a1b60430e6243ee0bccd881f104e82ef47f72d30d9cfecb908ae5aeaf6ec0</div></div>	<div>CONTRACT</div> <div>CALL</div>
<div><div>TX HASH</div><div>0xdf1b895e7ab576fb10e38936b746e2bd78d3342b1286373ac83d80068d2bc283</div></div>	<div>CONTRACT</div> <div>CALL</div>
<div><div>FROM ADDRESS</div><div>0x51015c7304E6f40744D69f100FAC901b54296E70</div></div> <div><div>TO CONTRACT ADDRESS</div><div>0xae9A2075432456ce3833ab128825dE06b9b8927E</div></div> <div><div>GAS USED</div><div>82260</div></div> <div><div>VALUE</div><div>0</div></div>	
<div><div>TX HASH</div><div>0x3aa950ffb38739d70eefc2a1fc168484a65ff3f506c9301364e8769094341319</div></div>	<div>CONTRACT</div> <div>CALL</div>
<div><div>FROM ADDRESS</div><div>0x36F4b2EE922f27c93CA2619c309B34dc24D0092B</div></div> <div><div>TO CONTRACT ADDRESS</div><div>0xae9A2075432456ce3833ab128825dE06b9b8927E</div></div> <div><div>GAS USED</div><div>112260</div></div> <div><div>VALUE</div><div>0</div></div>	
<div><div>TX HASH</div><div>0xb65678fbe85265718e3bab394465e568e32d13c55131bacf0f350ee5d12682e1</div></div>	<div>CONTRACT</div> <div>CALL</div>
<div><div>FROM ADDRESS</div><div>0x9e2c3f68576D55C5dc677Ff6EA305E409D77746a</div></div> <div><div>TO CONTRACT ADDRESS</div><div>0xae9A2075432456ce3833ab128825dE06b9b8927E</div></div> <div><div>GAS USED</div><div>43905</div></div> <div><div>VALUE</div><div>0</div></div>	
<div><div>TX HASH</div><div>0x65dfd91b67627735486ad6aad03ad73fb6357a8d226824c9d67ee8bada930176</div></div>	<div>CONTRACT</div> <div>CALL</div>

### Testnet Blockchain Ether Balance After Voting Process:

ACCOUNT 	
0x9e2...7746a	(99.96082482 ether)
0x9e2...7746a	(99.96082482 ether)
0x36F...0092B	(99.9977548 ether)
0x510...96E70	(99.99736244 ether)
0x7Ca...143bd	(99.99805456 ether)

## CONCLUSION

The results in our project emphasize the implementation of immutable ledgers in the decentralized voting system, showcasing the robustness of the developed smart contract in handling various aspects of the voting process within a blockchain environment. The successful execution underscores the potential of blockchain technology to revolutionize and secure democratic processes, setting the stage for decentralized governance systems to thrive with transparency and trust.

After successfully navigating through the various stages of deploying and interacting with the smart contract on the local Ganache blockchain, the conclusion is evident: the entire process demonstrates the seamless integration of blockchain technology into the voting system. Opening Ganache sets the stage for a local blockchain environment, while configuring and starting the blockchain ensures a controlled and secure space for smart contract operations.

Compiling the smart contract ensures that it is ready for deployment, and connecting to the local Ganache blockchain further establishes a direct link for deploying and interacting with the contract. The process involves crucial steps such as configuring gas limits, selecting admin accounts, and entering parameters, which collectively contribute to the smooth execution of transactions on the blockchain.

Witnessing the deployment of the smart contract is a pivotal moment, marking the initiation of the decentralized voting system. Subsequent actions, such as adding verified voter blocks and obtaining descriptions of party options, reflect the dynamic and versatile capabilities of the deployed contract.

The verification of voter identity, commencement of the voting process, and the subsequent casting of votes from multiple participants highlight the accessibility and inclusivity of the system. The conclusive end of the voting process, the retrieval of vote counts, and the examination of the application states and total voters affirm the transparent and auditable nature of the blockchain-based voting system.

The examination of individual voter status, exploration of blockchain transaction history, and the verification of the testnet blockchain's ether balance post-voting process add layers of transparency and accountability to the entire workflow.

## REFERENCES

1. M. S. Farooq, U. Iftikhar and A. Khelifi, "A Framework to Make Voting System Transparent Using Blockchain Technology," in *IEEE Access*, vol. 10, pp. 59959-59969, 2022, doi: 10.1109/ACCESS.2022.3180168.
2. Alvi, Syada Tasmia, et al. "DVTChain: A Blockchain-Based Decentralized Mechanism to Ensure the Security of Digital Voting System Voting System." *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, 1 July 2022.
3. Yu, B. et al. (2018). Platform-Independent Secure Blockchain-Based Voting System. In: Chen, L., Manulis, M., Schneider, S. (eds) *Information Security. ISC 2018. Lecture Notes in Computer Science()*, vol 11060. Springer, Cham.
4. Yi, Haibo. "Securing E-Voting Based on Blockchain in P2P Network." *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 28 May 2019
5. Denis González, Camilo, et al. "Electronic Voting System Using an Enterprise Blockchain." *Applied Sciences*, vol. 12, no. 2, 1 Jan. 2022, p. 531
6. Alvi, Syada Tasmia, et al. "Digital Voting: A Blockchain-Based E-Voting System Using Biohash and Smart Contract." *IEEE Xplore*, 1 Aug. 2020
7. S. T. Alvi, L. Islam, T. Y. Rashme and M. N. Uddin, "BSEVOTING: A Conceptual Framework to Develop Electronic Voting System using Sidechain," 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Semarang, Indonesia, 2021, pp. 10-15, doi: 10.23919/EECSI53397.2021.9624282.
8. Li C, Xiao J, Dai X, Jin H. AMVchain: authority management mechanism on blockchain-based voting systems. *Peer Peer Netw Appl.* 2021;14(5):2801-2812. doi: 10.1007/s12083-021-01100-x. Epub 2021 Mar 11. PMID: 33723494; PMCID: PMC7947943.
9. Khan, Kashif Mehboob, et al. "Investigating Performance Constraints for Blockchain Based Secure E-Voting System." *Future Generation Computer Systems*, vol. 105, Apr. 2020, pp. 13–26, <https://doi.org/10.1016/j.future.2019.11.005>.
10. Abuidris, Y., Kumar, R., Yang, T. and Onginjo, J. (2021), Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding. *ETRI Journal*, 43: 357-370. <https://doi.org/10.4218/etrij.2019-0362>

11. Pawlak, Michał, et al. "Towards the Intelligent Agents for Blockchain E-Voting System." *Procedia Computer Science*, vol. 141, 2018, pp. 239–246, <https://doi.org/10.1016/j.procs.2018.10.177>.
12. Shujaa, Mohamed & Uddin, Ahmed & muhsen, Ahmed. (2023). Secure blockchain e-voting system using speck cipher. 10.1063/5.0154775.
13. Y. Li et al., "A Blockchain-Based Self-Tallying Voting Protocol in Decentralized IoT," in *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 119-130, 1 Jan.-Feb. 2022, doi: 10.1109/TDSC.2020.2979856.
14. Halderman, J.A., Teague, V. (2015). The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election. In: Haenni, R., Koenig, R., Wikström, D. (eds) *E-Voting and Identity. Vote-ID 2015. Lecture Notes in Computer Science()*, vol 9269. Springer, Cham.
15. Ayed, Ahmed Ben. "A Conceptual Secure Blockchain Based Electronic Voting System." *International Journal of Network Security & Its Applications* 9 (2017): 01-09
16. Kukwase, Praful, et al. "Blockchain Based E-Voting System." *International Journal of Research in Engineering and Science (IJRES) ISSN*, vol. 10, no. 5, 2022, pp. 74–76.