Decentralized Voting Application Using Ethereum Blockchain

Gananjay Thanekar, Aveesh Shetty, Sagar Sethi, Sulochana Madachane, Hasib Shaikh
Student, Student, Student, Professor (Guide), Professor (Guide)
Computer Engineering,
K.C College of Engineering & Management Studies & Research (Mumbai University), Thane, India

Abstract: Decentralized voting app using Ethereum smart contract is aimed for being secure system. As the votes cannot be tampered or manipulated by anyone. Once the voter votes for the candidate the vote count is increased of that candidate and so that count will be maintained using public ledger. We use Ethereum Smart Contract from public blockchain system. The system will be highly secured as the voter will get logged into the system using his Private Key which is most secure way of voting we are trying to give to the voters. Distributed across device connected to the block chain. Blockchain is a peer-to-peer connection of nodes that talk to each other. If you have a device and it's connected to the blockchain network then you are a node and you talk to all the other nodes.

Index Terms **- Blockchain, Ethereum, Smart Contract, Voting Applications, Solidity.**

I. INTRODUCTION

Block chain is a peer-to-peer connection of nodes that talk to each other. On the block chain the data doesn't reside in a single server. It is distributed across every node. All of this data is contained in bundles of records called blocks which are chained together to create a pubic ledger and all of the nodes work together to ensure that data in the publication because it means that our account sent the transaction when we vote and our vote will go to the candidate and be recorded forever. Because all of the data is shared across devices in the block chain, it is database and as each node talk to each other it is a network so instead of traditional web model you can think of block chain as database and network all in one. The Ethereum allows us write code that get executed on the ethereum virtual machine with smart contracts .This is where our business logic of our application will lie. This is the code that'll responsible for reading and writing the data transfer the value and executing any business logic that we program.

II. LITERATURE SURVEY

During the development of this project, we read through papers. [1] A user-friendly voting and quiz system for classroom use with connected devices, 2016 IEEE International Conference on Consumer Electronics-China (ICCE-China) in classroom, teachers often ask questions to evaluate the students' understanding on a topic. An easy-to-use interactive response system with a companion mobile app will be useful for improving the teaching and learning experience. There are some voting systems in the market. However, most of them are not convenient to use and configure, making students and teachers reluctant to use.

The system proposed in this paper includes three major functions - voting/quiz, countdown, and real-time statistics. The voting/quiz function provides a simple interactive approach that can be used in any subject. The system is designed with simplicity in mind. Teachers will find it easy to create a voting or question, and students will find it easy to do a voting or answer a quiz. The students can login question or voting by scanning the corresponding Quick Response (QR) code or entering the Randomly Generated Code, without a complicated login system for the students.

Smart Contract:

Ethereum allows developers to program their own smart contracts. [2] Smart contracts can viz. Function as 'multi-signature' accounts, so that funds are spent only when a required percentage of people agree and manage agreements between users, say, if one buys insurance from the other and provide utility to other contracts (similar to how a software library works) and Store information about an application, such as domain registration information or membership records. A smart contract is a computer code running on top of a block chain containing a set of rules under which the parties to that smart contract agree to interact with each other. If and when the pre-defined rules are met, the agreement is automatically enforced.

System Interaction:

When tradition web application is used, a browser talk to central server over a network and all of the code to web application lives in a server and all of the data is stored in a centralized database. Anytime application is used it must talk to the central server and this is how web application works it's really bad to build a voting app this way, because

1) The data in the database could any time our vote could change or could be deleted entirely.
2) The code of our application could change and means that our rules in the election could change. Votes of the entire voters should be counted only once and correct candidates with most number of votes should win.

A. User (Voter)

User is the one who interacts with the application to read or write the data .In our case, Voter is able to increase the vote count of a particular candidate and would be able to vote only once. The user is able to change the state of the Ethereum block chain by reading and writing data to the public ledger. Only connected users can interacted with smart contract.

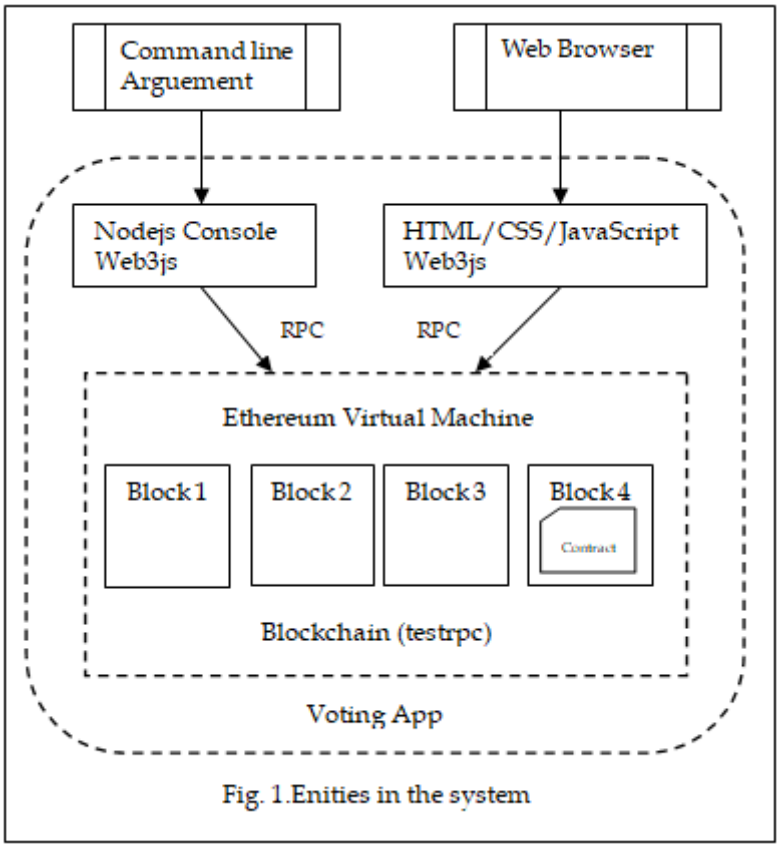


Fig. 1.Enities in the system

B. Candidate

Candidate is associated with the state in the public ledger that changes when voter votes. Candidate vote count could not be tampered and correct candidate with most number of votes would win. Present votes of all the present candidate is shown to the user voting.

Metamask

Metamask is chrome extension that we need in order connect to ethereuem blockchain as it is a network. We will be able to connect to Ethereum network with our personal account and interact with smart contract using metamask.

Metamask is an Ethereum wallet in your browser

MetaMask is an extension in your browser for accessing distributed applications or "Dapps" which have Ethereum enabeled. When a user loads a webpage, MetaMask automatically injects an Ethereum provider and a Web3 API (interface) instance for the webpage to use. This allows Dapps to access the blockchain and read their users' account.

MetaMask also lets the user to create and manage their own identities (via private keys, local client wallet and hardware wallets like Trezor™), so when a Dapp wants to perform a transaction and write to the blockchain, the user gets a secure interface to review the transaction, before approving or rejecting it.

Since this extension adds functionality to the normal browser context, MetaMask requires the permission to read and write to any webpage.

A. Using metamask with Ganache

Ganache is a graphical application that runs a blockchain that can be used for testing purposes. It runs on 127.0.0.7545 Note: We recommend specifying 127.0.0.1 instead of local host because the address does not require a network connection which makes it more suitable for development.

B. MetaMask network menu

In the box titled "New RPC URL" (to the right of "New Network") enter http://127.0.0.1:7545 and click Save.

The network name at the top will display "Private Network". Click the cross in the top-right of the current window to close the page and return to the Accounts page.

Now that we've connected MetaMask to Ganache, you'll be taken to the accounts screen. Each account created by Ganache is given 100 ether. The first account should have less than the others because this account supplies the gas for smart contract deployment. Since you've deployed your smart contract to the network, this account paid for it.

Click the account icon in the upper-right to create new accounts, the first 10 of these accounts will correspond to the 10 accounts displayed when you launched Ganache.

Ganache: Ganache is a personal block chain for Ethereum development which can be used to deploy contracts, develop your applications, and run tests. [3] It is available as both a desktop application as well as a command-line tool (formerly known as the TestRPC). Ganache is available for Windows, Mac, and Linux.

A. Main interface

When you launch Ganache, the screen will show some details about the server, and also list out a number of accounts. Each account is given 100 ether**.** Having ether automatically in all accounts allows you to focus on developing your application.

B. Ganache Accounts

There are four pages available:

- The Accounts page shows the accounts generated and their balances. This is the default view.
- The Blocks page shows each block as mined on the blockchain, along with gas used and transactions.
- The Transactions page lists all transactions that run against the blockchain.
- The Logs page shows the logs for the server, which is useful for debugging.

C. Settings

You can change some features of the generated block chain through the **Settings** pages, accessed by the gear icon in the top right corner.

- The Server page shows details about the network connection, including hostname, port, network ID, and also if each transaction is to be mined into block automatically or not.
- The Accounts & Keys page sets details about the number of accounts created, and whether to use a specific mnemonic or let Ganache generate its own.
- The Chain page sets details about the actual working of the generated blockchain, including gas limit and gas price.
- The Advanced page toggles Google Analytics, which is useful for the Ganache team to track and analyse usage of the application.

After making changes, you will have to click Restart on the application for the changes to take effect.

Truffle:

Install tuffle which will allows us to write decentralized app on Ethereum network and also provide a set of a tools for writing smart contracts on blockchain and deploying it. [4] It also provides a framework for testing our application. We can also develop a client side app in tuffle.

Web3js:

Web3.js is a collection of libraries which allows you to interact with a local or remote ethereum node, using a HTTP or IPC connection. You just need to get web3.js into your project.

Vuejs:

Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed to be incrementally adoptable. The main library focuses on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of supporting Single-Page Applications when used in combination with modern tooling and supporting libraries.

Nodejs:

Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world.

NPM:

Node Package Manager (NPM) provides two main functionalities: Online repositories for node.js packages/modules which are searchable on search.nodejs.org [5] and Command line utility to install Node.js packages, do version management and dependency management of Node.js packages.

Chai:
Chai is a Behavior-driven development / Test-driven development assertion library for node and the browser that can be easily paired with any javascript testing framework.

Mocha:
Mocha is a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing *simple* and *fun*. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases.

Solidity:
Solidity is an object-oriented, high-level language for implementing smart contracts. [6] Smart contracts are programs which govern the behavior of accounts within the Ethereum state. Solidity was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features. With Solidity you can create contracts for users such as voting, crowd funding, blind auctions, and mutli-signature wallets.

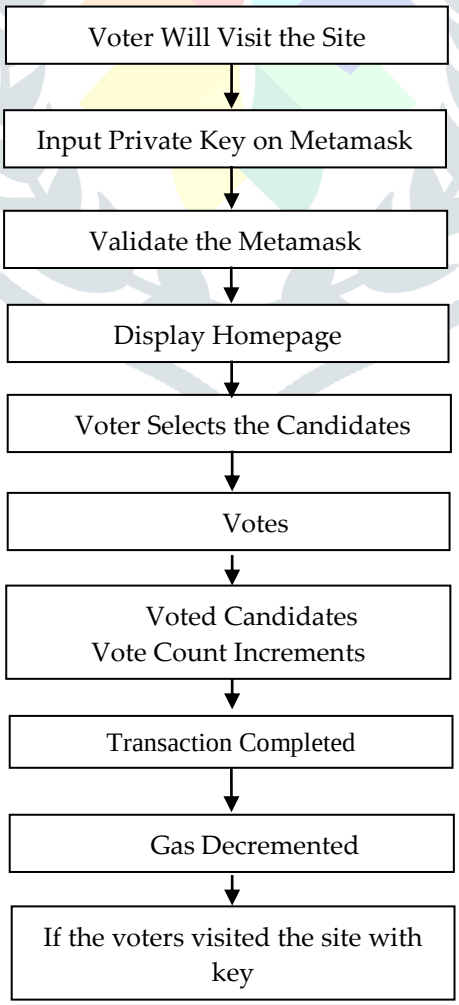
III. METHODOLOGY

The Ethereum allows us write code that get executed on the ethereum virtual machine with smart contracts .This is where our business logic of our application will lie. This is the code that'll responsible for reading and writing the data transfer the value and executing any business logic that we program. We can think of smart contract as micro-service that lives on the web. If public ledger represents data layer, then smart contract represents the layer reads and writes the data, does stuff with it and talk to other services. Also it's called a smart contract because it represents some kind of covenant or agreement.

A voting system which is unhackable or which cannot be tampered. Using blockchain based distributed networks to store and record votes which allows voting to be almost 99.9% secure as it is distributed among a cluster of networks. The system has a user friendly interface and can be used to run polls and elections with total security and zero percent chance of manipulation**.**

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. Compact and offline system.

Your votes will be counted only once, and they cannot be changed / tempered by anyone. The votes will be given to the right candidates without tempering.



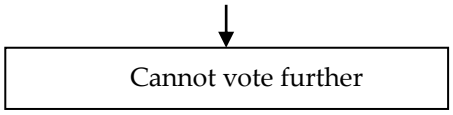


Fig. 2.Graphical Illustration of Methodology

IV. CONCLUSION

In a traditional web application, a hacker can hack into code that is residing in the central server or the database. Voting has been always offline due to the fact that making the voting process online can result in tampering of votes leading to victory of wrong candidate. DAPP stands for Decentralized web application solves the problem of a traditional web application. Blockchain is a peer-to-peer connection of nodes that talk to each other. If you have a device and it's connected to the blockchain network then you are a node and you talk to all the other nodes. Ethereum blockchain technology makes the web application secure and decentralized by applying the concept of cryptography and decentralization. Each user has to connect to Ethereum blockchain so that it can be a node of the ethereum network and can contribute to keeping the public ledger and contribute in voting process. The voting count that is stored on the public ledger available to all the nodes but is encrypted so no one can edit it neither the rules that are defined in the code or the data in public ledger that act as the database. Thus we conclude that we believe that the traditional web application should be replaced by decentralized web application so that developer of the website shouldn't have all of the user's data They may use them to manipulate the user for their own benefits. The rules defined in the application are applicable to all users whether it is normal user or admin.

V. ACKNOWLEDGEMENT

We would like to thank our guide Prof. (Mrs.) Sulochana Madachane, Associate Professor, K. C College of Engineering & Management studies & Research, Thane and Prof. (Mr.) Hasib Shaikh, Associate Professor, K. C College of Engineering & Management studies & Research, Thane, for initiating us into this field of research and for providing me with the necessary guidance, great encouragement throughout the preparation of this paper. We record our deep indebtedness to them for their support. We take this opportunity to express our gratitude to the Staff members, Non-Teaching Staff members and Research Scholars of the Department of Computer Engineering, K. C College of Engineering & Management studies & Research, Thane, for their timely help and encouragement.

REFERENCES

[1] A user-friendly voting and quiz system for classroom use with connected devices (19-21 Dec.2016 IEEE International Conference on Consumer Electronics-China (ICCE-China))
[2] https://github.com/ConsenSys/smart-contract-best-practices
[3] https://github.com/trufflesuite/ganache-cli
[4] http://truffleframework.com
[5] https://search.nodejs.org
[6] https://github.com/ethereum/solidity