Decentralised Voting System Using Blockchain

S.K.Risheek Rakshit
Department of Computer Science and Engineering
College of Engineering Guindy,Anna University
risheekrakshit7@gmail.com

Abstract:

Building an electronic voting system that satisfies the legal requirements of legislators has been a challenge for a long time. Distributed ledger technologies is an exciting technological advancement in the information technology world. Blockchain technologies offer an infinite range of applications benefiting from sharing economies. This project aims to evaluate the application of blockchain as a service to implement distributed electronic voting systems. We propose a novel electronic voting system based on blockchain that addresses all limitations we discovered. More generally this project evaluates the potential of distributed ledger technologies through the description of a case study, namely the process of an election and implementing a blockchain-based application which improves the security and decreases the cost of hosting a nationwide election.

1.Introduction:

1.1 Motivation

Election is an integral part of any democracy.In india paper voting method was being followed until 1999 which was later replaced with Electronic Voting Machine(EVM). Evm's have made the polling and the counting process much simpler but yet there are a lot of allegations regarding the reliability, safety and transparency of these machines . It is noteworthy that many developed countries have reverted back to the paper Ballet. Blockchain is a fast growing technology with immense potential in many sectors.

Initially used only as the underlying technology behind cryptocurrency, blockchain is finding use cases in Supply chain, Payments, Medical, Governance etc.

This project aims to evaluate the application of blockchain as a service to implement distributed electronic voting systems. We propose a novel electronic voting system based on blockchain that addresses all limitations we discovered.

1.2 Problem Statement:

1.2.1 Issues with the Current election model:

Elections, referendums, and polls are very important processes and tools for the smooth operation of a modern democracy. Some of the key considerations from the voters' perspective are: Choice of candidate must remain secret. Vote counting should be public and accountable but at the same time should be inaccessible during the voting period. Tamper proof by a third party. Maintaining social distance

Problems with existing systems include voting systems like ballot box voting or electronic voting suffer from various security threats such as DDoS attacks, polling booth capturing, vote alteration and manipulation, malware attacks, etc, and also require huge amounts of paperwork, human resources, and time. This creates a sense of distrust among existing systems.

1.3 Problem Scope:

We aim to build an online voting system where authenticated users can cast their vote. This system can be used for internal proposal polls,board meetings or a full fledged election. Users can view the list of candidates/proposals and cast their vote. The election status can also be displayed in real-time and the results can be announced instantly once the election concludes. Using the decentralized system we can make elections more secure,reliable,immutable and transparent.

2.Related Works

A consensus algorithm is a procedure through which all the peers of the Blockchain network reach a **common agreement** about the present state of the distributed ledger. In this way, consensus algorithms achieve reliability in the Blockchain network and establish trust between unknown peers in a distributed computing environment.

2.1 Consensus

2.1.1 Proof of Work (PoW)

Satoshi Nakamoto, Bitcoin's creator, invented the proof of work protocol.Proof of Work(**PoW**)
[14]PoW on Bitcoin works in the following way: All information contained in a candidate block is calculated by its hash value. This hash value generated must meet the criteria of difficulty level determined by the system. If the hash value does not meet the criteria, then the calculation will be repeated by changing the value of nounce (number once). Nonce is a value that does not have any meaning, but is intentionally added to the block in order to generate a hash value according to the conditions. If the hash value has not met the value of the rule, the nonce value will be changed again until the miner finds a hash value that meets the criteria(refer table 1)

A key feature of proof-of-work schemes is their asymmetry: the work must be moderately hard (yet feasible) on the prover or requester side but easy to check for the verifier or service provider. This idea is also known as a CPU cost function, client puzzle, computational puzzle, or CPU pricing function.

[4]The PoW in the Bitcoin system is commonly referred to as the mining process. The mining process in Bitcoin is an attempt to perform calculations using hash functions like Hashcash for a new block to be received into the blockchain. When the result is acceptable, the new block is added to the blockchain. Here the PoW can work to protect the blockchain. With the high level of difficulty, anyone who intends to change the transactions that have been recorded in a block, must do the recalculation of the block and also the next blocks.

The central idea behind this algorithm is to solve a complex mathematical puzzle and easily give out a solution.(refer **[4]**)
This mathematical puzzle requires a lot of computational power and thus, the node who solves the puzzle as soon as possible gets to mine the next block.

2.1.2 Proof of Stake (PoS):

[5]The first Proof-of-Stakes (PoS) network, Peercoin [**16**], was developed as a PoX consensus mechanism with the aim to reduce the computational requirements of PoW. Participants with higher coin age, i.e., product of network tokens and their holding time, have higher chances to be selected. Specifically, each node in Peercoin solves a PoW puzzle with its own difficulty, which can be reduced by consuming coin age. In the more recent PoS networks, the solution searching is completely removed, and the block leaders are no longer selected by computational power. Instead, they are selected based on the stakes that they are holding.(refer to table 1)

This is the most common alternative to PoW. Ethereum has shifted from PoW to PoS consensus. In this type of consensus algorithm, instead of investing in expensive hardware to solve a complex puzzle, validators invest in the coins of the system by locking up some of their coins as stake.(refer[5])If the block gets appended, then the validators will get a reward proportionate to their bets.PoS protocol is a lot more resource-friendly than PoW. In POW a lot of resources are wasted

Table1: Comparison of popular consensus algorithms

	Proof of Work	**Proof of Stake**
Energy Consumption	High	Low
Required Tools	Mining Equipment	No equipment necessary
Security	High	Untested
Decentralized vs Centralized	Tends to centralize	Users can remain in control of their tokens

2.2 Structure of a blockchain:

Each block in the stack is identified by a hash placed on the header. This hash is generated using Secure Hash Algorithm (SHA-256) to generate an almost idiosyncratic fixed-size 256 hash. The widely used algorithm was designed by

the National Security Agency (NSA) in 2001 and was used as the protocol to secure all federal communications **[15]**. The SHA-256 will take any size plaintext as an input, and encrypt it to a 256-byte binary value. The SHA-256 is always a bit binary, and it is a strictly one-way function. The figure 1 below shows the basic 256 encryption.

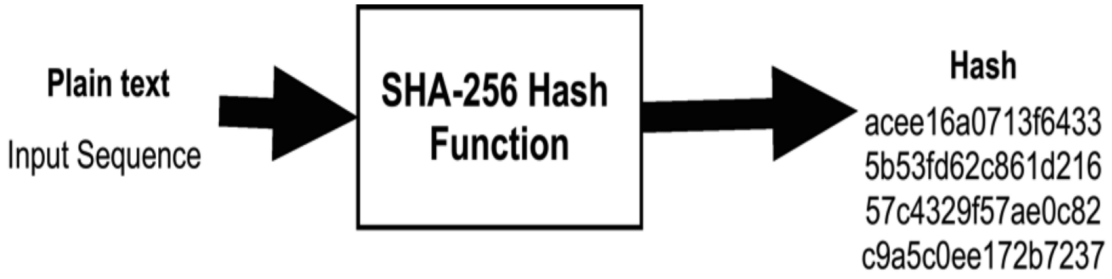

Figure 2.1 Working of SHA-256 Hash Function

The figure will be positioned right below this sentence is displaying the Basic Function of the SHA-256 Hash. Each header contains information that links a block to its previous block in the chain, which creates a chain linked to the very first block ever created, which is referred to as the foundation. The primary identifier of each 8 block is the encrypted hash in its header. A digital fingerprint that was made combining two types of information: the information concerning the new block created, as well as the previous block in the chain. The figure will be positioned right below this sentence is displaying the Creation of new Block containing a Hash Value and a Vote(refer to figure 2.2)

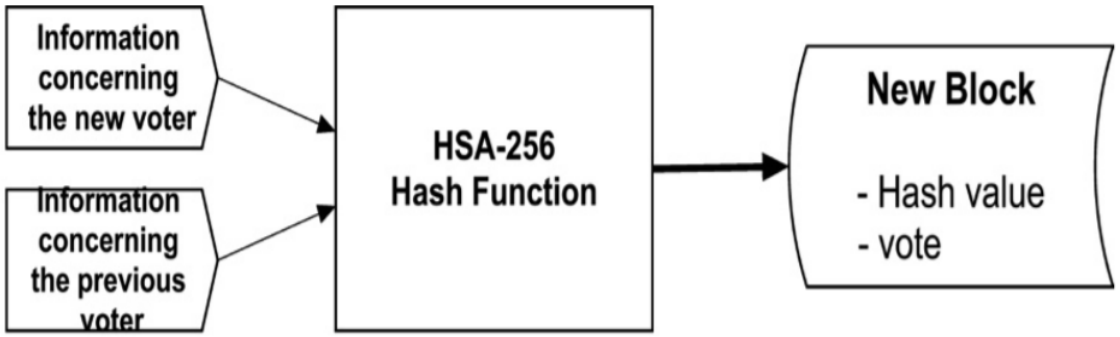

Figure 2.2 Hashing the contents of a block

3. System Design

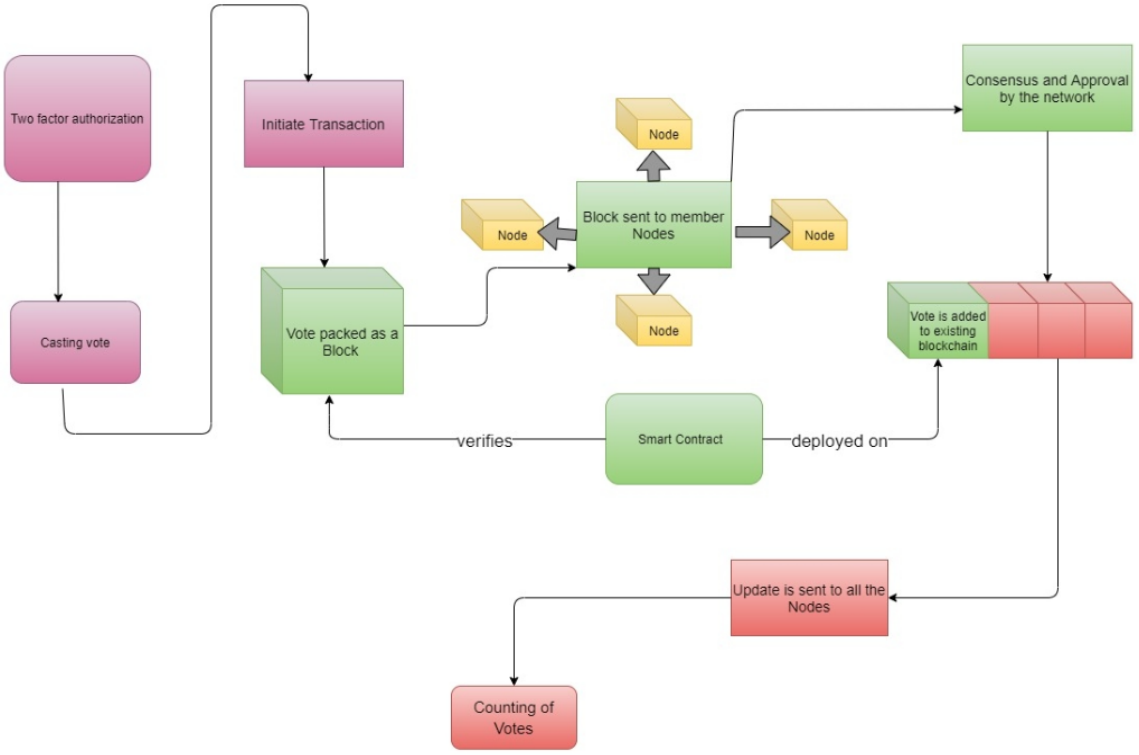

Figure 3.1 - System Architecture

3.1 Writing Smart Contracts

Smart contracts on the blockchain allow for transactions and agreements to be carried out among anonymous parties without the need for a central entity, external enforcement, or legal system. The transactions are transparent and irreversible.

Solidity is the programming language used to create smart contracts

Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour 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.

```
function vote(uint candidateId) public{
        CheckingPermission(msg.sender);
        Voter storage sender = voter_Mapping[msg.sender];

        require(
            sender.weight !=0,
            "Has no right to vote"
        );
        require(
            !sender.alreadyVoted,
            "Already voted"
        );
        sender.alreadyVoted = true;
        sender.vote = candidateId;
        Candidates[candidateId].votesGathered += sender.weight;
        // Candidates[candidateId].votesGathered += 1;
    }
```

Figure 3.1.1 Snippet of Smart Contract

3.2 Deploying the Smart Contract

Creating a blockchain pipeline for decentralised application using the ethereum virtual machine(EVM).Implementing the smart contracts on the blockchain

- ❖ We Migrate the Election Contract and deploy it on our local blockchain
- ❖ The Contract Application Binary Interface (ABI) is the standard way to interact with contracts in the Ethereum ecosystem, both from outside the blockchain and for contract-to-contract interaction.
- ❖ Data is encoded according to its type, as described in this specification. The encoding is not self describing and thus requires a schema to decode.

```
truffle(develop)> migrate

Compiling your contracts...
===========================
> Compiling .\contracts\Elections.sol
> Compiling .\contracts\elections.sol
> Artifacts written to D:\sem 6\CIP\Project\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang


Starting migrations...
======================
> Network name:    'develop'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


1_initial_migration.js
======================

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0x893256f1791ac2772d11013171509881db79f02b1648762b4b2e69d5372ab5a8
   > Blocks: 0            Seconds: 0
   > contract address:    0xE332b09062D95532186b9996895781B34dE01745
   > block number:        1
   > block timestamp:     1616057541
   > account:             0xc459914F648d472BB5862d5977E1193390E32Dae
   > balance:             99.99616114
   > gas used:            191943 (0x2edc7)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00383886 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00383886 ETH
```

Figure 3.2.1 Deploying Smart Contract

3.3 Designing User Interface

Create a User Interface for the user and admin which connects with the backend system. With the help of the UI users can login and cast their votes. Admin can start/stop the election. Results are also displayed in the webpage.

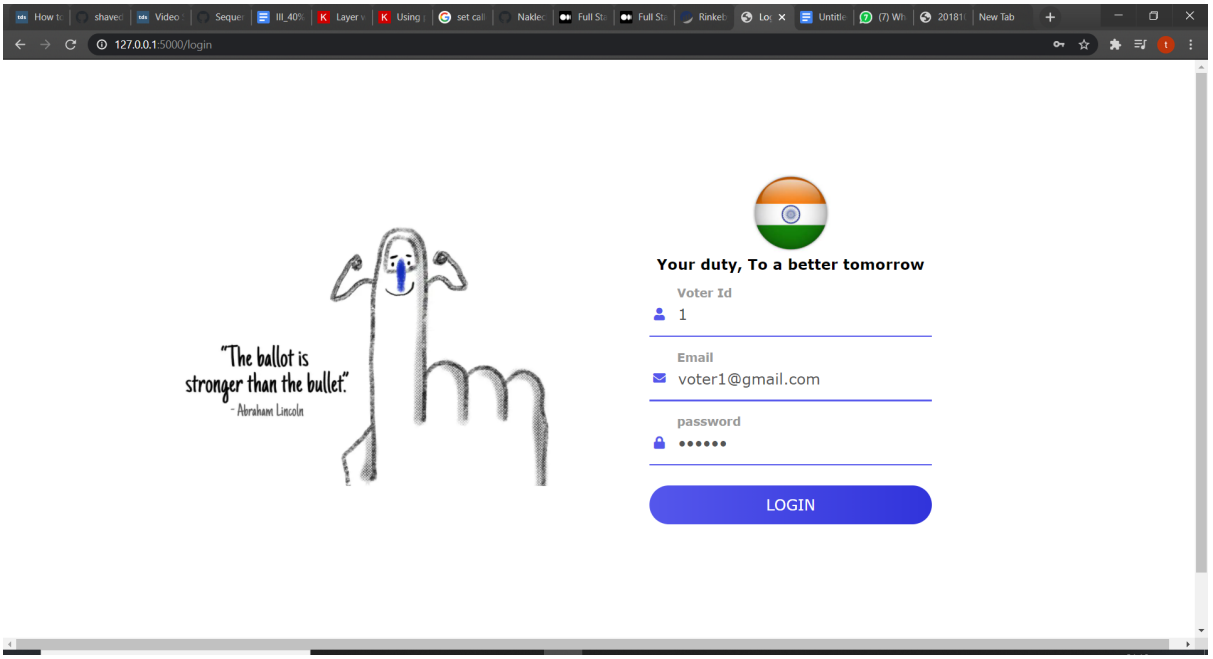

Figure 3.3.1 Login page for users

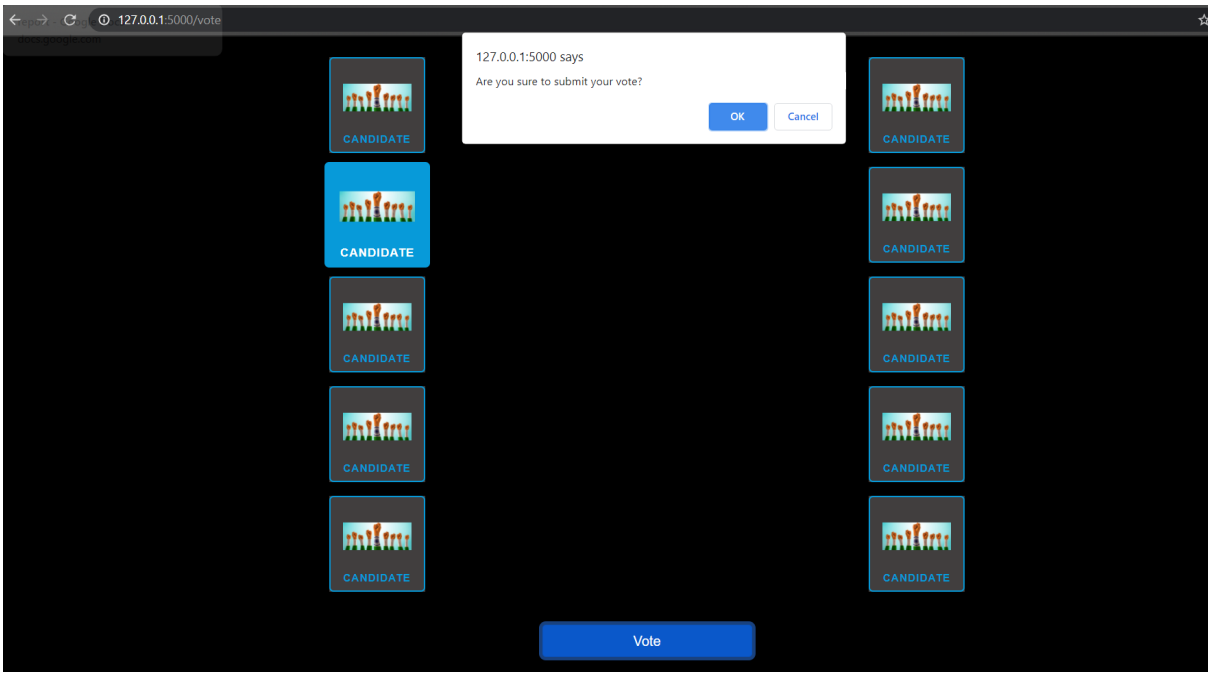

Figure 3.3.2 Page where the candidate list is visible to the voter

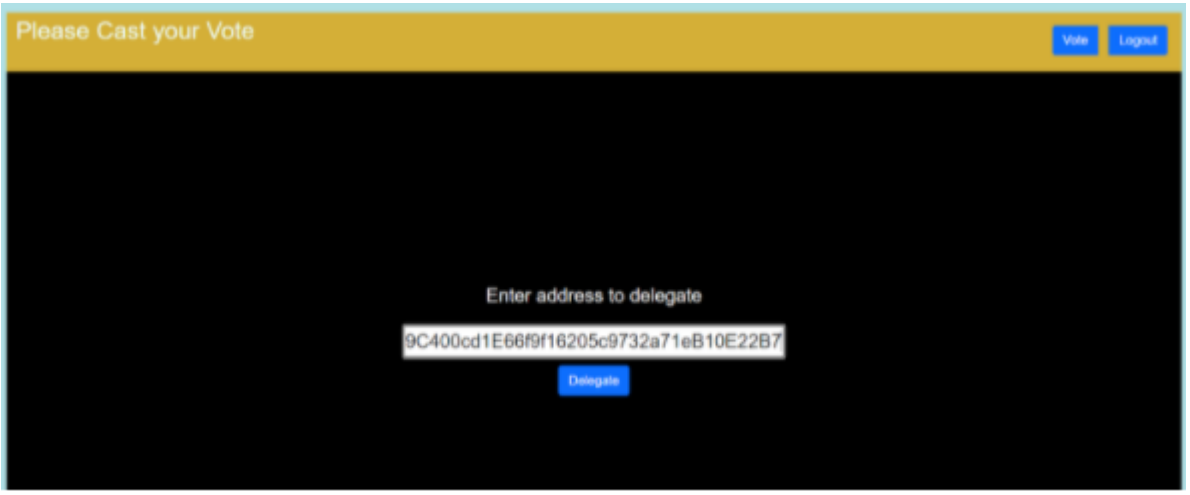

Figure 3.3.3 Page where the user enters the address to delegate

3.4 Additional Contribution

Two factor Authentication for Increased Security

- ❖ Users are first identified with their unique voter id, email and password. These details are securely stored in government database
- ❖ Facial features are matched with the reference image stored in the database
- ❖ Only when the two checks are passed user is allowed to vote
- ❖ We have provided an Admin view where the election commission officers can add/delete/modify voter details

Vote Delegation feature :

- ❖ In conventional methods if a voter is unable to travel to the voting station to cast his vote, there is no possibility to transfer his/her vote (refer to fig 3.4)
- ❖ We have incorporated a vote delegation method in our smart contract which securely delegates the voting feature
- ❖ A registered voter has to input the ethereum address of the person he wishes to transfer his voting rights to
- ❖ A transaction is created and if the person who received the delegation is authenticated, now he/she has a vote of weight 2 (his vote + received vote)

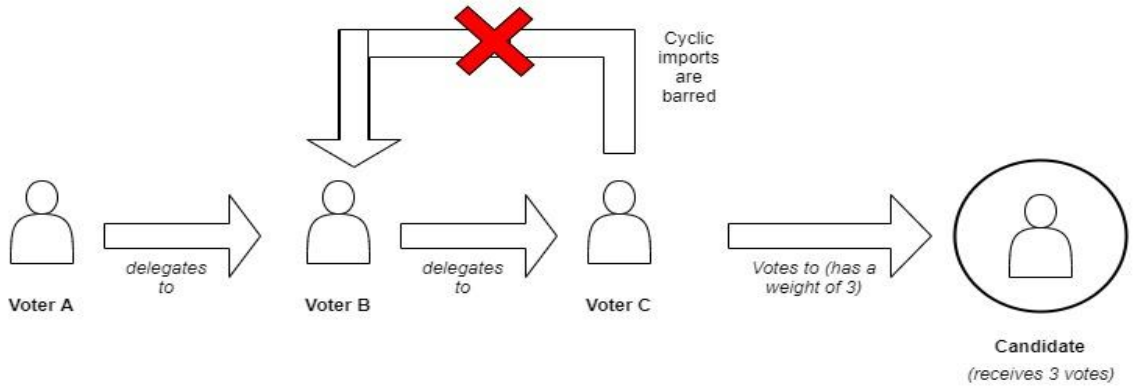

Figure 3.4 Basic intuition of delegation

3.5 Testing and Integration

An important phase in this development. Since our use case is online election/voting, there must not be any errors or bugs unnoticed. Various testing methods have to be performed to ensure that the system yields the expected outcome.(ref to fig 3.5 on the flow of the system is to be integrated)

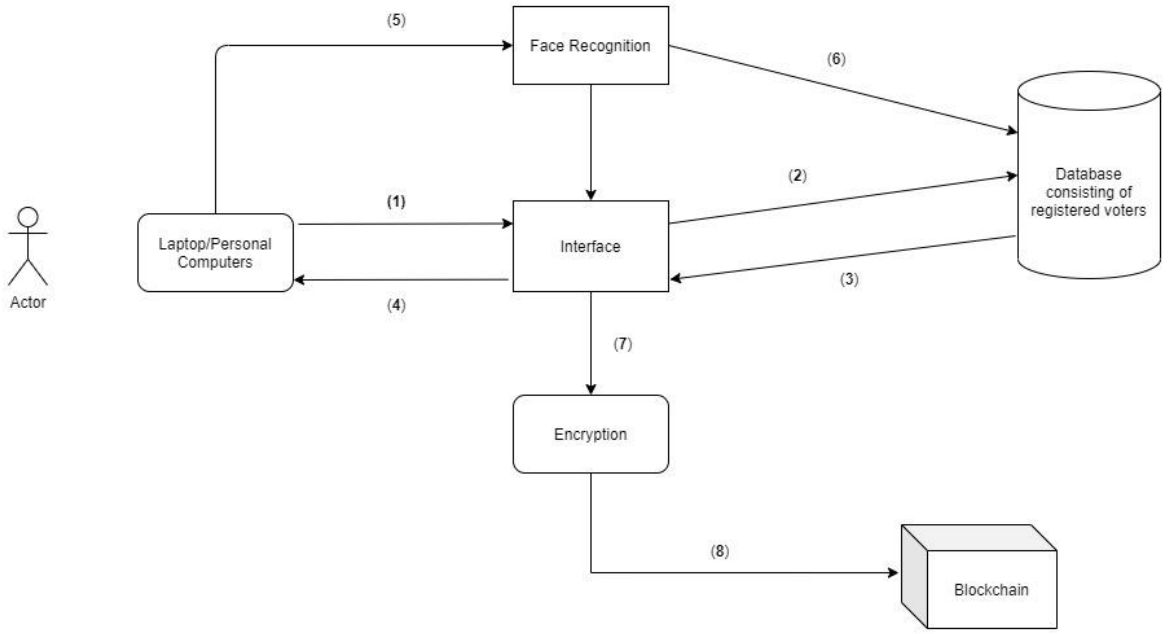

Figure 3.5 - Simplified Representation of the system

1. Unit Testing

- ❖ We have written test cases for unit testing. Each functionality of the Smart contracts have been tested and the errors have been resolved .
- ❖ Unit Testing has also been performed on 2 factor authentication

2.Integration Testing

- ❖ The objective is to take unit tested components and build a program structure that has been dictated by design.
- ❖ We have approached a bottom-up approach.Initially ,the core blockchain processes were tested and then towards the end UI components were tested.

4.Result Analysis

Gas refers to the fee required to successfully conduct a transaction on Ethereum. It is paid as an incentive to the miner
Transaction contains the data, value modified , state change etc. These values are hashed to get the transaction hash

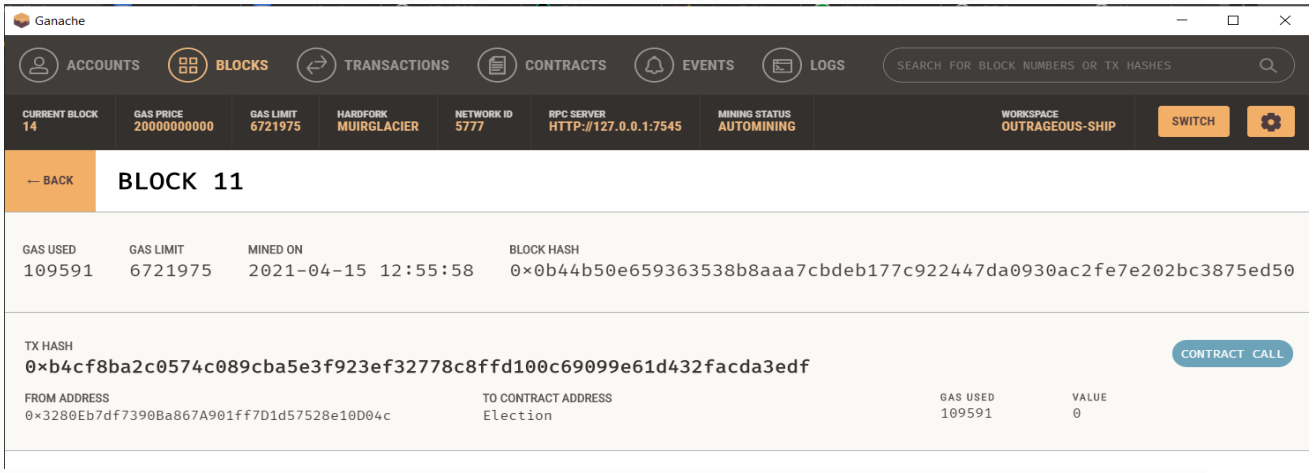

Figure 4.1 Block Details-gas used,gas-limit,mined time and nonce number

Block 11 has been mined on 2021-04-15, 12:55:5.
This block Hash this will be stored on the next block 12.

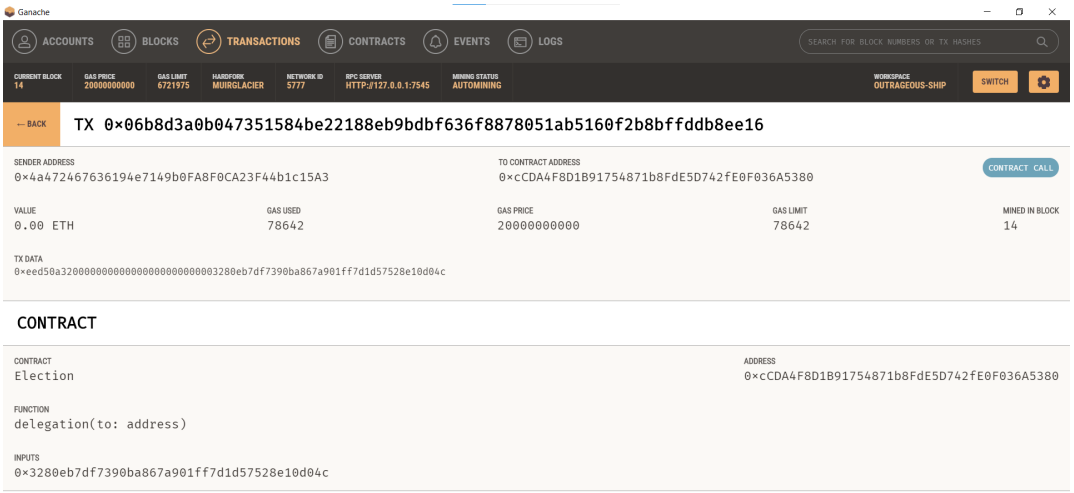

Figure 4.2 Delegate Method Transaction

The above fig(4.2) shows the transaction that is created with the delegate method in the smart contract is called

Figure 4.3 Vote Method Transaction

The fig 4.2 shows the sender address, gas,transaction data of the transaction occurred while calling

Table 2: Structure of a blockchain

Entity	Description	Size
Block size	Size of each block	4 bytes
Block header	Block headers identify individual blocks in a blockchain.	80 bytes

Transaction	Transaction taken place and stored in a block	Depends on the transaction size and the state variables

Table 3: Efficiency metric

Smart Contract Method	GAS Used	Gas Limit(Max gas the transaction can consume)
Vote	109591	6721975
Delegation	78642	1122161

Here the Tradeoff is between the extra gas used and the feature of Vote Delegation. Our system is bound to consume more gas than the existing works since for delegation a transaction is created and a gas fee has to be paid for it. We can tackle this cost by asking the user who avails the delegation feature to pay for it as it is an add on over the system.

Users will also be cautious and cost-effective when using the feature.

Equation 4.1

$$\text{Avg Gas} = G_{(\text{contract creation})} + G_{(\text{contract migration})} + G_{(\text{total votes})}$$

$$G_{(\text{total votes})} = G_{(\text{vote method})} * (n1) + G_{(\text{delegation method})} * (n2)$$

n1 = no of voters who have voted
n2 = no of users who have delegated

Base case (where n1=n2=1)

= 2,34,281 + 109591(n1) + 78642(n2)
= 191943 + 42338 + 109591 + 78642 = 422514 gwi

Conversion of gwei to eth

1 gwei = 0.000000001 eth 422514 gwei = .000422514 eth

5 Conclusion

In this Project we have proposed a blockchain based decentralized voting system. Despite a massive technological advancement in the past few decades, voting is still being done the conventional way. This is mainly due to the security concern and reliability of an online system. With the advent of blockchain we can overcome these limitations. Blockchain is immutable, decentralised, secure and these properties make it one of the brightest and a highly valued technology looking into the future. But the goal here is to focus on the process.Ethereum is the community-run technology powering the cryptocurrency, ether (ETH) and thousands of decentralized applications. Ethereum is a decentralized, open-source blockchain with smart contract functionality.

After Researching on blockchain we find that despite its various use cases it is still an infant technology with a lot of improvements to be done. Using this system small scale elections, board elections and proposals can be implemented with ease. The existing works we referred to did not have the vote delegation feature. We have incorporated the delegation feature where a voter can securely transfer his voting right to another voter.

References:

[1]Hjalmarsson, Friorik & Hreioarsson, Gunnlaugur & Hamdaqa, Mohammad & Hjálmtýsson, Gisli. (2018). Blockchain-Based E-Voting System. 983-986. 10.1109/CLOUD.2018.00151.
[2] Mrs Harsha V.Patil. A study on decentralised e-voting system using blockchain. *International Research journal of Engineering and Technology*, 1:1–23, 2018.
[3]Journal of Critical Reviews ISSN- 2394-5125 Vol 7, Issue 3, 2020

BLOCK CHAIN TECHNOLOGY FOR ELECTRONIC VOTING ONG KANG YI1, DEBASHISH DAS2*
[4]Xingxiong Zhu 2019 IOP Conf. Ser.: Mater. Sci. Eng. 569 042058
[5]Nguyen, Cong & Dinh Thai, Hoang & Nguyen, Diep & Niyato, Dusit & Nguyen, Huynh & Dutkiewicz, Eryk. (2019). Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2925010.
[6]A Ahmed Ben Ayed. A conceptual secure blockchain-based electronic voting system. *Journal of BlockChain*, 1:1–10, 2017.
[7]C. K. Adiputra, R. Hjort and H. Sato, "A Proposal of Blockchain-Based Electronic Voting System," 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2018, pp. 22-27, doi: 10.1109/WorldS4.2018.8611593.
[8]F. Þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa and G. Hjálmtýsson, "Blockchain-Based E-Voting System," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), 2018, pp. 983-986, doi: 10.1109/CLOUD.2018.00151.
[9]Vujičić, Dejan & Jagodic, Dijana & Ranđić, Siniša. (2018). Blockchain technology, bitcoin, and Ethereum: A brief overview. 1-6. 10.1109/INFOTEH.2018.8345547.
[10]Satoshi Nakamoto Bitcoin: A Peer-to-Peer Electronic Cash System 200 11.
https://ict.moscow/en/news/how-moscow-organized-voting-on-blockchain-in-2020/
12.Russia adopts blockchain voting for key referendum, Supreme Court- June 12, 2020 -Ledger Insig
[13]A. L. Selvakumar and C. S. Ganadhas, "The Evaluation Report of SHA-256 Crypt Analysis Hash Function," 2009 International Conference on Communication Software and Networks, 2009, pp. 588-592, doi: 10.1109/ICCSN.2009.50.
[14]Gemeliarana, I & Sari, Riri. (2019). Evaluation of Proof of Work (POW) Blockchains Security Network on Selfish Mining. 10.1109/ISRITI.2018.8864381.
[15]A. L. Selvakumar and C. S. Ganadhas, "The Evaluation Report of SHA-256 Crypt Analysis Hash Function," 2009 International Conference on

Communication Software and Networks, 2009, pp. 588-592, doi: 10.1109/ICCSN.2009.50.