BHARAT VOTE

Building Trust in Digital Democracy: A Polygon-Based E-Voting Platform

Debarshi Paul
CSE Core
Manipal University Jaipur

Aayush Anand
CSE (AI & ML)
Manipal University Jaipur

Harsh Srivastava
CSE Core
Manipal University Jaipur

Kumar Umang
CSE Core
Manipal University

I. Introduction

Traditional voting mechanisms, while foundational to democracy, suffer from significant logistical and security challenges. The procedure frequently uses a lot of resources, is slow, and depends on centralized authorities, which leads to failure points and chances for fraud or mistakes. The public's confidence in election results may be weakened by these weaknesses.

By bringing decentralization, immutability, and transparency, blockchain technology offers electoral systems a paradigm shift [1]. On a blockchain, a vote is a time-stamped, cryptographically secured transaction that is dispersed throughout a network of computers, making it nearly impossible to change or remove. This paper presents BharatVote, a practical proof-of-concept that leverages these principles to build a secure and transparent electronic voting system.

II. Foundational research

Electronic voting is not a new concept, but early implementations have often replicated the centralized architecture of paper-based systems. making them susceptible to online attacks. Voting on a blockchain has been suggested as a more reliable substitute [2].

A smart contract, which is a self-executing program with predefined rules stored on the blockchain, is the central component of our system [3]. Since a scalable voting application requires low transaction fees ("gas"), high throughput, and a strong developer ecosystem, we decided to build on the Polygon network, an Ethereum scaling solution [4]. The frontend is developed using modern web technologies to ensure a seamless user experience, connecting to the blockchain via a standard browser wallet like MetaMask

III. System architecture and methodology

The BharatVote platform consists of three primary components: the backend smart contract, the frontend user interface, and the underlying blockchain network.

A. Smart Contract Design The system's logic is encapsulated in a smart contract written in Solidity and deployed on the Polygon Amoy testnet. The contract has several key features:

- *Candidate Registry*: A publicly accessible list of candidates is hardcoded into the contract.
- *Voter Registry:* A mapping (address => bool) is used to track which wallet addresses have already cast a vote. This enforces the one-address-one-vote rule.
- *Voting Function:* The primary vote () function first verifies that the sender's address has not already voted using a require statement. If the check passes, it increments the vote count for the selected candidate and records the sender's address in the voter registry.

This design ensures that all voting rules are enforced automatically and transparently without the need for a central administrator.

B. Frontend Implementation The user interface is a web application built with Next.js and styled with TailwindCSS. Interaction with the blockchain is managed by the **Wagmi** and **Viem** libraries, which provide React hooks and utilities for wallet connection, data fetching from the smart contract, and transaction signing [5].

The frontend performs two main roles:

1. **Read Operations:** It fetches the list of candidates and their current vote counts directly from the smart contract in real-time.
2. **Write Operations:** When a user clicks "Vote," the frontend constructs a transaction that calls the vote() function in the smart contract and prompts the user to approve it through their MetaMask wallet.

C. User Workflow and Verification The process for a user to cast a vote is designed to be straightforward:

1. **Setup:** The user must have the MetaMask browser extension installed and configured for the Polygon Amoy testnet.
2. **Acquire Gas:** The user obtains free test POL tokens from a public faucet to pay for the transaction gas fee.
3. **Connect:** On the BharatVote website, the user clicks "Connect Wallet" to authorize the application.
4. **Vote:** The user selects a candidate and approves the voting transaction that pops up in MetaMask.
5. **Verification**: After a few seconds, the transaction is confirmed. The user receives a transaction hash, which can be used to independently verify their vote on a block explorer like PolygonScan.

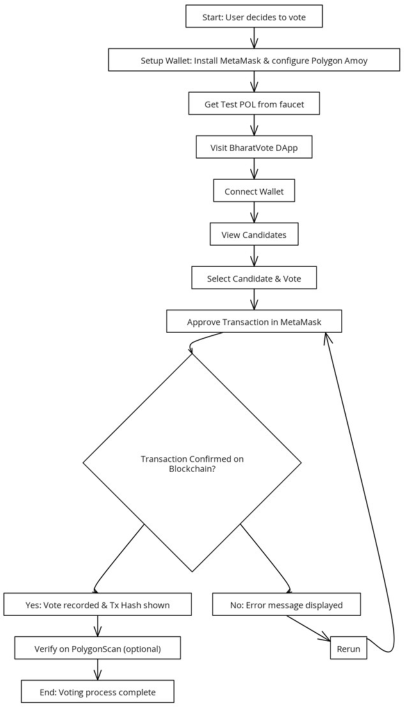
Start: User decides to vote
Setup Wallet: Install MetaMask & configure Polygon Amoy
Get Test POL from faucet
Visit BharatVote DApp
Connect Wallet
View Candidates
Select Candidate & Vote
Approve Transaction in MetaMask
Transaction Confirmed on Blockchain?
Yes: Vote recorded & Tx Hash shown
No: Error message displayed
Verify on PolygonScan (optional)
Rerun
End: Voting process complete

Fig1: User Voting Workflow

IV.RESULTS AND DISCUSSION

The BharatVote platform was successfully deployed and tested, demonstrating that all core functionalities perform as expected. Users were able to connect, vote, and see vote counts update in real-time. The on-chain verification process confirmed that votes were recorded immutably and correctly attributed.

The primary success of this project is its demonstration of **end-to-end transparency**. Any interested party can audit the election by reading the smart contract's data on the blockchain, verifying the total number of votes without needing to trust an intermediary.

However, as a proof-of-concept, the system has limitations. The most significant is the "one-address-one-vote" model, which is vulnerable to **Sybil attacks**, where an individual could create multiple wallets to vote multiple times. Furthermore, the link between a wallet address and a vote is public, which raises **voter privacy** concerns that would be unacceptable in a real-world political election

IV. CONCLUSION AND FUTURE PROSPECTS

BharatVote successfully validates the use of blockchain technology for creating a transparent and tamper-proof voting system. It serves as a strong foundation for future development by proving the viability of the core mechanics. Future work will focus on addressing the current limitations to move the project closer to a production-ready system. Key areas of development include:

- **Privacy Preservation:** Integrating **zero-knowledge proofs (ZK-proofs)** to allow voters to prove their eligibility and cast a ballot without revealing their identity or their choice [6].
- **Identity Verification:** Implementing a robust identity management system to link one vote to one real-world individual, thereby preventing Sybil attacks. This could involve integration with government digital identity frameworks.
- **Scalability and Accessibility:** Optimizing the contract for large-scale elections and developing mobile applications to ensure the system is accessible to all citizens.

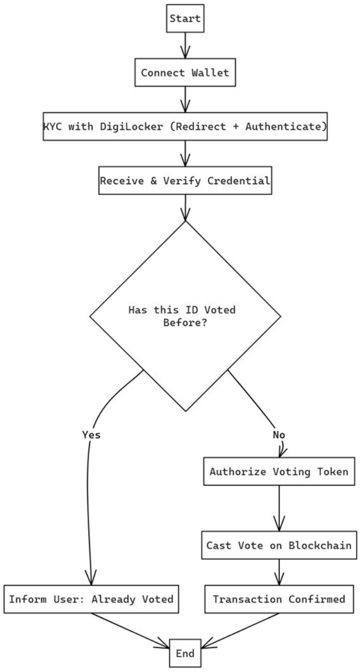
Start
Connect Wallet
KYC with DigiLocker (Redirect + Authenticate)
Receive & Verify Credential
Has this ID Voted Before?
Yes
No
Authorize Voting Token
Cast Vote on Blockchain
Inform User: Already Voted
Transaction Confirmed
End

Fig2: Implementation using Digilocker

Integrating digilocker provides a robust solution to the sybil attack problem by anchoring each vote to a unique, government-verified digital identity. Instead of the "one-address-one-vote" model, this shifts the paradigm to a true "one-citizen-one-vote" system. The process involves leveraging digilocker's api for kyc (know your customer) verification.

ACKNOWLEDGMENT

This project would not have been possible without the groundbreaking contributions of the entire blockchain ecosystem. We acknowledge the intellectual efforts of the developers behind the Ethereum platform, the Solidity programming language, and essential tools like MetaMask. On a philosophical level, the motivation for BharatVote is rooted in the democratic ideals formulated by Shri T.N. Seshan. His tenure as the Chief Election Commissioner of India remains a benchmark for electoral transparency and serves as the core inspiration for this project.

REFERENCES

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

[2] V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," unpublished, 2014. [Online]. Available: https://ethereum.org/en/whitepaper/

[3] NITI Aayog, "Blockchain: The India Strategy," Government of India, 2020. [Online]. Available: https://www.niti.gov.in/sites/default/files/2020-01/Blockchain_The_India_Strategy_Part_I.pdf

[4] A. Tanwar, A. Bhatia, and R. K. Singh, "A Blockchain-Based E-Voting System for Indian Elections," in *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, 2021.

[5] "Introduction to Smart Contracts," Solidity Documentation. [Online]. Available: https://docs.soliditylang.org/en/v0.8.20/introduction-to-smart-contracts.html

[6] "Introduction to Polygon Technology," Polygon Wiki. [Online]. Available: https://wiki.polygon.technology/

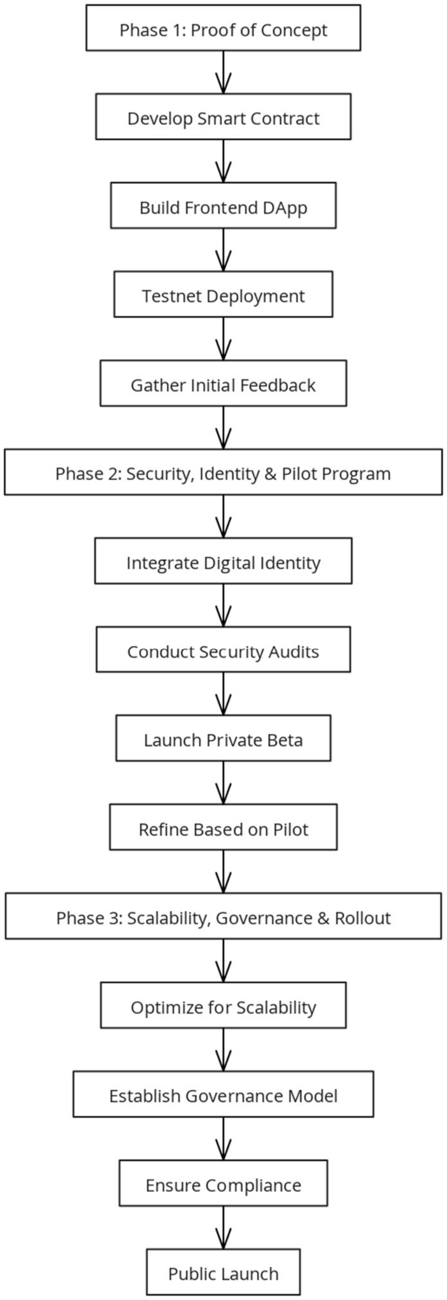

Fig 3:Roadmap for realistic deployment