SMART CONTRACT BASED VOTING SYSTEM

[1]NISHAD SAYYED , [2]SAYALI MIRGE, [3]RAJESHWARI SATPUTE ,[4]RACHANA KSHIRSAGAR, [5]ANIL NAIK

[1]Student , Department of Computer Engineering, S.Y.P Shreeyash College of Engineering and Technology (Polytechnic) ,Aurangabad / Chh. Sambhajinagar , India.
[2]Student , Department of Computer Engineering, S.Y.P Shreeyash College of Engineering and Technology (Polytechnic) ,Aurangabad / Chh. Sambhajinagar , India.
[3]Student , Department of Computer Engineering, S.Y.P Shreeyash College of Engineering and Technology (Polytechnic) ,Aurangabad / Chh. Sambhajinagar , India.
[4]Guide(lecturer) , Department of Computer Engineering, S.Y.P Shreeyash College of Engineering and Technology (Polytechnic) ,Aurangabad / Chh. Sambhajinagar , India.
[5]HOD , Department of Computer Engineering, S.Y.P Shreeyash College of Engineering and Technology (Polytechnic) ,Aurangabad / Chh. Sambhajinagar , India.

---***---

Abstract - *The integrity and transparency of electoral processes are critical in any democratic system. Traditional voting systems often face challenges such as voter fraud, vote manipulation, lack of transparency, and inefficiency in vote counting. Blockchain technology, with its decentralized, immutable, and transparent nature, offers a promising solution. This project proposes a* ***Smart Contract-Based Voting System*** *that leverages blockchain to conduct secure, tamper-proof, and transparent elections. Smart contracts automate the voting process, ensuring that each vote is recorded immutably and counted accurately without the need for a central authority. The system enables voters to cast their votes online securely while maintaining voter privacy and ensuring the election results are verifiable by all stakeholders. This approach enhances trust, reduces administrative overhead, and provides a scalable solution for modern electoral challenges.*

1. INTRODUCTION

Voting is the cornerstone of democracy, allowing citizens to express their preferences and influence decision-making. Traditional voting methods, including paper ballots and centralized electronic voting machines, are prone to errors, fraud, and delays in vote counting. In addition, these systems often require significant administrative resources to ensure accuracy and security.

A Smart Contract-Based Voting System utilizes blockchain smart contracts to automate and enforce voting rules. Smart contracts are self-executing programs that run on the blockchain and execute predefined conditions automatically. In the context of voting, smart contracts can manage voter registration, vote casting, vote tallying, and result declaration without human intervention.

Key benefits of a blockchain-based voting system include:

- Security: Each vote is cryptographically secured and immutable.
- Transparency: All transactions (votes) are visible on the blockchain for verification.
- Efficiency: Automatic vote counting reduces time and human error.
- Trust: Eliminates the need for a central authority and reduces the possibility of tampering.

This project aims to develop a secure, transparent, and decentralized voting system that leverages smart contracts to enhance the integrity of elections, making it suitable for governmental, organizational, and community-level elections.

2. Problem Statement

Traditional voting systems, whether paper-based or electronic, face several critical challenges:

- **Fraud and Manipulation:** Votes can be tampered with or miscounted due to centralized control.
- **Lack of Transparency:** Voters cannot independently verify that their vote was recorded or counted correctly.
- **Inefficiency:** Manual vote counting is time-consuming and prone to human error.
- **Security Risks:** Centralized databases of votes are vulnerable to hacking or unauthorized access.

These issues reduce public trust in the electoral process. A secure, transparent, and decentralized voting system is required to ensure the integrity of elections, protect voter privacy, and streamline vote management.

System Architecture

1. Voter Registration Module:

- Voters register using their unique identity (e.g., ID, email, or blockchain wallet).
- Registration data is stored securely on the blockchain or a secure database.

2. Smart Contract Layer:

- Smart contracts manage the voting rules and process automatically.

Responsibilities include:

- Verifying voter eligibility
- Recording votes immutably
- Ensuring one vote per voter
- Counting votes in real-time

3. Voting Interface (Frontend):

- Provides a user-friendly interface for voters to cast their votes.
- Interacts with the smart contract to submit votes securely.

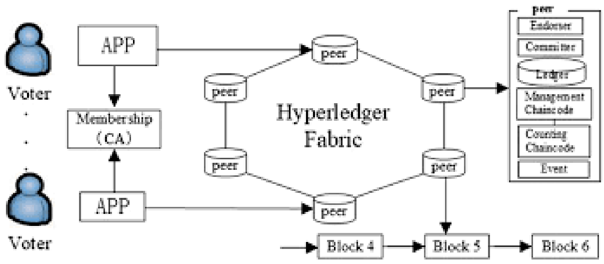
4.Block chain Network:

- Acts as a decentralized ledger where votes are recorded.
- Ensures transparency, immutability, and auditability of all votes.

Result Declaration Module:

Automatically tallies votes using smart contracts.

- Publishes results on the blockchain, allowing anyone to verify them.



BLOCKCHAIN OF SMART CONTRACT PROPOSED METHODOLODY

1.Voters (Users)

- These are the participants who want to cast their votes.
- Each voter interacts with the voting system through an **application (APP).**

2. Membership (CA - Certificate Authority)

- Before voting, each voter must be **authenticated and authorized** by the **Certificate Authority (CA).**
- The CA issues digital certificates to voters, verifying their identity and eligibility.
- This ensures that only registered and legitimate voters can participate.

3. Application (APP)

- The APP acts as the **user interface** for voters.
- Voters use the app to register, authenticate, and cast their votes.
- The app communicates with the underlying blockchain network to submit and retrieve data.

4. Hyperledger Fabric Network

- The core of the system is the **Hyperledger Fabric blockchain**, which is a permissioned blockchain framework.
- It consists of several **peers (nodes)**, each representing an organization or entity in the network.

5. Peers

- Peers maintain the ledger and smart contracts (called **chain code** in Fabric).
- They validate and record transactions (votes) on the blockchain.
- Peers communicate with each other to keep the ledger consistent and synchronized.

6. Transaction Flow in Hyperledger Fabric

- When a voter casts a vote, the APP sends the transaction proposal to the peers.
- Peers **endorse** the transaction by executing the smart contract and verifying the vote validity.
- Once endorsed, the transaction is sent to the **Ordering Service** to be ordered and grouped into blocks.

7. Ordering Service (Order)

- The order collects endorsed transactions from peers.
- It sequences transactions into blocks and distributes them back to peers in the network.

8. Ledger

- The ledger is the blockchain itself, where all validated transactions (votes) are stored immutably.
- The ledger contains two parts:

- **Blockchain:** Sequence of blocks containing transactions.
- **World State:** The current state of the data, such as the tally of votes or voter statuses.

9. Manage Chaincode (Smart Contract)

- Chaincode enforces voting rules such as:
  - One vote per voter
  - Valid candidate selection
  - Vote tallying logic
- The chaincode runs on peers and is invoked during transaction endorsement.

10. Counting Chaincode & Event

- The counting chaincode handles the aggregation and counting of votes.
- Events can be emitted from the blockchain to notify the application or administrators about transaction statuses or final results.

11. Blocks (Block 4, Block 5, Block 6, ...)

- Transactions are grouped into blocks by the orderer.
- Each block is appended sequentially to the blockchain ledger, ensuring data integrity and traceability.

Execution Flow-

1. **User Registration and Authentication**
   - Voters register through the **Membership Service Provider (CA)**, which issues digital certificates confirming their identity and eligibility.
   - Only authenticated voters can access the voting app.
2. **Voting via Application (APP)**
   - Voters log in to the APP and choose their preferred candidate.
   - The APP packages the vote as a transaction proposal and sends it to the blockchain network peers.
3. **Transaction Endorsement**
   - Peers receive the transaction proposal and simulate the smart contract (chain code) to verify the vote (e.g., check voter eligibility, prevent double voting).
   - If valid, peers endorse the transaction by signing it.
4. **Transaction Ordering**
   - Endorsed transactions are sent to the **Ordering Service**, which sequences and batches them into blocks.
5. **Block Distribution and Ledger Update**
   - Blocks are distributed to all peers.
   - Peers validate the block and append it to their local copy of the blockchain ledger.
   - The ledger now holds an immutable record of the vote.
6. **Vote Counting and Result Declaration**
   - Chaincode handles vote tallying automatically based on recorded votes.
   - The system emits events or updates that can be accessed by administrators and voters for transparent verification.

Table:-System Specifications and Development Prameters

Hardware requirements	Processor, RAM, Storage , Internet , Device
Software Requirements	OS, Solidity, JavaScript, Node.js, React.js, GitHub, Blockchain, HTML, CSS
Development Parameters	Blockchain Platform Selection (Ethereum, Hyperledger)
Security Parameters	Cryptographic hashing

Module Description

1. Voter Registration Module

- Handles voter enrollment and identity verification via the Certificate Authority (CA).
- Issues digital certificates used to authenticate voters on the blockchain network.

2. Voting Application (APP) Module

- User interface that allows voters to securely log in and cast their votes.
- Communicates with the blockchain network by sending transaction proposals and receiving feedback.

3. Smart Contract (Chaincode) Module

- Implements voting rules: one vote per voter, valid candidate selection, and vote tallying logic.
- Runs on peers during transaction endorsement to validate and process votes.
- Updates the world state and blockchain ledger with votes and results.

4. Peer Node Module

- Executes chaincode to validate and endorse transactions.

- Maintains a copy of the blockchain ledger and world state.
- Participates in consensus and shares ledger updates with other peers.

5. Ordering Service Module

- Collects endorsed transactions from peers.
- Orders and batches transactions into blocks to ensure a consistent transaction sequence.
- Distributes blocks to all peers for ledger update.

6. Ledger Module

- Stores the blockchain (immutable record of all transactions) and the world state (current vote counts and voter statuses).
- Provides transparency and auditability by allowing stakeholders to verify the election results.

7. Event Notification Module

- Emits events during key processes, such as vote submission confirmation and result announcements.
- Allows real-time updates and notifications to the APP and election administrators

CONCLUSION

The Smart Contract-Based Voting System built on Hyperledger Fabric offers a robust and trustworthy solution to many challenges faced by traditional voting methods. By leveraging blockchain's decentralization, immutability, and transparency, this system ensures secure voter authentication, tamper-proof vote recording, and automated, accurate vote counting. The use of smart contracts eliminates the need for intermediaries, reduces the risk of fraud, and allows all stakeholders to independently verify election results. Overall, this approach enhances the integrity and efficiency of the electoral process, making it well-suited for modern democratic systems and various organizational voting scenarios. Future improvements may focus on scalability, user experience, and integration with other digital identity platforms to further strengthen the system's usability and security.

REFERENCES

1. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System,"2008.[Online].Available: https://bitcoin.org/bitcoin.pdf

2. Hyperledger Fabric Documentation, Hyperledger, [Online]. Available:https://hyperledger-fabric.readthedocs.io/en/latest/

3. M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," OSDI '99 Proceedings of the Third Symposium on Operating Systems Design and Implementation, 1999.

4. A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol," Crypto, 2017.

5. Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data, 2017.

6. J. Jaiswal and A. Yadav, "Blockchain-based E-voting System to Make Election Transparent," International Journal of Computer Applications, vol. 180, no. 16, pp. 31–36, 2018.

7. E. Androulaki et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," Proceedings of the Thirteenth EuroSys Conference, 2018.

8. S. Kshetri, "1 Blockchain's roles in meeting key supply chain management objectives," International Journal of Information Management, vol. 39, 2018, pp. 80–89.

9. V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," Ethereum White Paper, 2013.[Online].Available: https://ethereum.org/en/whitepaper/

BIOGRAPHIES

1.MS. NISHAD SAYYED
Pursuing Poly(CO)
S.Y.P. SHREEYASH COLLEGE OF ENGINEERING AND TECHNOLOGY (POLYTECHNIC)

2.MS .SAYALI MIRGE
Pursuing Poly(CO)
S.Y.P. SHREEYASH COLLEGE OF ENGINEERING AND TECHNOLOGY (POLYTECHNIC)

3.MS. RAJESHWARI SATPUTE
Pursuing Poly(CO)
S.Y.P. SHREEYASH COLLEGE OF ENGINEERING AND TECHNOLOGY (POLYTECHNIC)

4.Ms.Prof.RACHANA KSHIRSAGAR
Dept. Of Computer Engineering
S.Y.P. SHREEYASH COLLEGE OF ENGINEERING AND TECHNOLOGY (POLYTECHNIC)

5.Mr.Prof.ANIL NAIK
Dept. Of Computer Engineering
S.Y.P. SHREEYASH COLLEGE OF ENGINEERING AND TECHNOLOGY (POLYTECHNIC)