Topic: Voting DApp

Members: Siddharth Gupta: 2020A3PS0478H
Vikhyat Singh Gaur: 2020AAPS1765H
Charith Goud: 2020A1PS2412H
Siddhesh Bahety: 2020A7PS1686H

Electronic voting or e-voting has fundamental benefits over paper based systems such as increased efficiency and reduced errors. The electronic voting system tends to maximize user participation, by allowing them to vote from anywhere and from any device that has an internet connection. The blockchain is an emerging, decentralized, and distributed technology with strong cryptographic foundations that promises to improve different aspects of many industries.

Expanding e-voting into blockchain
technology could be the solution to alleviate the present concerns in e-voting. Here we propose a blockchain-based voting system that will limit the voting fraud and make the voting process simple, secure and efficient.

Blockchain can help to implement a system that is immutable, transparent, and efficient and cannot be hacked into. The inability to change or delete information from blocks makes the blockchain the most effective technology for voting systems. Blockchain technology is supported by a distributed network consisting of variety of interconnected nodes. Each of these nodes have their own copy of the distributed ledger
(information) that contains the total history of all transactions the network has processed. There is no centralized system that controls the network. If the majority of the nodes agree, then they accept a transaction. This network permits users to stay anonymous. A basic analysis of the blockchain technology (including sensible contracts) suggests that it is an appropriate basis for e-voting and furthermore, it might have the potential to form e-voting a lot of acceptable and reliable.

For our proposed plan of work we are considering two modules :
1. Front-end for the application
2. Back-end using Solidity to implement Blockchain.

In the project, we'll get two candidates contesting for the election.

Every account or user will have the option of selecting and contesting their vote. Some ether will be used because of this, and the vote will be deployed in the blockchain.

Ethereum

• For developing E-voting using Blockchain we used Ethereum - a popular platform for creating distributed Blockchain applications that support smart contracts. Ether (ETH) is the native cryptocurrency of the platform.

Smart Contracts

• Smart contracts are self-executing contracts which contain the terms and conditions of agreement between peers.

They are simply programs stored on a blockchain that run when predetermined conditions
are met.

They typically are used to automate the execution of an agreement so that all participants
can be immediately certain of the outcome, without any intermediary's involvement or time loss.

. Smart contracts eradicate the need for a third-party intermediary of facilitator, essentially
giving you full control of the agreement.

Solidity

Solidity is a contract-oriented, high-level language for implementing smart contracts. It is statically typed, supports inheritance, libraries and complex user-defined types among other features.

Metamask

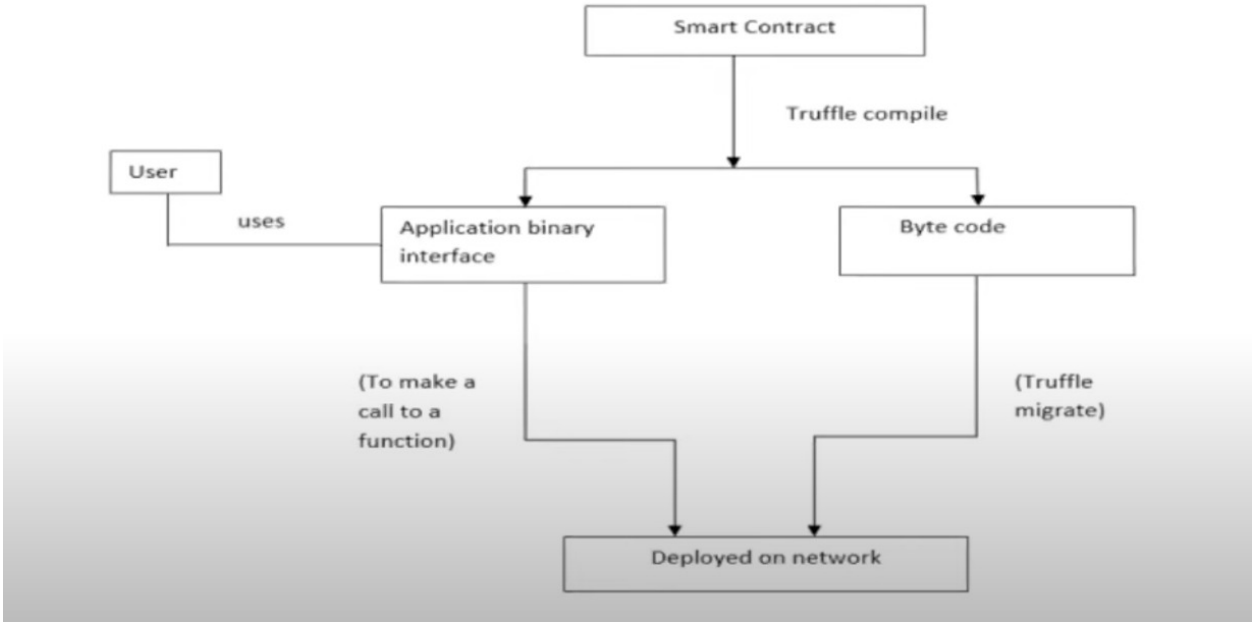
• For performing any transaction on the blockchain we require an account which will have unique account address. This can be created by using the Metamask chrome extension.

• Metamask is a crypto wallet & gateway to blockchain apps. It generates passwords (in the form of mnemonic land keys on your device, so only you have access to your accounts and data. It helps users in interacting with the blockchain network

Ganache

• Since working with the main ethereum network costs actual money for transactions, we are using a local RPC "Ganache".
• Ganache is a local test network for rapid Ethereum and distributed application development.
• It can be used across the entire development cycle; enabling us to develop, deploy, and test our dApps in a safe and deterministic environment.
• It provides us 10 accounts each having 100 ethers for testing purpose.

Truffle
• Now to interact with our compiled smart contract in a hassle-free manner we use Truffle suite.
• Truffle is the most popular development framework for ethereum which makes lots of work easier.
• This generates an artifacts which plays an important role in the successful deployment of our application.
• It takes care of managing our contract artifacts so we don't have to include support for custom deployments, library linking etc.



In the truffle project, we have a total of we have multiple folders named contracts, migrations, node_modules, source, and test. source contains the CSS and javascript files, migration contains deploy contracts and initial migration and in contracts, the solidity files are there, Election.sol and Migrations.sol.

The basic HTML code is there in the index.html file

We used port 8545 and the pragma solidity version 0.6.12, we are using ganache as Ethereum local workspace and we are using metamask as our wallet, having setup ganache local wallet on the metamask.

Election.sol
It is a solidity code to implement the backend part of the code.

The following member variables and functions have been used in this file:-

struct candidate-it stores the id,name and voteCount of the candidate.

A map- voters for voters that implies whether they have voted or not

A map- candidate that stores the no. of votes for a candidate.

Integer candidatesCount that stores the total number of candidates

A constructor that initializes with 2 candidates namely Candidate1 and Candidate2

Function addCandidate to add a candidate.It increments the count of candidates and maps accordingly in candidates map.

Function vote that enables voters who have not voted to vote.