TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PASCHIMANCHAL CAMPUS

LAMACHAUR, POKHARA

A Project Report on

DECENTRALIZED VOTING SYSTEM BASED ON BLOCKCHAIN

SUBMITTED BY:

Sagar Shrestha (PAS071BEX433)
Sajal Timilsina (PAS071BEX434)
Sandesh Chhetri (PAS071BEX435)
Shree Krishna Jamakatel (PAS071BEX439)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

SEPTEMBER, 2018

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PASCHIMANCHAL CAMPUS

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled "Decentralized Voting System Based on Blockchain" submitted by Sagar Shrestha, Sajal Timilsina, Sandesh Chhetri and Shree Krishna Jamakatel in partial fulfilment of the requirements for the Bachelor’s degree in Electronics & Communication.

____

Supervisor, Er. Hrishikesh Tiwari
Lecturer, W.R.C.
Department of Electronics and Computer Engineering

____

External Examiner, Er. Ram Sharan Baral
Telecom Engineer, Nepal Telecom

____

Head of Department, Er. Hari Prasad Baral
Lecturer, W.R.C.
Department of Electronics and Computer Engineering

DATE OF APPROVAL: 30.Sept.2018

COPYRIGHT

The author has agreed that the Library, Department of Electronics and Computer Engineering, Paschimanchal Campus, Institute of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, Paschimanchal Campus, Institute of Engineering in any use of the material of this project report. Copying or publication or other use of this report for financial gain without approval of the Department of Electronics and Computer Engineering, Paschimanchal Campus, Institute of Engineering and author's written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head
Department of Electronics and Computer Engineering
Paschimanchal Campus, Institute of Engineering
Lamachaur, Pokhara
Nepal

ACKNOWLEDGEMENT

We would like to express our gratitude to the Department of Electronics and Computer Engineering, Paschimanchal Campus and to several individuals of the department for continually providing support and guidance throughout the project.

The completion of this project is the outcome of incessant guidance and supervision bestowed by our project supervisor Er. Hrishikesh Tiwari who aided us with his knowledge and several materials to achieve the goals of our project.

A special thanks to Er. Hari Prasad Baral, Head of Department, Electronics and Computer Engineering for providing us the opportunity and necessary resources to undertake this project. Also, we would like to thank our BEX-071 classmates for providing us with technical support, suggestions and guidance throughout the project development.

ABSTRACT

The idea that we are still voting with pen and paper conventionally is so much redundant and error/fraud prone. And, electronic voting — whether at local or national government level, or in the context of corporations — is justifiably regarded with suspicion as the results seem open to manipulation without the relevant oversights. Because of the transparency offered by public blockchains, proponents of open government are vocal about the advantages of decentralized blockchain-based voting.

Blockchain is offering new opportunities to develop various new types of digital services. While research on this topic is still ongoing and emerging, most usually focus on the technical and legal issues instead of taking full advantage of this promising concept and creating advanced digital services. This project is going to leverage the open source Blockchain technology to propose a design for an electronic voting system which is secure, reliable and can improve the trust of citizens in their government.

Keywords : E-voting, Private Blockchain, Mining, Decentralization, Consensus

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

- *API - Application Program Interface*
- *CSS - Cascading Style Sheet*
- *DDoS - Distributed Denial of Service*
- *DRE - Direct Recording Electronic*
- *GIS - Global Interpreter Lock*
- *HTML - HyperText Markup Language*
- *HTTP - HyperText Transfer Protocol*
- *I/O - Input/Output*
- *IOT - Internet of Things*
- *JSON - Javascript Object Notation*
- *LAN - Local Area Network*
- *NIST - National Institute of Standards and Technology*
- *PoW - Proof of Work*
- *P2P - Peer to Peer*
- *SHA - Secure Hash Algorithm*
- *SMTP - Simple Mail Transfer Protocol*
- *UI - User Interface*
- *WWW - World Wide Web*

INTRODUCTION

1.1. Background

Voting has been a crucial event in many problem solving applications for the human generation and providing for it a fair and non-manipulated system has always been a challenge. The system has evolved by various ideas of the human mind and technologies from paper voting system to digitally signed electronic voting and even over the internet. While most notable voting applications are for holding elections at various levels, voting systems have an integral role in many decision making situations in society.

Many voting applications often fail to produce a fair output due to manipulation of votes in the systems popular even today. They are susceptible to unwanted changes and data loss in such that they cannot be trusted fully with ease. Paper ballot system has its repercussions in high since people that get one chance to participate in voting may end up placing the vote at an invalid area of the page, the vote management itself is redundant and time consuming. The electronic systems that most countries have adopted to tackle inefficient database and vote management also are full of potholes, susceptible to cyber crime, theft and machine failure. Even this system is yet to approach our country. Another system is the web based system, where voting is done over the internet using a simple application even at home but again with centralized system, it is not completely trustworthy.

With the emergence of new systems and applications in data science, a different technology has risen into popularity, not just because of the hype that the current world tend to bring about, but because it looks promising as a potential solution to most demerits of other systems. Blockchain has evolved as a solution for safe, secure and fair financial application, most notably Bitcoin; while its wide range of other applications make it all the more appealing. Along with financial systems and file management system, the integration of both these systems can be an important platform for voting applications. While it has not been applied yet for political elections in countries, it has been proposed in many cases and some even have approached closer to decentralization and secure voting systems. This project proposes one such system.

1.2. Statement of Problem

The traditional system, as in our country, implements voting via the ballot papers which are hand counted by designated personnel. The management of these paper ballots has been an issue and hand-stamping for voting has not always resulted in valid votes, mainly due to voters sometimes stamping on the paper out of the valid area out of intention or purely accident. There is a chance of data loss in case the papers are not handled correctly, some votes may be unaccounted for, while some may be double counted since there are multiple people usually employed for vote counting. It cannot be trusted because any of these staff could themselves be involved in intentionally miscounting the votes, under pressure or just for making a mockery of the system.

The traditional e-voting systems have been evolved but employ a centralized database for storing votes in a server that can be tampered or altered or hacked with some effort and hence is still not secure.

One more issue this project attempts to address is people being oblivious to such a clean solution to voting systems in blockchain and the government even banning an application, Bitcoin [1]. The full advantages of Blockchain are yet to be explored in many parts of the world, even if it is gradually growing in popularity and application.

1.3. Objectives

The main objective of this project is to develop a private Blockchain system to support for voting systems in the field of selecting a choice from a number of available candidates. This system allows the user to view the data without requiring an extra layer of security.

1.3.1. General Objective

- to develop a system which stores, manages and verifies votes as transactions in a blockchain and does this in a decentralized and secure manner.

1.3.2. Specific Objective

- to explore the boundaries of Blockchain
- to implement SHA-256 algorithm in security applications
- to establish a decentralized network of devices and allow decentralized storage
- to understand how mining works and results in such a secure blockchain paradigm

LITERATURE REVIEW

2.1. Evolution

Traditional paper ballot systems have widely been in use since ancient times for voting in elections at various levels. It has been used by almost every country at least at one time. It dates way back to ancient Greece, practicing the earliest forms of democracy, and conducting negative elections; i.e. the voters (male landowners) were asked to write the name of the candidate, on broken pieces of pots as ballots, who they wanted to be exiled for the next 10 years [2]. During the 13th century, the Venetian state solidified and elected a Great Council comprised of 40 members to implement approval voting for finding the most highly acceptable candidate. Voting system in the US has evolved dramatically since 1776, the paper ballots replaced by mechanical devices in the late 1800s, and pointer/punch card system recorded votes since around 1892 [3]. These improvements greatly reduced vote counting speed and reduced potential error and fraud in contrast to hand-counted papers. Nepal has yet to cross this barrier. The first ever electronic voting system was introduced by David Shaum in the early 1980s that used public key cryptography to cast votes and keep them anonymous, also employing Blind Signature Theorem [4]. These electronic systems involve either Direct Recording Electronic System or the Internet Voting Systems. Blockchain has not yet been implemented in election processes but still proposed. On March 7, 2018, Sierra-Leone was reported to have first employed blockchain to tally its presidential elections but only to have been informed later that it was used alongside the actual system (i.e. indirectly) to see how elections could be conducted using Blockchain [5]. But the biggest news on blockchain based voting system was the successful completion on May 8, 2018 of first government-run, blockchain supported voting in US history. Although implemented in small scale and used by only a selected group of voters such as military personnels, the response was positive and the idea of blockchain in voting continues to ramp up.

2.2. Paper Ballot Voting System

It is the most traditional means of voting and still very much in use in Nepal for all level of elections. The process for voting and handling the votes is the same as used (or was once used) in other parts of the world. The system looks fairly (or misleadingly) simple but still has drawbacks. The voters at each polling place are issued paper ballots, and the ballot is marked at different locations according to which position the candidates are voted for. Then the voters deposit it in a ballot box. They can only vote if they have with them their voter ID card. After the polls close, the box is opened, the ballots are hand-counted by the election judges, and the totals are reported [6]. The threats that must be accounted for in a paper ballot election include ballot box stuffing and dishonest ballot counting. In addition, we must guard the privacy of the voter, so that voters are not subject to harassment (or worse) because of the way they cast their vote. The election officers are given the authority to interpret which votes are valid but constrained under government law and administrative rules. Still, human ego and nature cannot stop them completely from swaying their duties and may become biased on what to considered valid vote. Ballot box transportation also needs to be assessed to prevent tampering when taken from the polling station to the counting location and again from storage location to the recounting location.

A research about ‘Measuring the Usability of Paper Ballots: Efficiency, Effectiveness and Satisfaction’ [7] explores the usability of three different voting methods using the metrics recommended by the National Institute of Standards and Technology (NIST): efficiency, effectiveness, and satisfaction. Three ballot types were studied - bubble, arrow and open-response. It was found that all three ballot types produced reasonable levels of efficiency; i.e. reasonable voting time completion. Effectiveness was a concern with unreasonable percentage of error. In terms of satisfaction, Bubble ballots were found to be most preferred but this could have mainly been because the participants were college students and they were quite familiar with that ballot type, encountering them frequently in colleges.

2.3. E-voting in the USA

In the United states, currently, various forms of electronic voting systems are in practice, all of which use computers to tabulate the votes [3]. Some of them use computers as an input device for voters to cast a vote. These systems differ in how the voters verify their vote. The revised Federal Voting System Standards recognizes two methods :

- Direct Verification - Here, the voter gets to check their ballot in which they voted in its original form. The system is based on punch card system and optical scanning.
- Indirect Verification - Here, the voter's ballot is recorded on some computer-readable medium and then displayed back electronically. The system uses DRE machines to provide for electronically voting and electronic verification.

Each state are responsible for setting their own standards and procedures for registering voters, casting ballots and counting votes while the federal government is uninvolved in day-to-day election operations. The integrity and security of US elections are tied completely to the integrity and security of computer and software since they are a part of almost every functions in voting. A specialized software is run on computers at polling place, which are off-the-shelf desktop machines running on a standard operating system equipped with electronic mail and web browser software along with specialized voting software. Computers at each election offices are connected to one another via wired or wireless LAN that may have a direct or indirect connection to the Internet. Some states outsource various back-end functions, often concerned with voter registration database and ballot definition, to an election service contractor providing specialized assistance. Most employ DRE machines which stores voter's choice on a memory card. [8] Prof. Blaze (2017) provided a testimony on the security issues in electronic voting systems and even recommending the DREs be replaced by paper ballots and risk-limiting audits. The design of DREs makes them inherently difficult to secure and also is a must that they should be secure because any attacker with the capacity to alter the software running on the machine has the potential to alter the vote tally, as well as make it impossible to conduct a recount.

2.4. Estonian Internet Voting System

Estonia became the first country to allow citizens to cast their vote over the internet in the parliamentary election in 2007 after their pilot success in municipal elections in 2005, with upto 25% voters casting their ballots online [9]. [4] The voters require only their electronic national identification card designed to run on an IC Java chip platform and protected with 2048 bit pin. It creates signatures using SHA1/SHA2. The voter should download the voting application, authenticate themselves using their electronic ID. If the voter is eligible to vote, a list of candidates are displayed and a vote can be cast. The vote will be encrypted using the election's public key and signed with the voter private key. As soon as the vote is cast, it will be sent to a vote storage server controlled by the Estonian government. Voters could vote multiple times, and only the last vote will be considered valid. This is done to prevent vote buying.

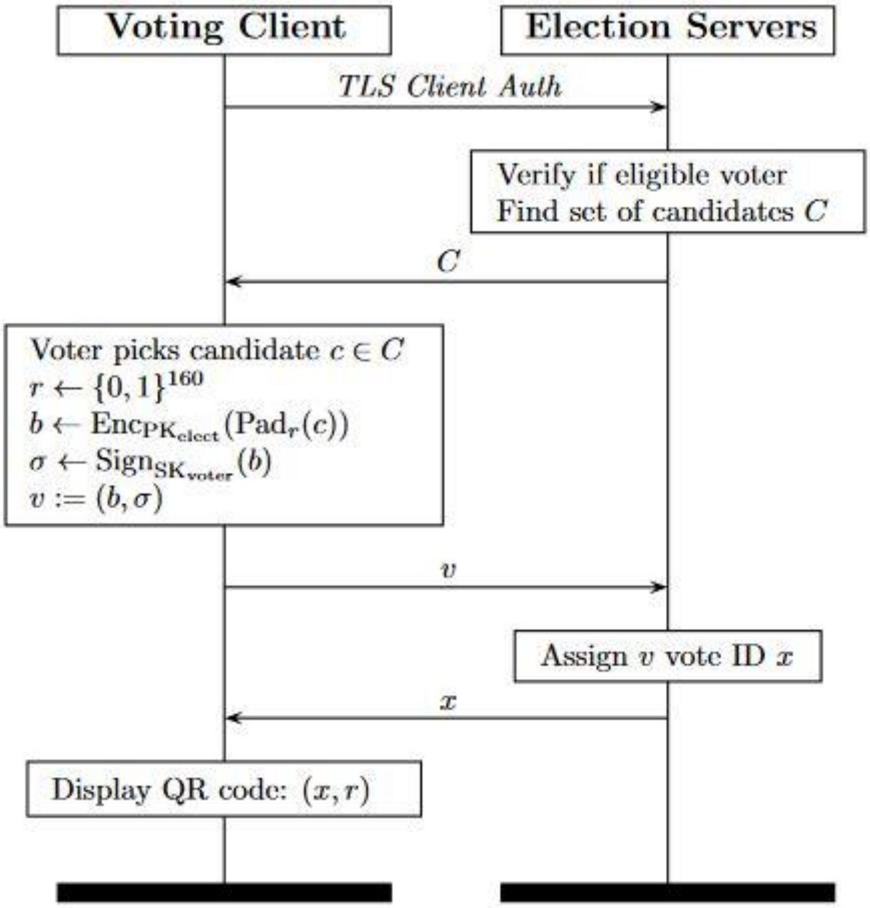

Voting Client
Election Servers
TLS Client Auth
Verify if eligible voter
Find set of candidates C
C
Voter picks candidate $c \in C$
$r \leftarrow \{0,1\}^{160}$
$b \leftarrow \mathrm{Enc}_{PK_{elect}}(\mathrm{Pad}_r(c))$
$\sigma \leftarrow \mathrm{Sign}_{SK_{voter}}(b)$
$v := (b, \sigma)$
v
Assign v vote ID x
x
Display QR code: (x, r)

Fig 01 : Vote Casting in Estonian I-Voting System [10]

The issue with I-voting is that centralization of the vote storage server makes is susceptible to DDoS attacks such that the network is flooded with multiple requests, most unnecessary, and cannot service any more requests like when a user needs to vote. And questions were raised about the secrecy of critical parts the code in Estonian I-Voting. The script to post the vote on the system's server is made closed that raise questions about transparency.
The centralized server is always susceptible to hacking attack and can compromise the whole voting system and the results.

The blockchain implementation is applied to tackle some issues that arises due to flaws in centralized, closed-source systems to establish a decentralized private blockchain network for voting application and enable secure, transparent, real-time and easy representation of transactions.

METHODOLOGY

3.1. Architecture

The system used in the project is not just specified to be for voting in various levels of elections, it can also be implemented for other voting applications like say voting for favourite contestants in a reality show and so on, since the application is used over the internet.

The simple architecture of the project is given below :

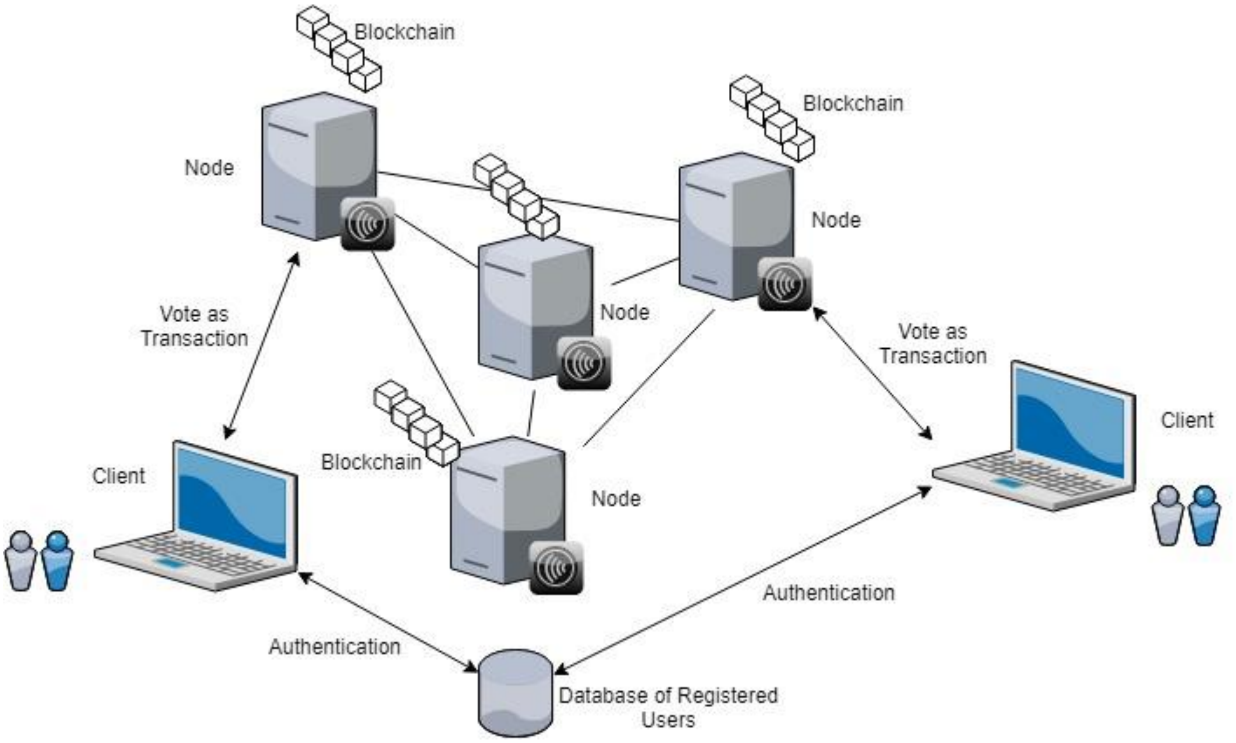


Fig 02 : Project Architecture

The following components are involved :

3.1.1. User

The user here refers to the voting personnel who is authorized to cast a vote. The user, in election applications, has to go to a polling station in order to cast their vote where they are provided a laptop or any device (client side) with a working internet connection and a browser. The device should be connected to the private blockchain via arbitration server, discussed later. The user would be redirected to the user interface in the browser allowing them to vote.

3.1.2. Database and Authentication Server

The authentication server is a centralized web server with a backend database that contains the credentials of all the people eligible to vote [11]. Each such user is provided with a unique ID in the database for this project so that they can log in with that credential to acquire access to vote when the time comes. The credential that they enter in the authentication portal (fig. 03) is cross-checked with the database and allows access if the user is registered in the system. As they are granted access, they are required to join the network so that they can get a token from the network to cast the vote.

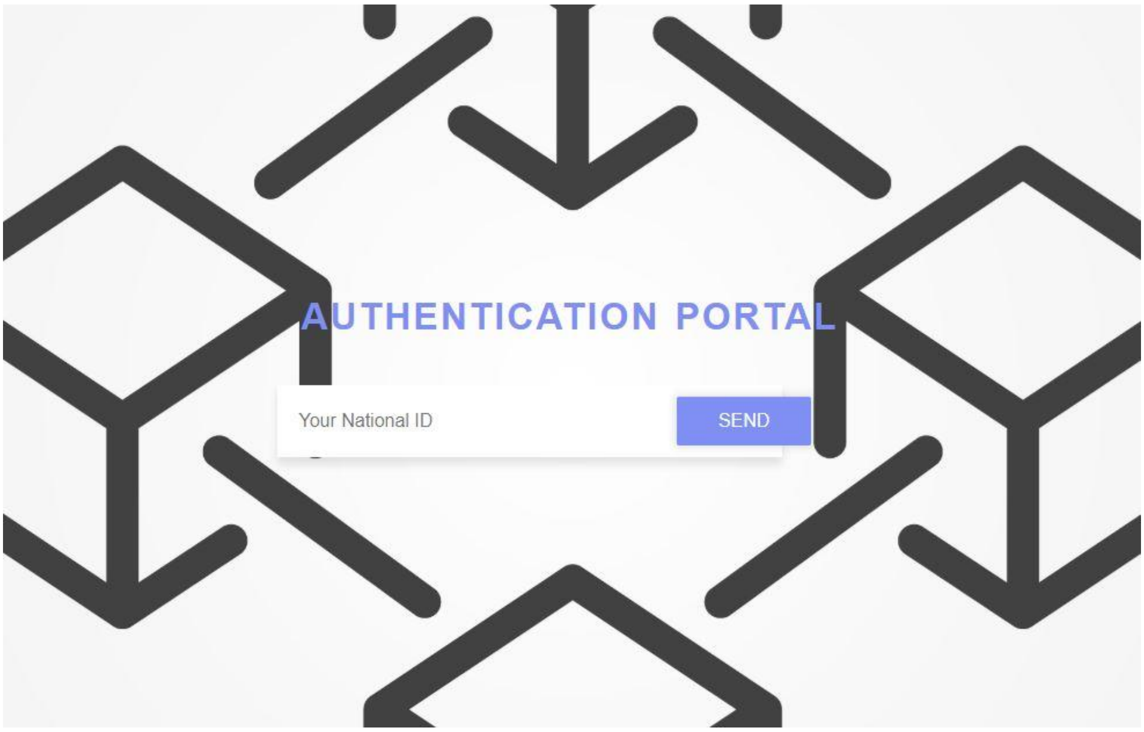


Fig 03 : Authentication Interface

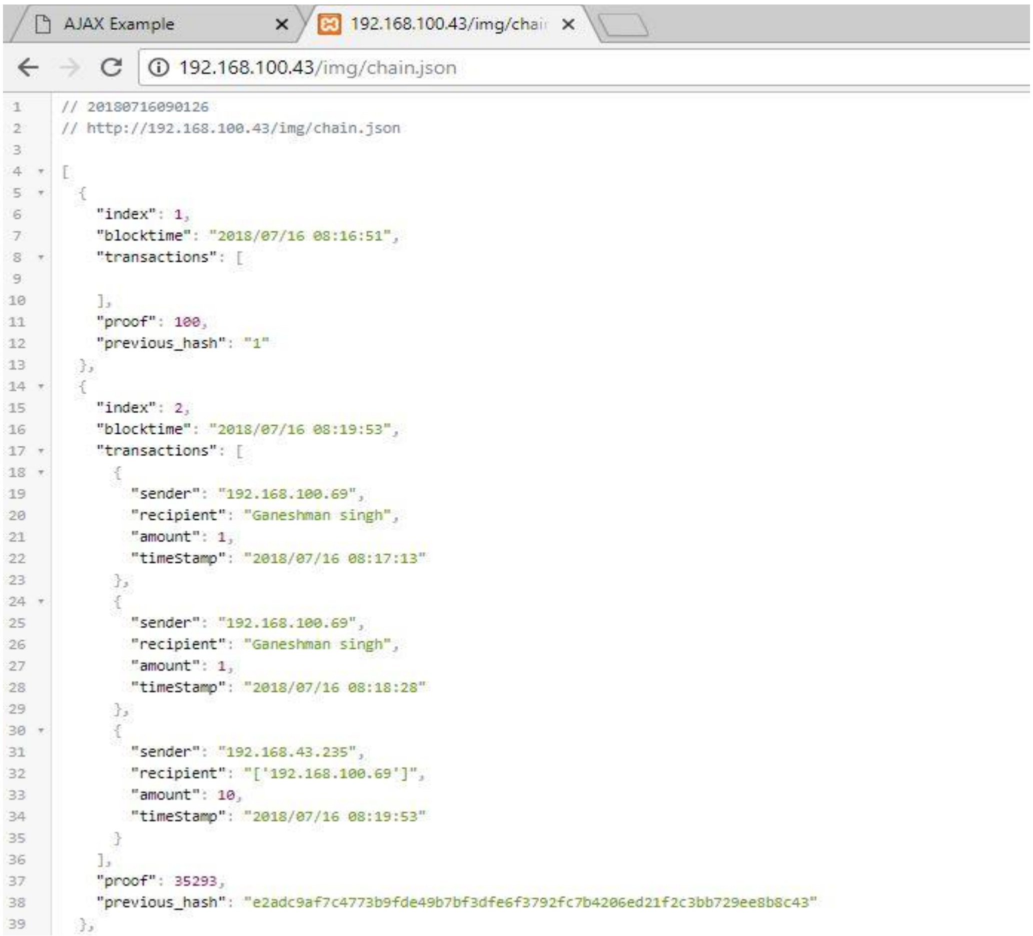
The user makes a new transaction for the blockchain by selecting a candidate in the browser interface and casting the vote. The transaction takes three parameters into account:

- Sender : It is the user who is provided with a unique IP address when s/he gets authenticated to the network.
- Recipient : It is the candidate to whom a user selects to vote. They are also provided with IP addresses of their own to which the votes can be dumped into.
- Amount : It is represented as a tokens in the system. Each Token represents a single vote transferred from the sender to the recipient when selected. A user is provided with a single token to vote for their favourable candidate when they access the system. The sender token would be reduced to zero while the candidate's token will increase by one for each vote they receive.

3.1.3. Arbitration Server

The arbitration server is the part that acts as the mediator between the user interface and the blockchain backend. An API, which is built in Python Flask framework, allows the user to explore the data inside the blockchain network. Thus this allows a user to view all that is in the blockchain. In this system, they can view all the transactions/votes that is ever made and even view the data about the specific user or candidate and thus provides transparency. Since the voter is represented by their IP address, they remain anonymous. The votes can be evaluated by simply viewing the amount of tokens for each candidate.

Through API calls, the amount of tokens of each candidates is imported from the blockchain system in JSON format. This JSON data is parsed with Javascript and presented in the web browser when enquired.



```
// 20180716090126
// http://192.168.100.43/img/chain.json

[
  {
    "index": 1,
    "blocktime": "2018/07/16 08:16:51",
    "transactions": [

    ],
    "proof": 100,
    "previous_hash": "1"
  },
  {
    "index": 2,
    "blocktime": "2018/07/16 08:19:53",
    "transactions": [
      {
        "sender": "192.168.100.69",
        "recipient": "Ganeshman singh",
        "amount": 1,
        "timeStamp": "2018/07/16 08:17:13"
      },
      {
        "sender": "192.168.100.69",
        "recipient": "Ganeshman singh",
        "amount": 1,
        "timeStamp": "2018/07/16 08:18:28"
      },
      {
        "sender": "192.168.43.235",
        "recipient": "['192.168.100.69']",
        "amount": 10,
        "timeStamp": "2018/07/16 08:19:53"
      }
    ],
    "proof": 35293,
    "previous_hash": "e2adc9af7c4773b9fde49b7bf3dfe6f3792fc7b4206ed21f2c3bb729ee8b8c43"
  },
```

Fig 04 : JSON output

3.1.4. Blockchain System

This portion is the core of the voting system, in reality, where the actual voting takes place. It represents a network of interconnected devices to be called as nodes that store the blockchain data. Nodes in this project are present at each station that manage the data which obtain the vote from the browser interface. The node then adds the transaction to the blockchain depending on the smart contracts that exist on it. The smart contracts are the rules that the nodes follow to not only verify but also add the vote as transaction in the system. Each node follows the smart contract to verify the vote.

3.2. Layering the Project

Blockchain used in this project is built from ground up to provide for a voting system, instead of using the available platforms like Ethereum or Stellar. Such a platform is called a Private Blockchain.

Blockchain is a public ledger of transactions or information that is collected through a network sitting on top of the internet providing an entirely new way of documenting data over the internet. The way of storing is how it gets its name - transactions that require confirmations from several devices are collected in groups in what is called a block along with some information about the block itself, stored not in one central store or the web but on a number of participating *nodes* and each of these blocks successively linked together like a chain [12]. Hence the voting data is not stored in the web server but in this blockchain network. These decentralized storage, linkage and confirmations are what make it impenetrable to alteration and theft, highly available and fault tolerance.

Conveniently, the project is processed as a whole in 5 layers representation of which 4 imply what we call Blockchain [13].

3.2.1. Application Layer

Application layer provides application interfaces on top of blockchain. Applications are built in the application layer making use of blockchain protocols and networks which are described in other blockchain layers. Blockchain can be considered as a distributed backend database where information are written or read. The applications' data and records (transactions) are cryptographically stored in a decentralized manner in the blockchain. Unlike WWW where information is stored and aggregated in the application layer, which works on top of internet protocols like TCP/IP, SMTP, etc, in Blockchain most of the information is stored in protocol layer and only a fraction of information stored are distributed among the applications in the application layer.

Applications that treat blockchain as a backend are hosted on some web servers and that requires web application development, server-side programming, and APIs, etc. A good blockchain application doesn't use blockchain for everything to ensure blockchain is light and effective and without unnecessary traffic.

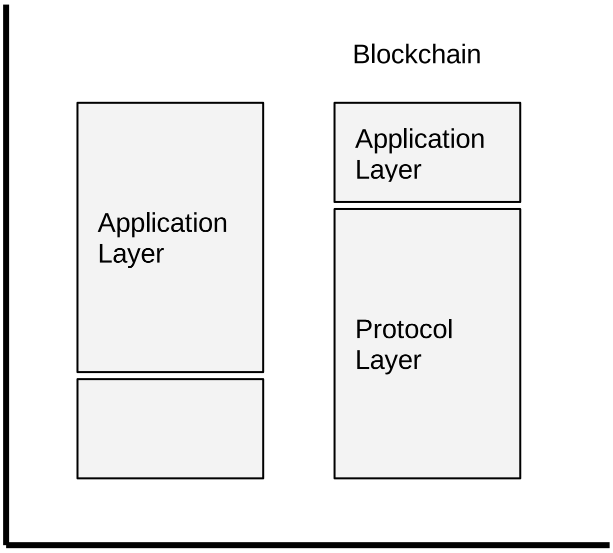


Fig 05 : Web vs Blockchain (Web 3.0)

The application we developed is a web app used to communicate with the Blockchain over HTTP. The frontend of the app is developed using HTML, CSS and JavaScript. Whereas Python Flask Framework was used in mapping endpoints to Python functions, this allows us to communicate to the blockchain over the web using HTTP requests.

3.2.2. Semantic Layer

The vote as token from the application layer now enters into the semantic layer from which the essence of what we call *Blockchain* begins. It is the layer that provides the correctness to the identity of the blocks in the blockchain based on the transactions in each block [15]. Here, the transactions are validated after execution. Execution involves performing a set of instructions on the transaction so that the designated task is done [14]. In financial applications of Blockchain, this means having the currency transferred from the sender to the receiver. In voting system, the transaction is simply a vote transferred to the selected candidate as one token from the voters account.

For transactions, the true purpose of semantic layer is to define their validity. Every node continually listen for transactions in the network. The node captures the executed transactions propagated across the network and creates a ‘transaction pool’. It then selects the eligible transactions from the pool to be added to the new block. For this, first, all the transactions already a part of any previous blocks are removed [15]. The remaining ones are verified by checking the sender’s account for the amount of tokens that they can spend [16]. This defines if one is authorized to make a transaction and prevent double-spend attack (since execution is already done, the sender’s account will have no tokens to spend, if a previous transaction proposed for the same block empties the sender’s account). Thus invalid votes are rejected before proceeding to the mining layer.

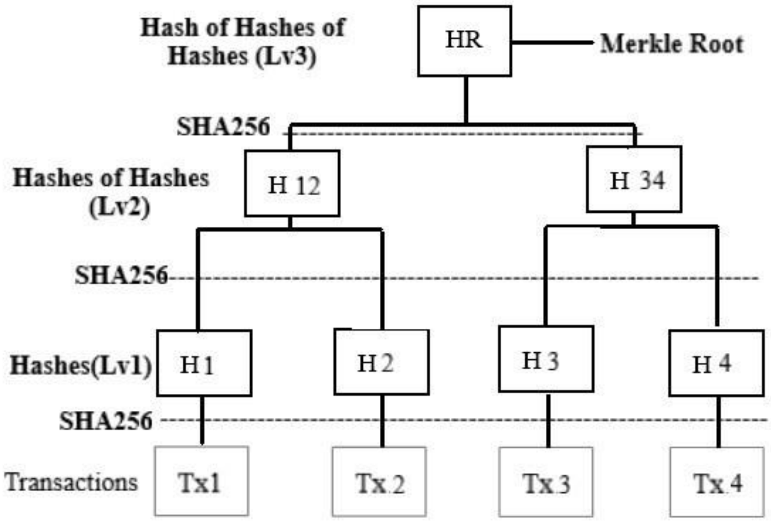


Fig 06: Merkle Tree

The semantic layer also defines a couple of components of the block header that defines a block. First is the data model or structure like Merkle tree that maintains a relation of the set of valid transactions with the block header [14]. [17] A Merkle tree results in a 64 digit hash value called the Merkle root by a number of hashing processes on the transactions and their subsequent hashes. Each transaction is hashed with a SHA256 function to generate each of their own 256 bit value. These hash values are paired and each pair is again hashed with SHA256 to produce a new 256 bit/64 digit value at the next level of the Merkle tree. This pairing and hashing is done at each level until only one hash value is evaluated in the final level which is the Merkle root. In case any level of the tree has an odd number of hash values, the last one is duplicated and then hashed with itself for the next level hash. Such Merkle tree is also an essential part of the consensus layer to validate the transactions in proposed block and maintain data integrity.

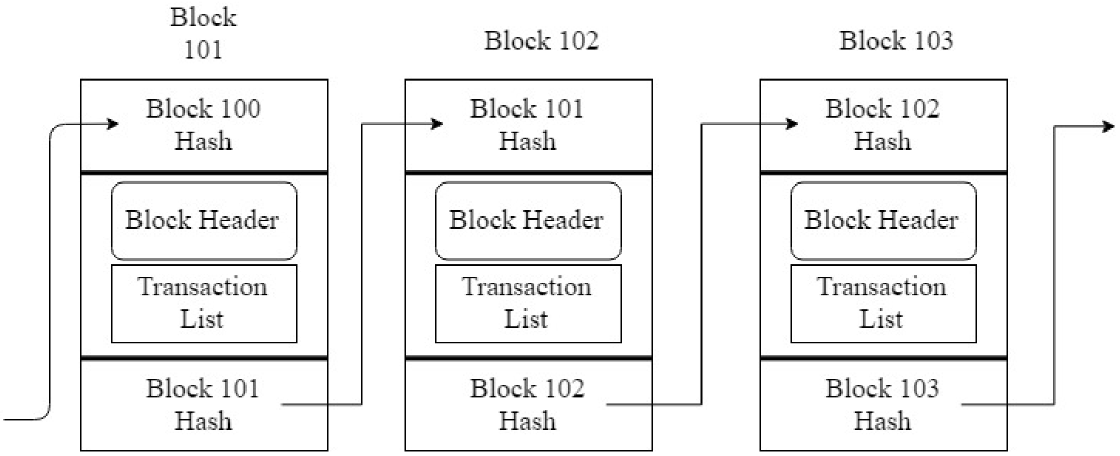


Fig 07 : Block Linkage

Second, this layer also defines how the blocks are linked with each other. Each and every block in the chain includes the hash of the previous block in its block header. This hash is the identity of a certain block resulted by the SHA256 function of the block header. The rule of the block header is defined by this layer which thus defines the difficulty for the mining layer, also a component of block header. Mining results the formation of a block by computing the block hash of the block data (discussed later) which conform to the definition set out by the semantic layer. [16] [15] The most common rule is that the hash value must begin with a certain number of leading zeroes. The highest value of this acceptable hash is called the target (which is converted to difficulty for the block with a certain mathematical algorithm). For example, Bitcoin employs a rule of requiring 18 leading zeros in the 64 digit hash for the block to be considered valid. Therefore, the target would be:
000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Hence, any hash below this target will be a valid hash for that block. The essence of target can be interpreted in another way. It is defined for the mining layer such that it takes a certain fixed amount of time to compute the valid hash by any miner. It Bitcoin, it should take around 10 minutes to compute the valid hash. Currently, this time limit implies there will be 18 leading zeros as defined earlier. In time if any node reaches that target in much less time than defined (possibly due to high computation power), then the difficulty is adjusted to change for another specific target [18].

Thus, a block has the hash of the previous block which has the hash of the block preceding it and so on until the second block has the hash of the first block also called the genesis block. As a result it this defines the linkage of all the valid blocks in the chain to the genesis block.

Genesis Block [19] : It is the first block in the blockchain to which all other blocks trace their origin to. Also called *Block Zero,* it defines the Blockchain's initial parameters like difficulty level, consensus algorithms, etc.to mine blocks. It does not have a pointer to its previous block since it doesn't have any, unlike any other block. So its previous block hash is set to 0000000000000000000000000000000000000000000000000000000000000000. Also, it doesn't hold any transaction. It is not created like normal blocks, rather hardcoded into the clients into the network for them main network. As for its mining, the network developers initially mine the hardcoded block zero. It is essential for block synchronization between peers (if they have the same one).

Node Types [20] : Nodes are devices that have internet and are connected to the blockchain network. They store the chain as a local copy using their hardware storage. On the basis of portion of chain stored, nodes are of two types :
a)*Full Nodes* - They store the complete blockchain ledger locally and thus consumes much of the nodes storage.
b)*Lightweight* or *partial nodes* - They store only a portion of the chain relevant to their use. Both of these nodes can participate in mining, but the partial node acts only for mining. So they ensure they have the latest copy of the chain by connecting to the full nodes.

The presence of a number of full nodes is precisely why blockchain is a Decentralized technology. When a number of full nodes store the blocks, one need not refer to a single central authority for the data but to any of these decentralized full nodes.

SHA-256 : It is a function that takes any length of input and creates a 256 bit value called hash as the output. Thus an input, whether one character or a whole book produces a unique hash. SHA on same input always produces the same output, hence tampering in blocks can be identified by comparing the given block hash and the hash computed by applying SHA-256 on the block which makes it easy to verify.

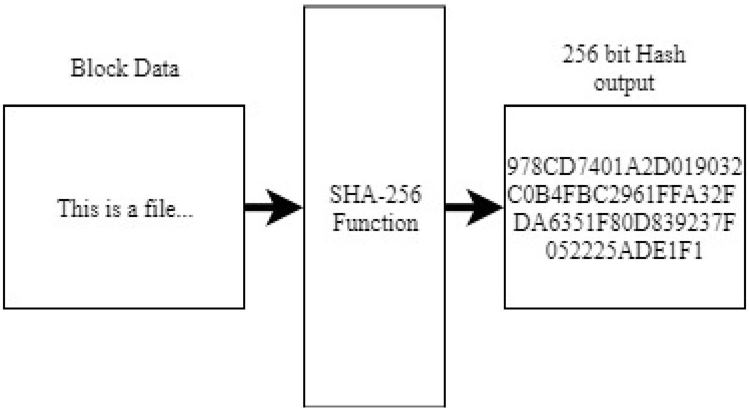


Fig 08 : SHA-256 example

It is a one way function i.e. its reverse cannot be performed do that there is no way of simply manipulating the hash by adding a malicious input which makes it all the more difficult to predict the hash of any input unless SHA applied and thus provide extra security.

3.2.3. Mining Layer

While Semantic layer defines what links two blocks, mining layer defines how the link is generated, i.e. how the block hash is computed according to the definition of the semantic layer.

Mining refers to the formation of a new valid block of transactions in the chain to be accepted by other broadcasted nodes. In financial applications, mining does not directly mean generating the cryptocurrency but it is a step in mining [15].

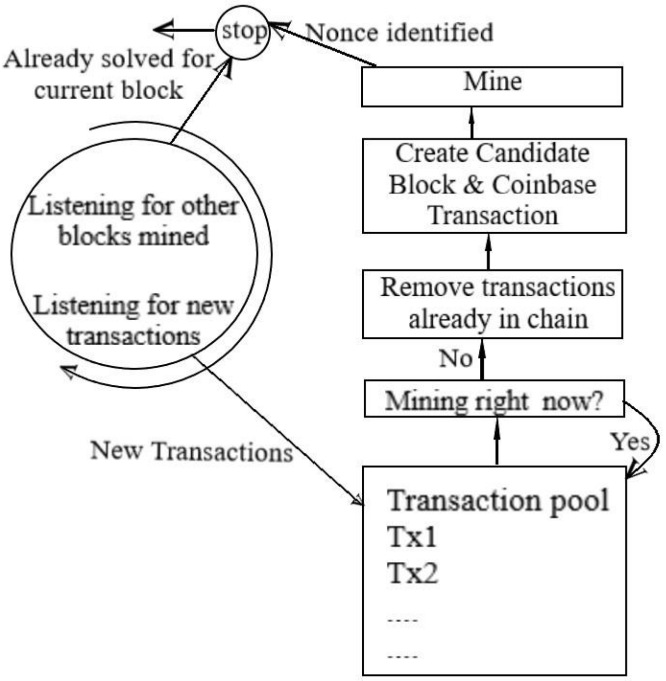

stop
Nonce identified
Already solved for current block
Mine
Create Candidate Block & Coinbase Transaction
Listening for other blocks mined
Listening for new transactions
Remove transactions already in chain
No
Mining right now?
Yes
New Transactions
Transaction pool
Tx1
Tx2

Fig 09 : How transactions are transferred for mining

First, the contents to be mined are discussed. [21] [15]

These parameters are defined in a mining node just after when the latest block is found to be mined (detected by listening on the network continually done by the mining nodes in competition, then the mining for that blocks stops as in fig no), their local copy of chain is updated after verification (consensus layer) of the newly found block and the valid transactions are selected.

- Current Block Height - the position of the block to be added in the chain
- Previous Block Hash - obtained from the local copy of the latest block
- Timestamp - the length of time (in seconds) of beginning of the current block creation from that of the genesis block
- Version - a version number to track the protocol and software updates
- Target (or Difficulty) - defined by the Semantic layer
- Merkle Root Hash of the transactions

  In usual applications, along with the regular transactions, an additional one is created called the coinbase transaction by the miner itself. This is the step for cryptocurrency creation as an incentive or reward to the miner itself for using up their computational power to solve the problem for mining. The amount of incentive is defined by the network and the creation is complete only after the block is validated in the consensus layer. If any block is not accepted as part of the chain, the coinbase transaction and the cryptocurrency generated becomes meaningless. And if currency creation in coinbase transaction is not supported then the mining layer should define the incentive and ensure that there are enough incentive for participants to continuously be able to forge and confirm a new block.

- Nonce - The most important parameter for mining is the nonce which is a numerical value when hashed along with all other parameters produces the valid block hash (as defined by Semantic layer). It is clear by the nature of the SHA-256 function that only one value of nonce can produce a definite hash.

With all the parameters above, collectively called *Block Header*, a so called '*Candidate Block*' is formed, called so because it is not yet a part of the main chain and is just one of many similar blocks being worked on by a number of nodes competing against each other to mine create the valid block first so that they can earn the incentive.

Mining :

Now, the task of the miner is to do some computational work and produce the hash under the defined target. The proof of the miner's such work is defined by the nonce, also useful for consensus layer. Hence, the problem for mining, mentioned earlier, is to find this nonce such that hashing of the candidate block produces a valid hash. Value of nonce can not be easily predicted since an increment of even one produces a completely different hash and hence is not consistent (a feature of SHA-256). Thus, nonce is iterated to be set to various different values, usually initialized to '0', until the value is found that produces a valid hash. The mined data along with nonce and produced hash completes the formation of a 'block' and thus mining is complete.

Any node mining on a block if receives a new block of the same position already validated, then it abandons that mining and starts collecting eligible transactions from Semantic layer to mine for the next block.

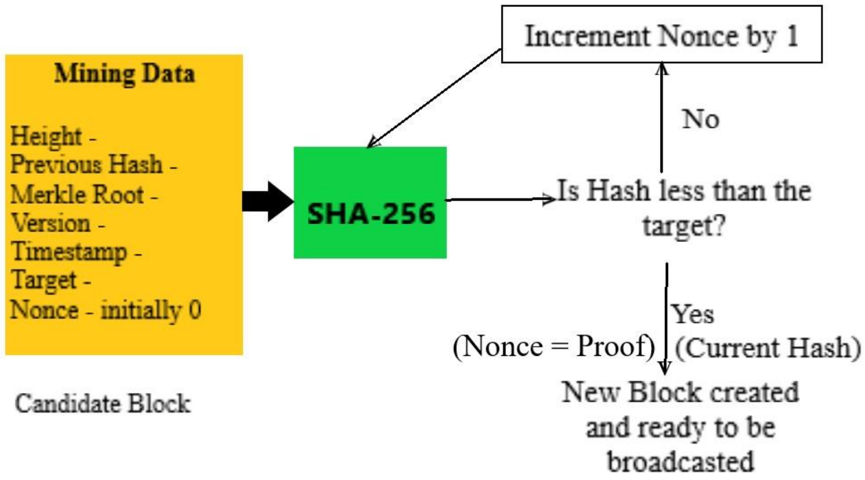


Fig 10 : The ‘Mine’ Step (Solving the computational Problem)

It is to note that the transactions are not themselves used for computing the block hash. It is indirectly used as a Merkle root and altering any of the transaction will immediately change the Merkle root which in turn changes the block hash thus still securing the transactions. To make the block valid again, new value of nonce is required which again should be evaluated since it cannot be predicted. But by the time new nonce is determined, another block would likely have been formed already making the new tampered block insignificant. And if altering of any data in any of the previously formed blocks is attempted, due to the linkage by the block hash that instantly changes on data change, it would be extremely difficult for the malicious party to have their copy of the chain accepted since they need to mine for all the subsequent blocks and have them accepted as well. By that time, blocks would be piling in the chain and the malicious party would not catch up to the latest block that way, since they require multiples of time to even catch up to the state when they started altering data, let alone the latest of the newly piled blocks. Besides, the transactions themselves can be cryptographically secure as soon as they are created by using PublicKey-PrivateKey Cryptography in the application layer and the details of the involved parties are made anonymous while still being able to show that a transaction has occurred (anonymity and transparency).

Blocks mined by different miners are somewhat different with different hash and nonce values even if all of them are valid. First, the coinbase transactions are not common since it involves the miners themselves only. Second, the time at which mining begins and the software used may all differ according to the miner. Finally, the number of and nature of transactions at their own time and often different amount of transactions are selected according to their propagation.

3.2.4 Propagation Layer

This layer defines how transactions and new blocks so formed by successful miners are propagated across the network. It is the peer-to-peer communication layer that allows the nodes to discover, talk and sync with each other to obtain the present state of the network. Once a transaction is executed, it gets broadcasted from the node of creation to all the peers immediate to the node only. During transaction propagation, different nodes will have different amount of transactions according to when they mine a block or due to latency issues [22]. The blocks being proposed are propagated in the same way to the immediate peers for verification and consensus. This includes the proof of the miner's work i.e. the nonce nad block hash with all the block contents and transactions.

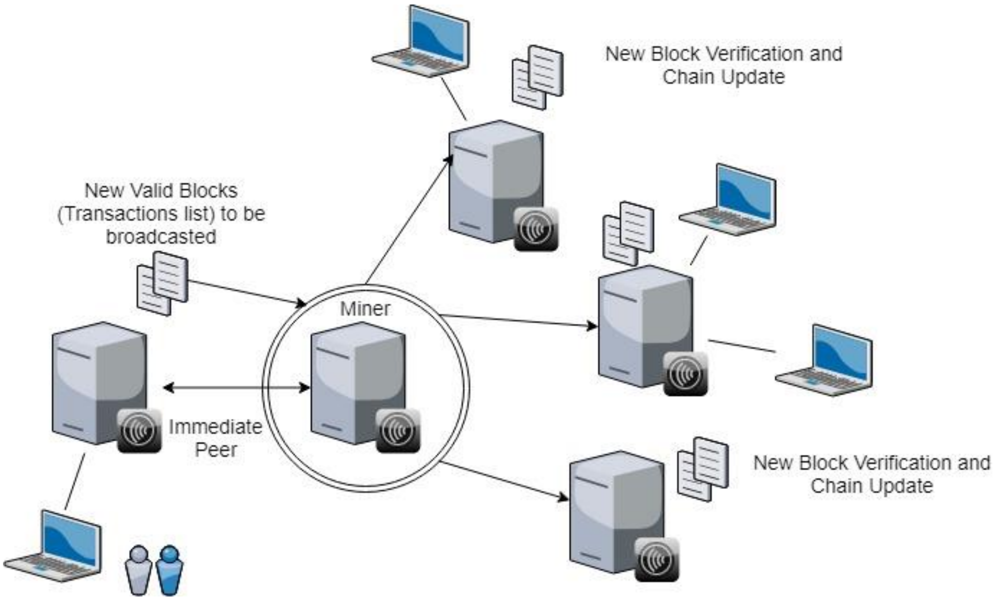

New Block Verification and Chain Update
New Valid Blocks (Transactions list) to be broadcasted
Miner
Immediate Peer
New Block Verification and Chain Update

Fig 11 : Peer-to-peer decentralized communication

The immediate peers verify the block (discussed in consensus layer) and then propagate the block, if valid, to its immediate peers and so on until the entire network at some time will have the latest block.

The network traffic is also managed by the propagation layer by not transferring the blocks considered invalid by the receiving node. When a malicious node attempts to send an invalid block, the immediate peers who verify it first and reject to further propagate the block so that the entire network is relieved of the malicious new block request that tend to continuously flood the network. It allows only valid requests.

This thus prevents the DDOS attacks that is a problem of other distributed networks due to flooding of requests in the network, all valid or invalid, with no form of validation, such that any other service request is denied [22]. It provides stability to the system as well.

3.2.5. Consensus Layer

It is the base layer of a blockchain which defines how all the nodes must agree on one consistent state of the blockchain, thus concerns with the security of the network.

This is the layer that validates any proposed block and their transactions and decides on storing it into their local copy and propagate to the immediate peers. It also defines acceptance of one main chain should there arise a case where there are two or more chains being worked on by different groups of nodes.

As the proposed block with its proof - nonce and hash arrives from the p2p link into a node, it is verified here. Verification is simple by just using the block header and the nonce on a SHA-256 function, then comparing the obtained hash with the one received with the block. If it matches, the block is flagged as valid and is allowed to be propagated to other nodes. Otherwise, the block is flagged valid and the node rejects to propagate it further. This method of consensus is called the Proof-of-Work (PoW) protocol since a node consents a block is valid by recognizing the proof, i.e. nonce, that a certain amount of computational work has been done to determine that nonce rather than just blatantly adding a random value for the sake of incentive, which of course will be invalid.

Similarly, the transactions in the proposed block are also verified by simply using the Merkle tree's pairing and hashing algorithm, discussed earlier, on the obtained transaction list. The resulting root is then compared with the received Merkle root [17].

One important situation where a consensus needs to be addressed is when two or more miners produce valid results for one particular block around the exact same time on different parts of the network which often happens. The dilemma is deciding which block should the other nodes accept as the block for the main chain since both of them are valid but has different contents. In such a case, a fork is established for each valid block of the same position [23]. The forks are not yet a part of the absolute chain.

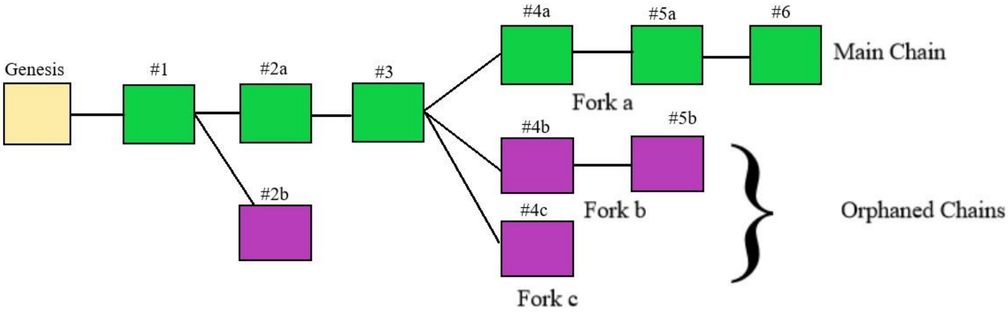


Fig 12 : Consensus by Forking and Longest Chain Method

Since each miner broadcasts the block to its immediate peers only (different for each fork), there will be the respective set of nodes working on what they received as the latest block. Peers and miners on the first fork mine on the next block for that fork after the dilemma block and so on for other forks. Thus, the number of nodes in the forks will not be the decider of the absolute chain, it goes beyond.

The next block for each fork is under construction. If any one miner of one of the fork solely creates a valid block first while no other miner has yet solved, it updates the chain in its fork. As soon as the network has that fork longer than all the other ones by one block due to aforementioned case and is being continually being propagated to majority of nodes first, the whole network must agree on working on that fork as the main chain and thus abandon mining for other forks. Thus a consensus is reached here.

In case, the block of next position is mined by two miners of same or different forks at the same time, the forks of these two valid blocks are taken and all other forks are abandoned. The next further block is mined for these forks and it goes on until at a time one fork immediately crops up with at least one block longer than all other forks. The longest chain is accepted as the main chain and other forks are left orphaned (hence called *Orphaned Chain*).

3.3. Implementation and Working

The Private blockchain incorporated in the architecture of the project is implemented as the flowchart show hereafter :

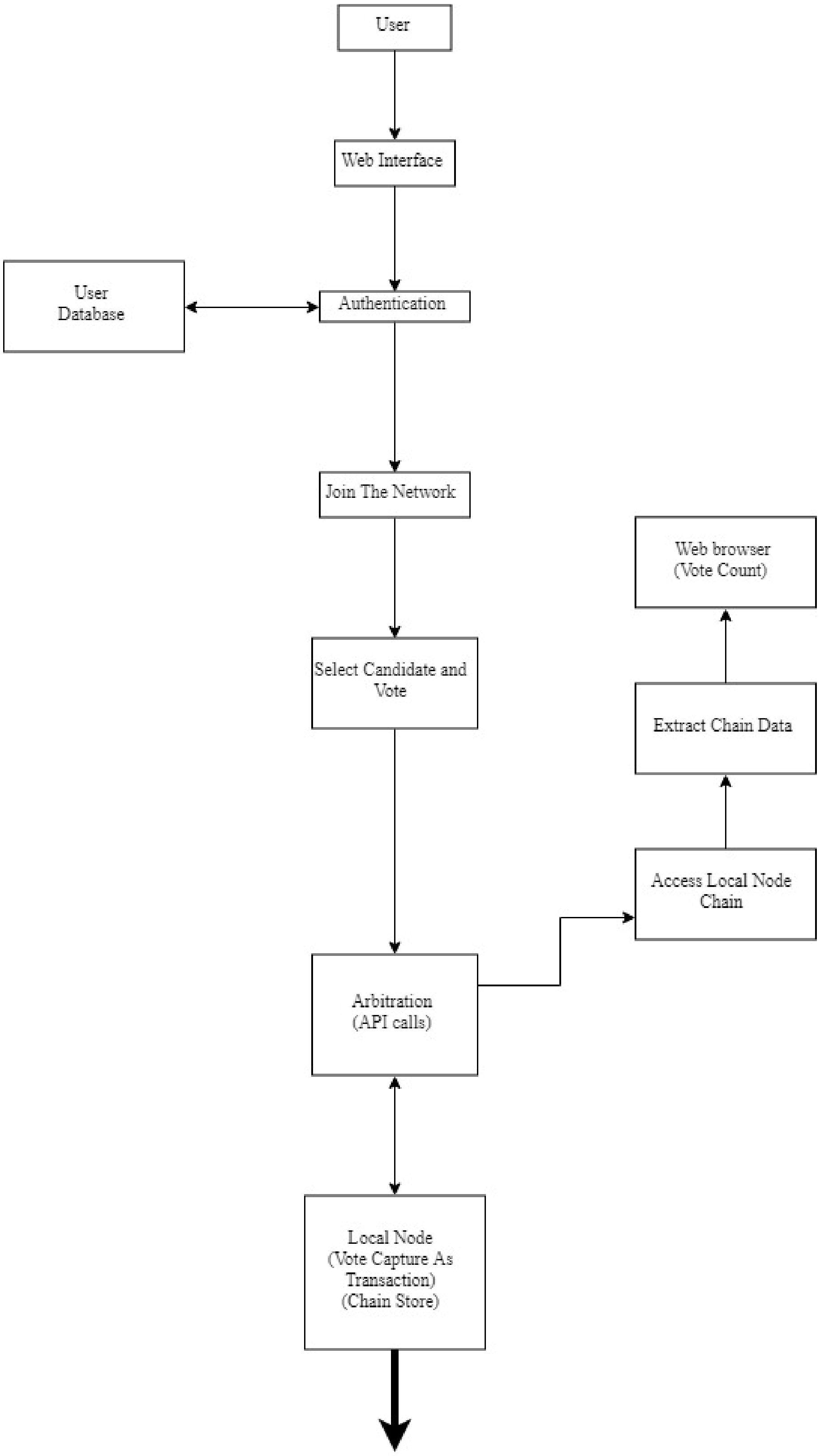


Fig 13 : FlowChart in Voting System Pt. I - Application Layer

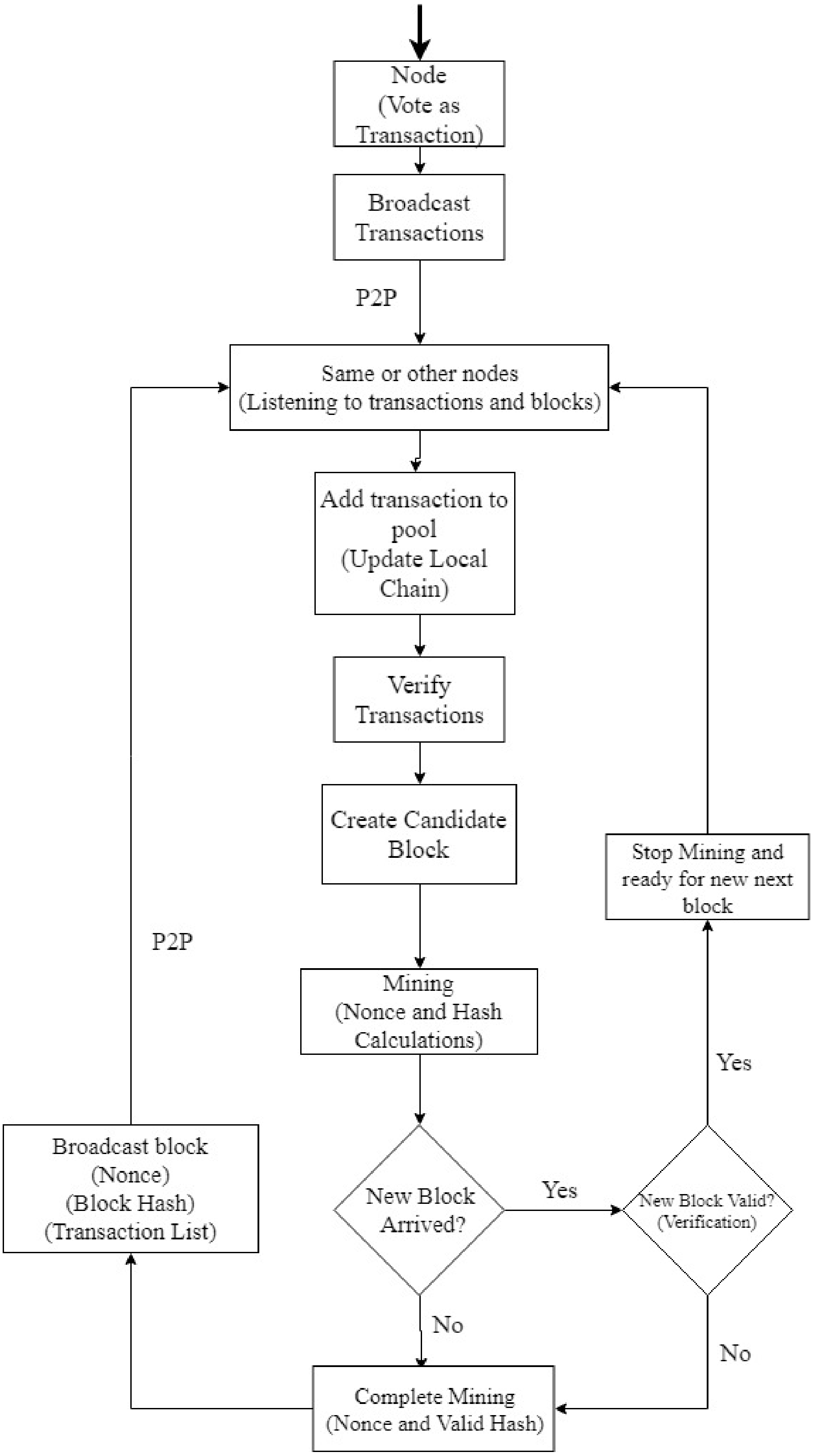


Fig 14 : FlowChart in Voting System Pt. II - Private Blockchain

At the time of voting, the user will have access to the client device already connected to the private blockchain network. In the browser interface page of the voting system, the user should enter their credentials where required. This project requires only their ID as designated to the users as they are registered when they are eligible. The authentication server cross-checks the credentials with the database to determine if the users are registered in the database. Only if they are registered in the database they are redirected to voting interface where they need to join the network as the user themselves, sort of like creating an account, where they are provided with a token to cast one vote. They can now select a candidate of their preference from a list of candidates and cast their vote. Once they do that, the token amount of the user is dropped to zero, meaning they can no longer vote for another candidate preventing double spending, until the system opens up another voting poll.

The arbitration server connects the vote with the local node, a part where blockchain begins, which is a device in the same polling station. The node is responsible for maintaining its local copy of the chain and providing the chain data to the application end when required. All the nodes, viz. at each polling station, here are full nodes that store the whole chain since we are concerned with a rather small application of blockchain.

The node captures the vote as a transaction with its parameters sender as the address of the user who voted provided to them during joining, recipient as the candidate’s address to whom the user voted and the amount as one token of vote. Thus this causes an increase in the tokens/amount of the candidate representing an increase in vote. The total votes received by the candidate can so be available in the blockchain network at any time. It is important to note that as soon as the vote is casted, the vote as a transaction will readily be a part of the Blockchain system in the node and not a part of the web application layer on the internet. The nodes then broadcast the transaction to the network by P2P communication as discussed in the previous section. The other nodes that continuously listen on the network and the node that broadcasted the transaction add the transaction to their ‘transaction pool’.

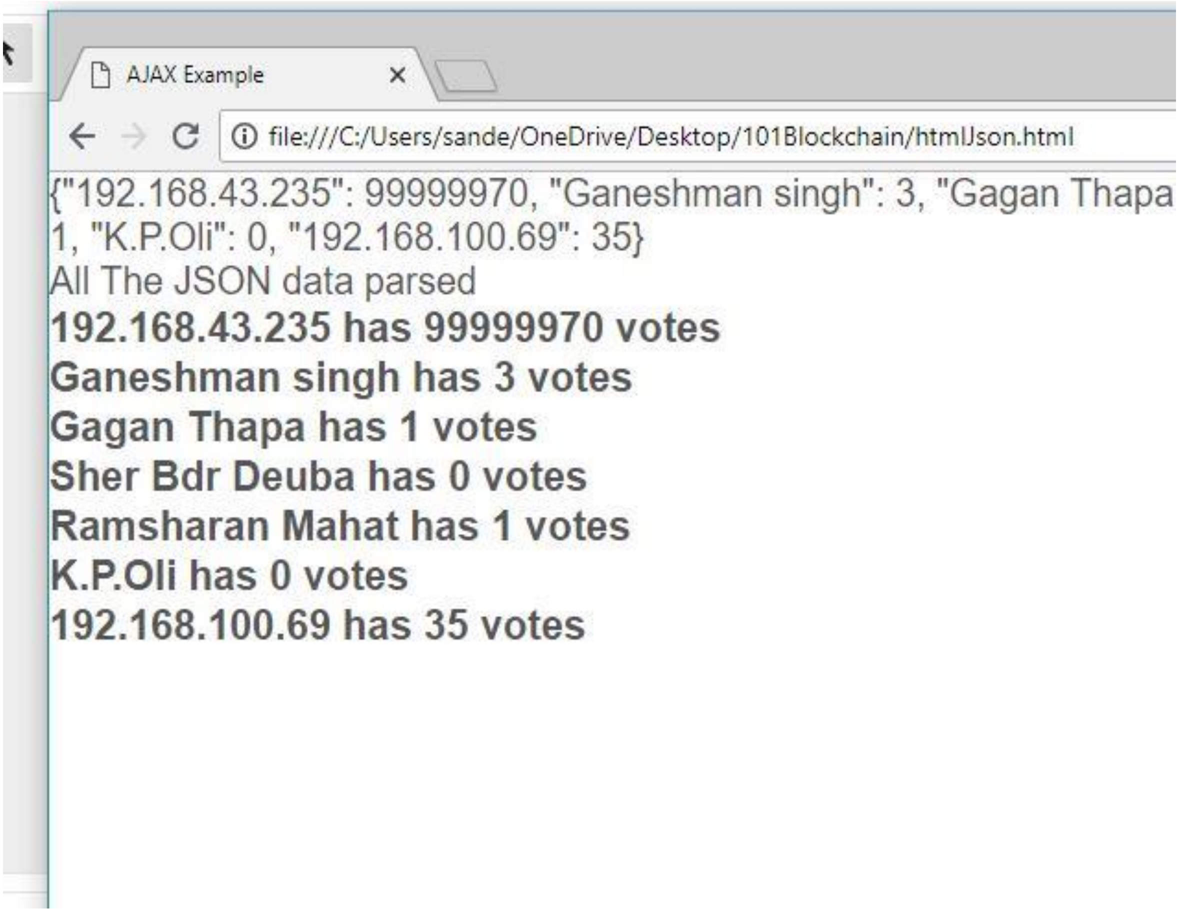
Before the nodes start mining for the latest block, they verify the validity of the transactions by checking the sender’s account and then only select them to be a part of the latest block. Then they start mining the block as discussed in section 3.2.3. They create a candidate block by evaluating all the block header components defined in mining layer and then start trying to solve for the nonce of the block. During mining however, if a new block of the position, that the node is working on, arrives from some other node claiming to be valid then the receiving node first verifies the block as defined in the consensus layer. If the block as well as the transactions in the block appears to be valid by hash comparisons, the node stops mining for block of that position and immediately gets ready for mining the next block since transactions would have been accumulated during that lost time as well. It updates its local copy of the chain and begins mining process again.

In case the new block is found to be invalid (they are rejected by the receiving node) or no blocks arrive by the time the node finishes mining for the current block, the node presents the proof of their work and broadcasts the block with it and the selected transaction list to the network and starts mining for the next block. The valid blocks are distributed and accepted by other nodes in the system with the consensus protocols and the chain expands with the introduction of new blocks again and again.

This project has, instead of using Merkle tree to represent the transactions, used the transactions themselves as part of the candidate block to be hashed directly with the other components.

The miners for this system are the concerned authority at each of the polling station who can only manage the blocks and accept valid one but not alter any of it due to the difficulty in altering the latest block let alone any of the previous blocks (section 3.2.3.).

This system importantly allows for efficient display of the results. It can even be in real time since the chain is readily available in the nodes. The local copy of the chain is accessed from the local node via the arbitration server and the result can be viewed in candidate by candidate basis or of the candidates at once in the browser interface. The vote is cast as the new block containing the vote transaction is found valid.



AJAX Example
file:///C:/Users/sande/OneDrive/Desktop/101Blockchain/htmlJson.html

```
{"192.168.43.235": 99999970, "Ganeshman singh": 3, "Gagan Thapa
1, "K.P.Oli": 0, "192.168.100.69": 35}
All The JSON data parsed
192.168.43.235 has 99999970 votes
Ganeshman singh has 3 votes
Gagan Thapa has 1 votes
Sher Bdr Deuba has 0 votes
Ramsharan Mahat has 1 votes
K.P.Oli has 0 votes
192.168.100.69 has 35 votes
```

Fig 15 : Vote Tally

EPILOGUE

4.1. Result

The main objective of this project was to build a blockchain network from scratch and implement voting application on top of it. The main components of blockchain technology that are Decentralization, Immutability, Consensus Mechanism, Distributed ledger(list of transaction) etc. were all implemented in the project successfully. And, a simple voting application was built on top of the blockchain network utilizing the essential components of blockchain to produce a secure and tamper proof voting system.

4.2. Problems Faced

During the project development various technical problems were faced; some of the problems were compromised due to technical difficulty, while other problems were solved by implementing different technologies available.

The blockchain technology being completely new to the project members, a thorough study of technology required a significant time of given project duration.

The blockchain was hard coded using Python, thus while performing I/O operations (or read/write) to blockchain a specific problem called GIS occurred which caused I/O threads to be scheduled. Ultimately limiting the execution of our I/O operations. The solution to the problem was building the entire blockchain using other languages like GO, but then due to time constrictions we had to compromise on this specific problem.

4.3. Limitations

The private Blockchain built in this project is still a rudimentary and due to this topic still being evolved, it cannot be implemented in serious applications like conducting a national level voting system directly. Thus its application is limited to conducting voting at our own personal level.

4.4. Scope and Enhancements

The voting project was implemented on top of private blockchain. This project can be enhanced by building the voting application on top of public blockchain for example Bitcoin Blockchain, Ethereum Blockchain etc. This allows to create a complete autonomous system without a major authority controlling some specifications of systems.

Blockchain in itself is a exciting platform, it can be implemented in different areas beyond voting. In financial sectors the idea of cryptocurrency, in health sectors the idea of storing patients data in blockchain to avoid data breach, in IOT the idea of connecting devices securely etc. are easily feasible with blockchain technology.

4.5. Conclusions

At first, research on this project when deciding on taking it forward took us in several directions until getting acquainted with the various aspects of Blockchain. The unfamiliarity with a bunch of these aspects made the project take off at a slow pace during the project execution. But over time, we were able to complete the project within the targeted scheduled time while attaining the objectives set out at project planning.

There is still more to be done to achieve a more refined version of this project with additional functionalities. Blockchain is a still growing technology with yet a large amount of unexplored aspects that are yet to be explored and integrated in various other applications.

4.6. References

[1] Rabin Sharma Lamichhane, "Bitcoin is illegal in Nepal says Central Bank of Nepal, Bans On Any Transactions", Rabinsxp, 2017. [Online]. Available: https://www.rabinsxp.com/news/bitcoin-illegal-nepal-says-central-bank-nepal-bans-transactions/. [Accessed: 12-Feb-2018]

[2] M. Hogan, "History Of Elections," Duval Elections. [Online]. Available: https://www.duvalelections.com/General-Information/Learn-About-Elections/History-Of-Elections. [Accessed: 18-Feb-2018].

[3] R. F. Celeste, D. Thornburgh, and H. Lin, "Asking the right questions about electronic voting". Washington, D.C.: National Academies Press, 2006.

[4] A. B. Ayed, "A Conceptual Secure Blockchain Based Electronic Voting System," International Journal of Network Security & Its Applications, vol. 9, no. 3, pp. 01–09, 2017.

[5] D. Pollock, "Blockchain For Elections: Advantages, Cases, Challenges," Cointelegraph, 17-May-2018. [Online]. Available: https://cointelegraph.com/news/blockchain-for-elections-advantages-cases-challenges. [Accessed: 5-Jun-2018].

[6] D. W. Jones, "Voting on Paper Ballots," Douglas W. Jones on Ternary Computing, 15-Apr-2012. [Online]. Available: http://homepage.divms.uiowa.edu/~jones/voting/paper.html. [Accessed: 10-Feb-2018]

[7] S. P. Everett, "MEASURING THE USABILITY OF PAPER BALLOTS: EFFICIENCY, EFFECTIVENESS, AND SATISFACTION," issue brief, Department of Psychology, Rice University, Houston, TX, 2006.

[8] M. Blaze, "Hearing on Cybersecurity of Voting Machines." pp. 11–19, 2017.

[9] Wikipedia, "Electronic voting in Estonia," Wikipedia, 01-Aug-2018. [Online]. Available: https://en.wikipedia.org/wiki/Electronic_voting_in_Estonia. [Accessed: 12-Apr-2018].

[10] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman, "Security Analysis of the Estonian Internet Voting System." Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. (2014), pp. 703-715.

[11] S. Shah, Q. Kanchwala, and H. Mi, "Block Chain Voting System", Northeastern University, 2018.

[12] Lisk Academy, "What is Blockchain? » Start Learning | Lisk Academy", Lisk. [Online]. Available: https://lisk.io/academy/blockchain-basics/what-is-blockchain. [Accessed: 2-Mar-2018]

[13] D. Xiao, "The Four Layers of the Blockchain", Medium, 22-Jun-2016. [Online]. Available: https://medium.com/@coriacetic/the-four-layers-of-the-blockchain-dc1 376efa10f. [Accessed: 27-Sep-2018].

[14] B. Singhal, G. Dhameja, and P. S. Panda, Beginning Blockchain: a beginners guide to building Blockchain solutions. New York: Apress, 2018, pp. 17-22.

[15] D. Cosset, "Blockchain: What is Mining?," The Practical Dev, 05-Jan-2018. [Online]. Available: https://dev.to/damcosset/blockchain-what-is-mining-2eod. [Accessed: 11-Feb-2018].

[16] S. Jimi, "Blockchain: how mining works and transactions are processed in seven steps," Medium, 02-May-2018. [Online]. Available: https://medium.com/coinmonks/how-a-miner-adds-transactions-to-the-blockchain -in-seven-steps-856053271476. [Accessed: 12-Feb-2018].

[17] Chainthat, "Blockchain Basics Explained - Hashes with Mining and Merkle trees", YouTube 07-Feb-2016. [Online]. Available: https://www.youtube.com/watch?v=lik9aaFIsl4. [Accessed: 15-Mar-2018].

[18] P. Bajpai, "Block (Bitcoin Block)," Investopedia, 08-Aug-2018. [Online]. Available: https://www.investopedia.com/terms/b/block-bitcoin-block.asp. [Accessed: 15-Feb-2018].

[19] A. Singh, "What is a Genesis Block? | Blockchain Semantics Blog", Blockchainsemantics, 06-Apr-2018. [Online]. Available: https://www.blockchainsemantics.com/blog/genesis-block/. [Accessed: 16-Feb-2018].

[20] Blockchain Semantics, "Bitcoin Blockchain Nodes – Types | Blockchain Semantics Blog", blockchainsemantics, 31-Jul-2018. [Online]. Available: https://www.blockchainsemantics.com/blog/nodes-bitcoin-blockchain/. [Accessed: 18-Feb-2018].

[21] D. Cosset, "Blockchain: what is in a block?," The Practical Dev, 2018. [Online]. Available: https://dev.to/damcosset/blockchain-what-is-in-a-block-48jo. [Accessed: 11-Feb-2018].

[22] Blockgeeks, "Consensus and Mining on the Blockchain", YouTube, 30-Jan-2018. [Online]. Available: https://www.youtube.com/watch?v=2HcmwfVzPEU. [Accessed: 14-Mar-2018].

[23] MobileFish, "Blockchain tutorial 24: Blockchain and miners", YouTube, 25-Jun-2017. [Online]. Available: https://www.youtube.com/watch?v=s7-8a71KgT4. [Accessed: 20-Apr-2018].

[24] D. V. Flymen, "Learn Blockchains by Building One – Hacker Noon," Hacker Noon, 24-Sep-2017. [Online]. Available: https://hackernoon.com/learn-blockchains-by-building-one-117428612f46. [Accessed: 25-Feb-2018].