Blockchain-Powered Voting Application

Matei Mihnea-Cristian, Clapa Adrian-Gabriel
"POLITEHNICA" University of Bucharest

1. Description

The application proposed is a secure and innovative platform to transform the traditional voting process by integrating the blockchain technology with a simple, yet effective, interface. The system provides a verifiable and tamper-proof record of votes, while ensuring the integrity of the voting process. It addresses the voting accessibility issue of disabled or isolated individuals, the transparency of the voting process, as all votes are viewed as public, immutable transactions on a public, well-known blockchain and security of the voting process. The decentralized voting system is deployed on the Ethereum-compatible Volta network, provided by energyweb and it leverages smart contracts to ensure the transparency, security and integrity of the voting process.

The user registration is facilitated by using Metamask as wallet. Metamask is an Ethereum wallet which provides a user-friendly and secure interface, therefore easing the participation in the election process.

The smart contract used in the application is written in Solidity, a curly-bracket language designed for Ethereum. It is responsible of voter authentication, candidate registration, voting and results retrieval and it is also responsible of enforcing the election's rules, such as no multiple voting, or voting within the timelimit. The smart contract implementation also allows for retrieving the candidates list and their respective vote counts.

The interface of the application is developed using HTML and JavaScript in order to provide an intuitive experience.

This voting system highlights the potential of blockchain technology in improving the integrity of the democratic process of voting. It underscores the power of using smart contracts in complex processes, while also reducing the need of intermediaries and single points of failure. Using the Volta network, the application makes use of its scalability and cost-effective environment for decentralized applications (dApps).

2. Links

2.1. GitHub Repository

The Blockchain-Powered Voting Application can be found on GitHub, here: https://github.com/MihneaMatei/BlockchainPoweredVotingApp

2.2. Documentation Used

- https://github.com/NomicFoundation/hardhat
- https://docs.ethers.org/v5/
- https://docs.soliditylang.org/en/v0.8.23/introduction-to-smart-contracts.html

- https://create-react-app.dev/docs/getting-started
- https://docs.metamask.io/wallet/how-to/connect/set-up-sdk/javascript/react/

3. Documentation

3.1. Abstract

The purpose of a blockchain-powered voting application is to create a transparent and secure voting system using blockchain technology. This approach provides a verifiable and tamper-proof record of votes while ensuring the integrity of the voting process, preventing fraud. Using a blockchain-based voting system will enable remote voting for individuals physically unable to go to the polling stations.

3.2. Issues Addressed

Accessibility: a blockchain-powered voting application would allow individuals with mobility issues, or individuals living in isolated areas to vote via remote voting.

Transparency: this application would ensure transparency of the voting process by utilizing a public blockchain, in which every vote is recorded as a transaction on the blockchain, visible to all participants.

Security: a blockchain application would provide a secure environment for voting by using public-key cryptography for voter authentication and end-to-end encryption for communication between the mobile app and the blockchain. The decentralized architecture specific to blockchain applications would reduce the risk of a single point of failure.

Fraud: by using a robust consensus mechanism, such as Proof-of-Work (PoW) or Proof-of-Stake (PoS), each vote will have to be agreed by all the nodes in the consensus group, preventing manipulation of the vote record by malicious actors.

3.3. Components

The components needed by the proposed application are:

- Smart contracts for vote recording and counting.
- User authentication system using a Public-Key Infrastructure.
- Front-end for users to cast votes.
- Real-time vote count display

3.4. Implementation

Technologies used in the implementation of the app are the following:

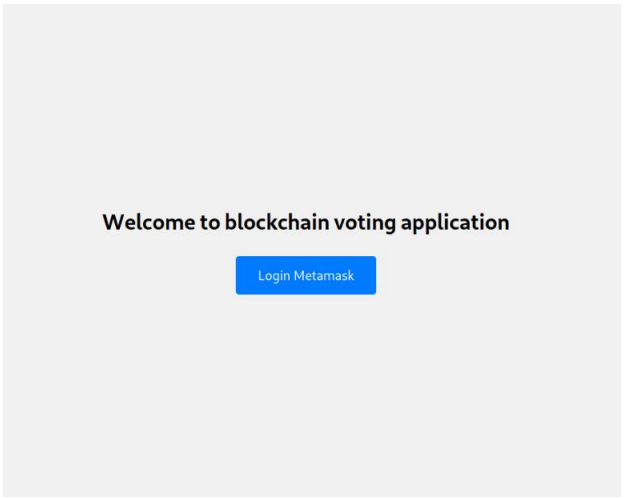
- User Interface – ReactJS;
- Smart Contract – Solidity, ethers.js and HardHat;
- Blockchain Connection – Metamask;

Implementation details:

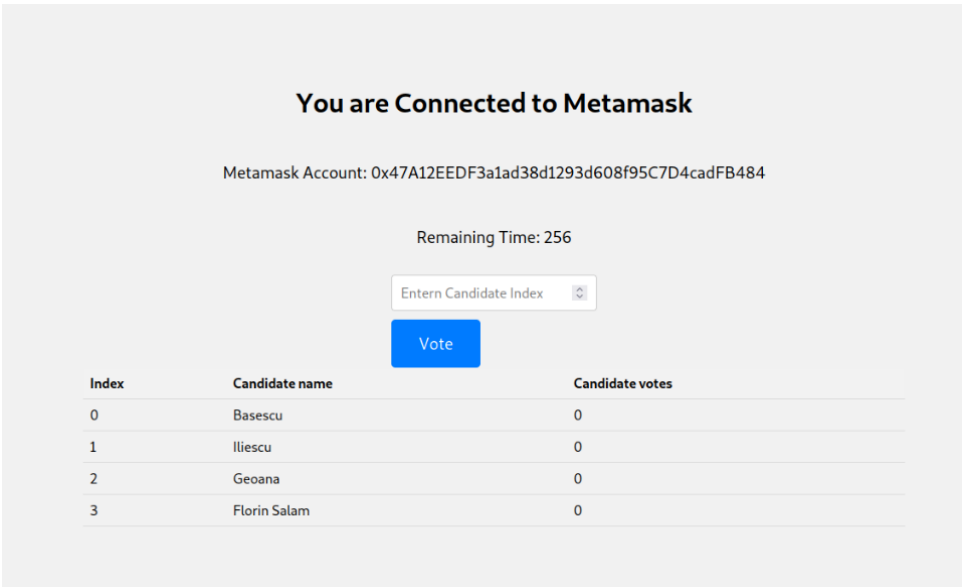
- The app uses a React UI and the Metamask browser extension to connect to the blockchain (a Metamask account is required).
- A SC is created using Solidity and then deployed to the Blockchain via Volta network.
- The SC contains a list of candidates, a list of voting accounts and a time limit for the voting process.
- It contains functions to vote, get the number of votes for each candidate, the remaining voting time and the voting status.
- To compile and deploy the contract, we used Hardhat.
- The contract is first compiled (npx hardhat compile) and then is deployed on the Volta network using the deploy script (npx hardhat run --network volta scripts/deploy.js). It initializes the candidates list and the voting period at the same time.
- The deployment of the SC is done on the network using the owner's private key to the wallet and will return a contract address (which will be used as a transaction destination) on which the application will run.
- The app uses the Metamask, which is an e-wallet and has a browser extension for Google Chrome to get the current user's account address.
- Once the user is connected and the app has its account address and the contract address the ethers.js library is used to interact with the SC.

App utilization demo:

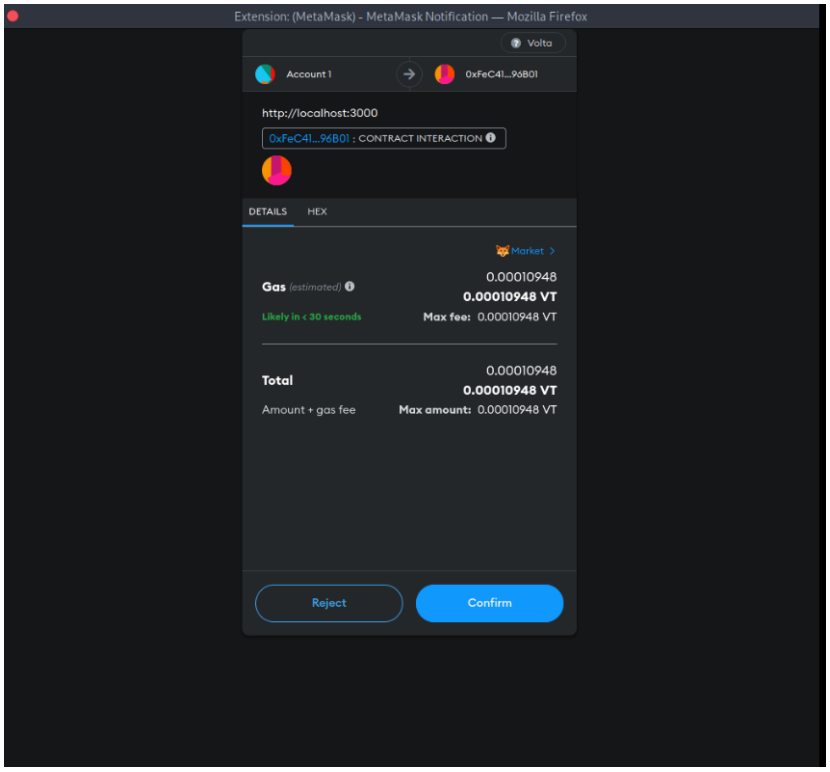
- To run the app a Metamask account is needed.
    - https://metamask.io/download/
- The app is built using the Volta network, so the testnet needs to be added in Metamask.
- To add Volta to Metamask, one of these two methods can be used:
    - https://csb-qo27t.netlify.app/
    - https://energy-web-foundation.gitbook.io/energy-web/ew-dos-technology-components-2023/trust-layer-energy-web-chain/ewc-guides-and-tutorials/connect-to-energy-web-chain-main-network-with-metamash
- In order to interact with the smart contract, a small fee needs to be paid; to get testnet VT coins, the following link can be used:
    - https://voltafaucet.energyweb.org/
- After the setup is done, the login process is simple:



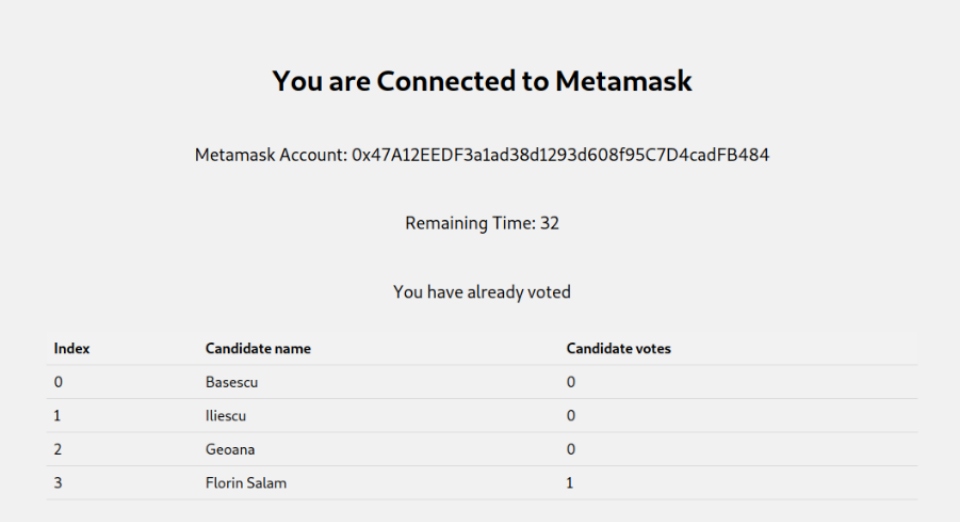
- After the login the current voting session is presented:



- A user can only vote once from an account, using its address; the blockchain transaction takes a few seconds to complete.
- Voting means interacting with the contract - a transaction is made and the fee is paid from the wallet:



- After the transaction is processed the voting result is shown:



Implementation drawbacks:

- The application guarantees that a wallet votes a single time, however it does not guarantee that a user votes only once, as everyone can create a large number of wallets and vote multiple times with different wallets.
- Delays can be caused by the blockchain block time. For Volta network, the average time for a transaction to be processed is aprox. 7.5 seconds.

4. Conclusion

In conclusion, our blockchain-based voting app represents a substantial advancement in electoral technology, however it is essential to acknowledge certain challenges. While transactions are public and can be viewed by anyone on the blockchain, they are still anonymous transaction in regard to the user's real identity. Our application can ensure that each wallet can only vote once, but because the application uses Metamask, which allows wallet creation with ease, an individual can create multiple wallets, therefore casting multiple votes. Using a mainnet, instead of a testnet and another, more secure, type of wallet should limit this issue.

Although the app prioritizes wallet addresses to maintain anonymity and to ensure one-time voting, future iterations may explore enhanced identity verification mechanisms to mitigate the risk of multiple voting and ensure maximum integrity of the voting process.