

Decentralized Voting System Using Blockchain Technology

¹Sayali Karande, ²Ajay Matkar, ³Shruti Mirajkar, ⁴Pragati Pejlekar

^{1,2,3}Students ⁴Assistant Professor,
Department of Information Technology,
Saraswati college of Engineering, Mumbai

Abstract: Voting for anything is one of the most important ways to ensure fair representation and equal voice when making decisions. The larger the implications for each decision, the more people who participate in the process. As such, it can become difficult to accurately and efficiently keep track of each voter's eligibility and validity to participate. On top of this, there are also other issues including lack of transparency and potential corruption which discourages many from voting at all. In this project, we have designed an electronic voting system using Ethereum blockchain technology. The whole voting process would be decentralized, which means that there is no central agency. Anybody can participate and become a node in the system as long as they meet requirements. Smart contract will be designed in such a way that double voting problem is solved.

Keywords: blockchain, ethereum, decentralized, smart contract.

I. INTRODUCTION

Blockchain in simple terms is defined as a chain of blocks, that are cryptographically secured and joined together. A blockchain acts as a Distributed ledger which is open to anyone. Blockchain also has property, that once the data is recorded inside the blocks of blockchain it becomes difficult to change it. The blockchain is used for the secure transfer of items like money, property, contracts, etc. without requiring a third-party intermediary like bank or government.

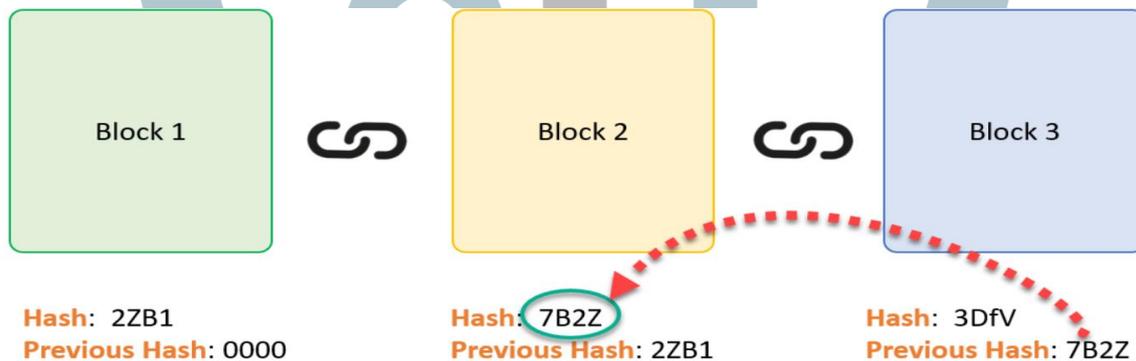


Fig 1.1 Blocks in blockchain.

Each block in blockchain contains:

- 1.Data.
- 2.Hash.
- 3.Previous block hash.

Centralized vs. Decentralized

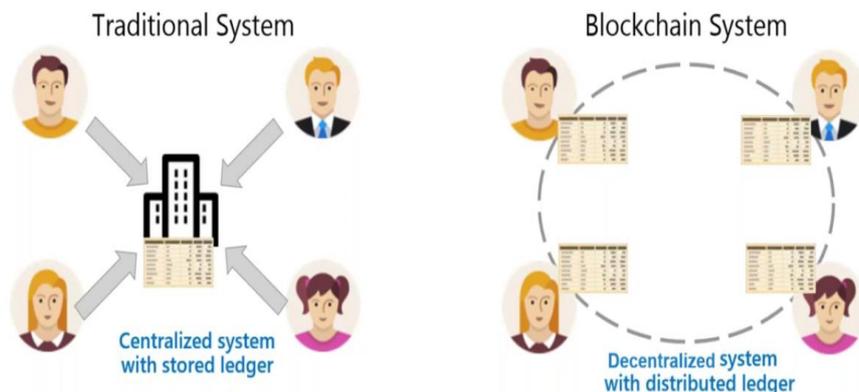


Fig 1.2: Centralized ledger v/s Decentralized ledger

As shown in the fig 1.2 we can see that in centralized system a central agency is involved. In traditional system the votes that are given by the people goes to the central agency and then the results are displayed by this central agency to everyone. Whereas in the blockchain system there is no central agency all the people involved in the process of election are connected to each other through a network so whatever changes happens in one person database that will be reflected to all others that are connected in the network. The problem of transparency is solved.

II. HISTORY AND EVOLUTION

Blockchain started back in 1990's. The first mention of the blockchain starts from 1991, when Stuart Haber and W. Scott Stornetta did the first work on the blockchain. In 1992 Blockchain like technology improved by incorporating the Merkle Trees. Stornetta did the first work on the secure chain of the blocks, next year in 1992 blockchain saw the introduction of the Merkle trees in the blockchain ledger design which enabled the multiple documents to be stored within a single blockchain, increasing the blockchain efficiency. In year 2008 Satoshi Nakamoto published a white paper titled "Bitcoin - A Peer to Peer Electronic Cash System". This paper concluded to solve the problem of double spending of money, which was a major problem in the digital currency. In 2015 Vitalik Buterin launched a blockchain named "Ethereum". Ethereum blockchain was much better than the Bitcoin blockchain, it has a major added feature called smart contracts. Which ran on the Ethereum network and were Turing complete, but both the blockchains "Bitcoin and Ethereum" had same consensus also called Proof of Work, but this required the heavy investment of the mining rigs to run and maintain the network, which can calculate the computational hashes and miners are given an incentive, which are called "Transaction fee" which goes to the miners.

III. REQUIREMENTS

- 1) Metamask – To use Ethereum blockchain we should have a Metamask chrome extension and account created in Metamask.
 - 2) Visual Studio Code - The code for frontend and backend is written in Visual Studio Code. Frontend is written in JavaScript. Backend is written in Solidity language to design smart contract.
- Solidity is an object-oriented programming language designed for the Ethereum network. Solidity helps to write the smart contract, which can make state changes via making a transaction on the Ethereum network. Solidity is a contract programming language that has similarities to JavaScript and C. The code that is written in Solidity is compiled in EVM (Ethereum Virtual Machine).
- 3) Dependencies like:
 1. Node.js
 2. Truffle framework- It gives us a set of tools to write the smart contract with the Solidity programming language and deploy smart contract. It also gives a framework for testing smart contracts.
 3. Ganache- It is a virtual blockchain which sets up 10 default Ethereum addresses with private keys and 100 ethers for each address.

IV. PROPOSED SYSTEM

Step 1: Firstly voter has to login in their metamask account and the account should contain ethers for voting purpose.

Step 2: Open the voting webpage which has table of candidate. Each candidate has an id, name and vote count.

Step 3: Voter should select the candidate whom he/his wants to vote.

Step 4: Voter should click on vote button.

Step 5: Voter will then get a metamask notification. In that notification voter can see how much gas fee is used for the vote caste and two options will be there reject or confirm. The voter has to click on confirm button if he/she wishes to vote.

Step 6: Voter will get the notification of confirmed transaction.

Step 7: Finally the voter can see that which candidate has got how many votes.

As we say that this is decentralized system so all the transaction are visible to each voter and this can be seen in etherscan.io.

V. RESULTS

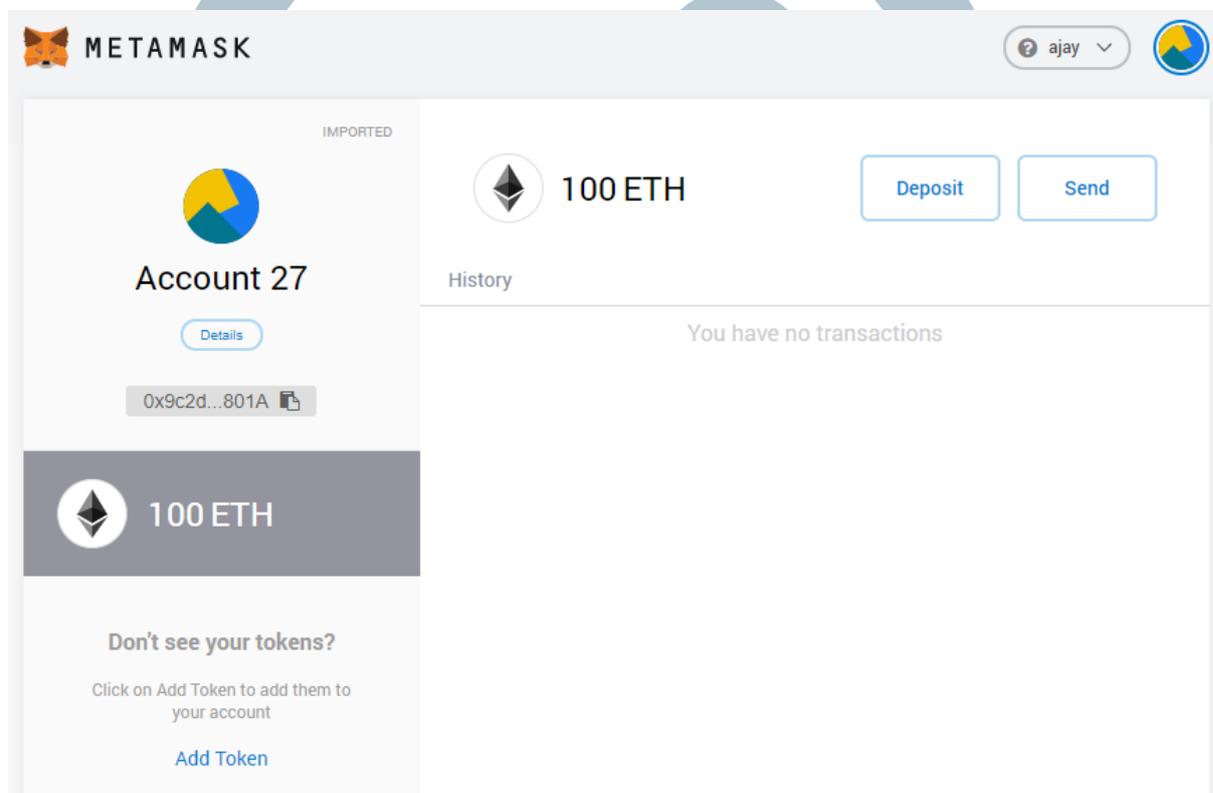


Fig 5.1: Metamask account.

Election Results

#	Name	Votes
1	Candidate 1	0
2	Candidate 2	0

Select Candidate

Candidate 1

Vote

Your Account: 0x852e861d11efcf0942baa32e856c86cdf9ac628b

Fig 5.2: Client side Application.

This is election result page over here we have table of candidate. Each candidate has an id, name and vote count.

The screenshot shows a browser window with two tabs: 'Election Results' and 'MetaMask'. The 'Election Results' page displays a table with two candidates, each with 0 votes. Below the table is a 'Select Candidate' dropdown menu with 'Candidate 2' selected, and a 'Vote' button. The account address is 0x9c2d9c215d33e525a32eb57e6c33bd17cae0801a. A Metamask notification overlay is open, showing a transaction for 'Account 27' with a gas fee of 0.000096 ETH and a total amount of 0.000096 ETH. The notification includes 'Reject' and 'Confirm' buttons.

Fig 5.3: Metamask notification.

After selecting the candidate the voter has to confirm the transaction.

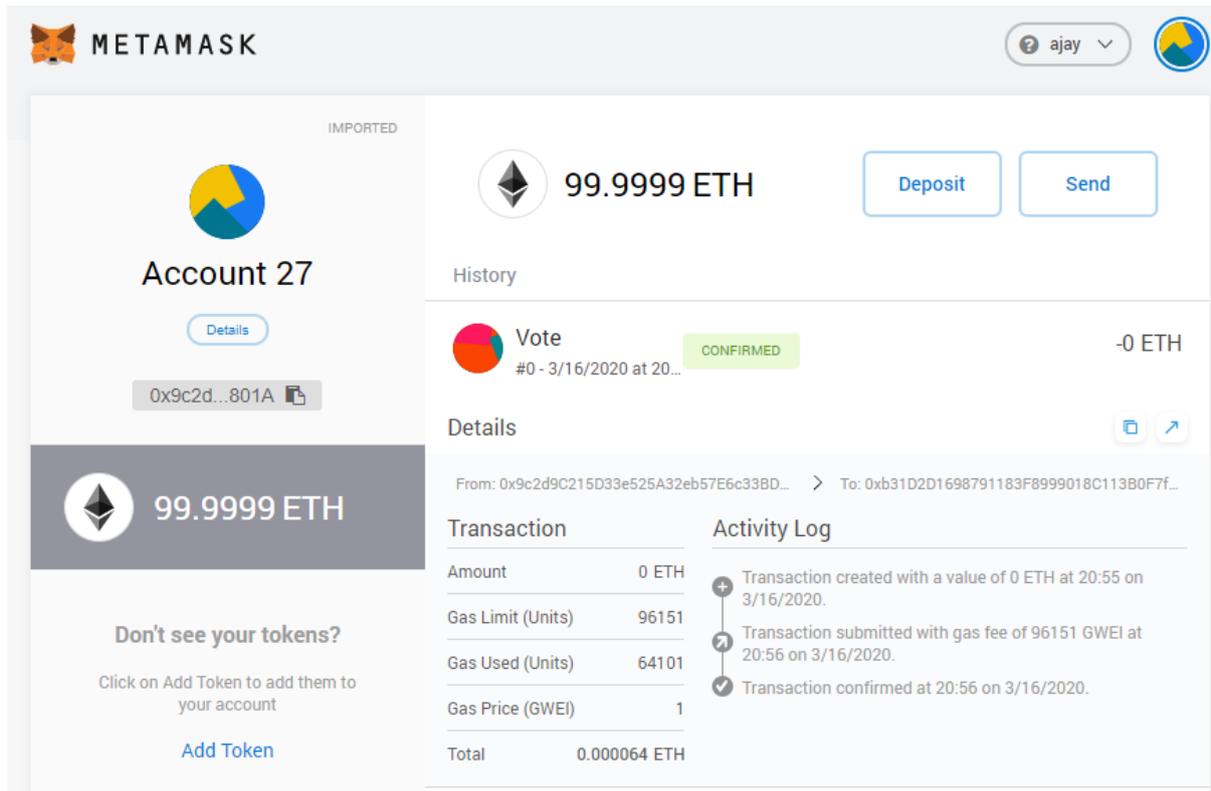


Fig 5.4: Transaction Details.

Here we can see that from whom to whom the vote is cast and at which time all the details are mentioned and how much gas fee is used is also mentioned.

Election Results

#	Name	Votes
1	Candidate 1	2
2	Candidate 2	3

Your Account: 0x1600ec0b8945c6e4f305dec3161f1189707ba632

Fig 5.5: Election Result

VI. CONCLUSION

Today also we have traditional system for voting, the people have to go to the booth and then cast vote and after few months the result is displayed. So E-voting is still controversial within political and scientific circles. Blockchain based voting systems provide full transparency by eliminating the need for central authorities to be involved. The problem of double voting is also solved as we have designed smart contracts in such a way that voters can vote only once. Results are also displayed immediately after the time period of voting is over. Technology can potentially be an answer to the question of how to keep elections and voting as secure and foolproof as possible.

REFERENCES

[1] BroncoVote: Secure Voting System using Ethereum's Blockchain:

Gaby G. Dagher, Praneeth Babu Marella, Matea Milojkovic and Jordan Mohler.

[2] Digital Voting with the use of Blockchain Technology;

Andrew Barnes, Christopher Brake and Thomas Perry.

[3] Towards Secure E-Voting Using Ethereum Blockchain:

Gökhan Dalkılıç Doku, Ali Kaan Koç, Umut Can Çabuk, Emre Yavuz.

[4] Blockchain-based Smart Contracts: A Systematic Mapping Study of Academic Research:

Maher Alharby, Amjad Aldweesh, Aad van Moorse.

[5] <https://solidity.readthedocs.io/en/v0.5.11/solidity-by-example.html#voting>.

[6] <https://www.youtube.com/channel/UCY0xL8V6NzzFcwzHCgB8orQ/playlists>.

[7] <https://www.ibm.com/blogs/blockchain/2018/07/what-are-smart-contracts-on-blockchain/>

