TEST Engineering & Management

May-June 2020
ISSN: 0193-4120 Page No. 5321-5324

Blockchain Based E-Voting System Using Ethereum

Harshiniy U[1], Himarshini Ganga[2], Janhavi M P[3], Gopinath R[4]
[1,2,3]UG Students, B-TECH, [4]Assistant Professor, Department of C and IT,
REVA University, Bangalore, India
[1]uharshiniy@gmail.com, [2]ppganga12@gmail.com, [3]janvi3009@gmail.com,
[4]gopinath.r@reva.edu.in

Article Info
Volume 83
Page Number: 5321-5324
Publication Issue:
May-June 2020

Article History
Article Received:*19 November 2019*
Revised: *27 January 2020*
Accepted: *24 February 2020*
Publication: *16 May 2020*

Abstract

To build an electronic voting system that must satisfy the requirements of legal principles has been very demanding for a long period of time. Advancements in the IT world has paved new ways for distributed ledger technologies to come into existence to improvise the legality of the existing systems. But limitations still prevail. Through this paper, we realize blockchain technology as a service to distributed ledger technologies. The three pillars of blockchain technology are transparency, immutability and decentralization. Now, a voting application should exhibit transparency, fairness, confidentiality and security against malicious interventions which is all possible with the help of blockchain technology. Since a public ledger is created every time a transaction is done, there is complete transparency among all the users of the blockchain and since it is decentralized, the election process is fair; as there is no central authority accountable for the entire election. Confidentiality is maintained by assigning one account per person with their own private key and our smart contracts will ensure that once a vote is casted, it cannot be tempered due to the nature of the blockchain. The smart contracts will also be responsible for allocating one vote per account to avoid duplicate votes. The elections will be time sensitive and once the election ends the statistics will be displayed to the voters.

Keywords: *Ethereum, Ganache, Metamask, Smart Contracts, Truffle*

1. Introduction

Election plays a very vital role in our democracy. It gives every citizen of that democracy, the right to elect their leader and country's representatives. The current voting system's groundwork or under structure is not in its right form. It is either subject to frauds or bribery or manipulation of the voter's databases. The voting machines and databases are completely centralized, which means that it is completely under the control of a single authority. This might lead to potential tampering of election databases collected, as it is possible to penetrate or manipulate just a single system. To overcome this circumstance among various others and to empower the legitimacy of the election results, blockchain technology can be used. Blockchain technology lays out distributed, centralized and transparent architecture.

This architecture of blockchain comprises a peer-to-peer network of computers called nodes. These nodes distribute and fragment all the data and the code involved in that particular network. So, if your device is allied to the blockchain, your device represents a node in the network and you can communicate with all the computer nodes in the network as your computer has a legitimate copy of code and data on the blockchain. This is our distributed ledger framework that allows the data and the code of the network to be a public ledger of the blockchain. There are no central servers present. Just a group of computer nodes that communicate to one another on the same network. Blockchain with its exclusive distributed and decentralized technology resolves many issues in the current digital era and it can easily aid the process implementation of an e-voting platform. Which will be free from frauds and threats and yet keep the voter's identity and vote intact and unviolated. The voting process starts with the user creating an account with a valid address with some ether (Ethereum cryptocurrency). Once an account has been created and authenticated, they can cast a vote and pay a small transaction fee, this fee is called gas.

2. Literature Survey

Shubham Pareek, et al. [1]. Implemented an e-voting process on ethereum as the development platform. The main purpose of their paper is to implement an ideal platform which being the ethereum blocks to be made in real time and these blocks are validated by the minors, and then these minors solve complex algorithmic problems which generate nonce values, which in turn makes a link with the previous blocks and as the result, all the blocks are connected to form a blockchain. All the data and code are stored in this blockchain using cryptography. Drawbacks of their paper are the lack of authentication of voters, integrity, verification and non-repudiation of votes.

Fridrik P Hjalmarsson, et al. [2]. Proposed a method to send hundreds of transactions per second onto the blockchain, utilising every aspect of the smart contract to ease a load on the blockchain. This was carried out using an ethereum private blockchain (ganache). The main advantage of this election scheme is that it allows individual voters to vote at a voting district while guaranteeing that every individual voter's votes is counted from the correct district which potentially increases the voter turnout. Demerits of this paper is the lack of correct corroboration of voters and legal issues related to the account owners.

Jorge Lopes, et al. [3]. Implemented this e-voting system on a html interface for the application users. A cryptographic server was used to encrypt and decrypt the votes and three smart contracts that were deployed on the ethereum blockchain were written in solidity language and the application program interface was picked to act as a bridge for all the components. After every voter votes, the application programming interface sends an eth call with the hashed candidates to retrieve the current encrypted word count. Then later, the API decrypts all the votes and displays them. This method leads to the drawback with the inability to prevent the possibility of fraud or incorrect data being delivered into the contract, i.e. the process of registration must ensure that the voter's details belong to the person who is registering. Secondly, the voter has to possess a certain amount of ether in order to pay for the transaction gas.

3. Body

3.1 Proposed System

With an intention to make e-voting more open, transparent and independently auditable, we have used the blockchain technology on the ethereum platform using solidity programming language. Blockchain creates an immutable ledger, every time a transaction takes place. Once a vote is casted, it cannot be tampered. To make this into a decentralized application, we have written smart contracts in Solidity specifying the various conditions of the election. Our smart contracts are responsible for communicating with the local blockchain and running the election. After successfully running our smart contract, we were able to create an application for conducting elections with the help of ganache, which is our local blockchain. It is user-friendly and an important advantage is that a voter can cast their vote from any location, thus increasing the overall participation. One account on blockchain is associated with only one voter, who can cast only once, thus ensuring that there are no duplicate votes. The election results are displayed after the voter has finished voting, which guarantees complete confidentiality.

3.2 Dependencies Involved

- **Solidity** is an object-oriented programming language for writing smart contracts. It is used for implementing smart contracts on various blockchain platforms but majorly ethereum.
- **Ethereum** is an open source, blockchain based, distributed computing platform. It is also an operating system featuring smart contract functionality. Ether is the cryptocurrency which is generated by the ethereum platform. Ethereum is designed to be a general purpose programmable blockchain.
- **Truffle Framework** Truffle is a development environment testing framework and an important asset timeline for ethereum. Truffle boxes are helpful boiler plates that allow you to focus on what makes a decentralized application unique. They also consist of other useful modules such as solidity libraries and front-end views, which help to complete the decentralized application. We have used the Truffle pet shop box in particular.

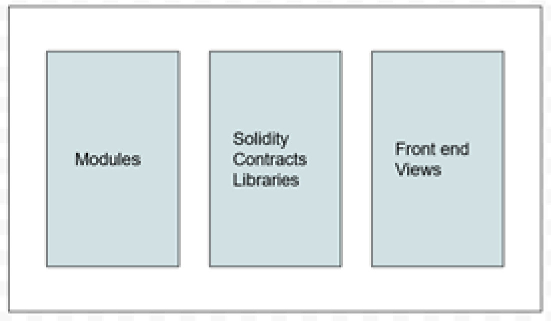

Figure 3.2.1: Representation of Truffle Framework

- **Ganache** is a personal ethereum blockchain which mimics the characteristics of the main blockchain. It can be used to run tests, execute commands and inspect state, while controlling how the chain operates.
- **Metamask** gives the liberty to run ethereum d-Apps, right in your browser without running a full ethereum node. A web browser can be converted into a blockchain browser by installing Metamask extension. It aids in managing our personal account that we need to pay for transactions.

- **Web3.js** is a JavaScript library and the front-end module for our proposed system. It helps the clients to interact with the ethereum blockchain by performing actions like sending ether from one account to another, read and write data from smart contracts and also create new smart contracts.

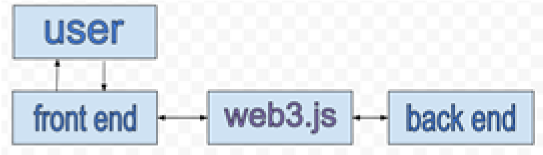

Fig 3.2.2 Overview of Web3.js

- **Smart Contracts** are the building blocks used to create the blockchain. They are programs that we can write with source code and deploy to the blockchain. They are written in Solidity programming language. Smart contract becomes the public ledge agreed upon the parties involved.

4. Working

4.1 Smoke Test

It is used to check if our local blockchain or ganache is running after we write a basic piece of our smart contract. To deploy a smart contract, we create a new migration file. In order to get the desired output, we need to make sure ganache is running in the background. Migration of contracts require ether (cryptocurrency) and hence ether from our account in ganache is reduced after migration.

4.2 Writing Our Smart Contract

4.2.1 Model a Candidate

This includes defining a smart contract named an election and defining the structure of candidate with three attributes; ID, name and votecount.

4.2.2 Basic Voting Conditions

- To check if an account has already voted or not, we create a mapping in solidity and assign to a boolean value.
- To store the candidate count, we create a public mapping and store in a variable called candidateCount()
- An event is created with the help of votedEvent() which triggers a block of code that registers that particular vote.

4.2.3 Candidate Registration

In order to add our candidates, we create a constructor that is public and call the addCandidate() function. This function is private and is not accessible to everyone.

4.2.4 Specific Voting Conditions

1. First condition is that the voters have not voted before. In case an account has already voted, the transaction will fail.
2. If the candidate count is more than the registered candidates, it will throw an error.
3. To record that a voter has voted, we set msg.sender to true; this helps to track the votes and avoid any duplicate votes.
4. To update the candidate vote count, we increment it every time by calling the vote count condition.
5. We trigger the event by calling the votedEvent() function.

4.2.5 Client-side Application

We have built the client-side application by writing a simple html code in the index.html file of the pet shop box. The voters are provided with options and they have to use the drop-down menu to select the candidate and cast their vote.

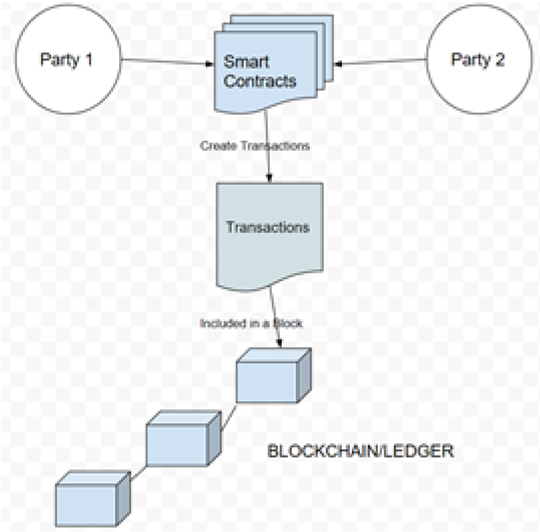

Figure 4.2 Illustration of client side application using smart contracts between two parties

5. Evaluating Blockchain as a Service

After the voters are provided with the drop-down menu to pick the candidate he wishes to vote for, he/she simply has to pick the vote button. This will trigger an event in our smart contract and will lead to an authentication page on the metamask window. This window checks if the account has voted before or not. If he/she has not voted, it will display the statistics of the voters so far, otherwise the transaction will fail.

6. Conclusion

We have proposed a unique E-voting system to ensure transparency, integrity, fairness, which realises a completely decentralized system. A voter is now able to cast a vote using one account using a private key. The vote count is incremented using our smart contract. Smart contracts make it possible to check if the voter has already voted or not, thus preventing the voter from voting multiple times. Since the public ledger on Metamask is time stamped, it cannot be changed. If the voter tries to vote twice, the metamask transaction will fail. This is ground breaking since conventional voting systems do not possess such airtight method for checking duplicate votes. The word count cannot be manipulated as they are part of our smart contract and once deployed, the smart contract cannot be edited. Our proposed system is a theoretical model and can be used as a basis for conducting elections online with the help of blockchain.

References

[1] E-voting using Ethereum Blockchain- this was published by department of computer science at SRM Institute of Science and Technology, Chennai. Year of publication- 2018. Authors- Shuban Pareek, Anuj Upadhyay, Satya Doulani, Siddharth Tyagi, Adtiya Verma. http://www.ijrti.org/papers/IJRTI1811006.pdf

[2] Blockchain based e-voting system- this was published by Friðrik Þ. Hjálmarsson, Gunnlaugur K. Hreiðarsson at Reykjavik University, Iceland. Year of publication-2017. https://skemman.is/bitstream/1946/31161/1/Research-Paper-BBEVS.pdf

[3] Blockchain based e-voting system: A proposal- this was published by Jorge Lopes, José Luís Pereira and João Varajão at Universidade do Minho, Portugal. Year of publication- 2019. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1296&context=amcis2019

[4] Secure digital voting system based on blockchain technology- this was published by Kashif Mehboob Khan, Junaid Arshad and Muhammad Mubashir Khan at NED University of Engineering and Technology, Pakistan. Year of publication-2017. https://core.ac.uk/download/pdf/155779036.pdf