A PROJECT REPORT ON

“Enhancing Traditional Voting Systems Using Hyperledger Technology”

SUBMITTED BY

Om Lachake (BI36)
Jitwar Kurhade (BI35)
Pranay Malusare (BI42)
Manas Tole (BI74)

GUIDED BY

Prof. Sonali Patil

DEPARTMENT OF INFORMATION TECHNOLOGY

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, PUNE - 411057

SAVITRIBAI PHULE PUNE UNIVERSITY

2022 – 2023

A Project Report On

“Enhancing Traditional Voting Systems Using Hyperledger Technology”

submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

INFORMATION TECHNOLOGY

Submitted By

Om Lachake (BI36)
Jitwar Kurhade (BI35)
Pranay Malusare (BI42)
Manas Tole (BI74)

Under the guidance of

Prof. Sonali Patil

DEPARTMENT OF INFORMATION TECHNOLOGY

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,

HINJEWADI, PUNE-57

2022 – 2023

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HINJEWADI, PUNE-57

DEPARTMENT OF INFORMATION TECHNOLOGY

Certificate

This is to certify that the project report entitled

“**Enhancing Traditional Voting Systems Using Hyperledger Technology**”

which is being submitted by

Om Lachake (BI36)
Jitwar Kurhade (BI35)
Pranay Malusare (BI42)
Manas Tole (BI74)

have partially completed the Project entitled” Enhancing **Traditional Voting Systems Using Hyperledger Technology**”, under my guidance in partial fulfillment of the requirement for the award of the Bachelors of Engineering in Information Technology of International Institute of Information Technology, Hinjewadi, by Savitribai Phule Pune University for the academic year 2022 – 2023.

Prof. Sonali Patil
Internal Guide
Department of Information Technology

Dr. Jyoti Surve
Head of Department
Department of Information Technology

External Examiner:

Dr. Vaishali Patil
Principle

Date:

Acknowledgement

We would like to express our special thanks to our internal guide Prof. Sonali Patil as well as Dr. Jyoti Surve, Head of Department who gave us the golden opportunity to do this project on the topic “**Enhancing Traditional Voting Systems Using Hyperledger Technology**”, which also helped us in doing a lot of research and I came to know about so many new things.

While doing this project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much pleasure. We would like to show our gratitude Prof. Sonali Patil, Seminar Coordinator for giving us good guidelines for the project throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in completing this project. Lastly, we also extend sincere thanks to all the staff members of the Department of Information Technology and Dr. Vaishali V. Patil, Principal for helping us in various aspects.

(Students Name & Signature)

Om Lachake	
Pranay Malusare	
Manas Tole	
Jitwar Kurhade	

Contents

List of Figures

List of Tables

Abstract

Being the largest democracy in the world, India has an abundance of different cultures, religions, languages, and beliefs. But unity in diversity comes with its own costs. It is a challenge to conduct nationwide elections for such a big and diverse population. The recent upsurge in mistrust towards electronic voting machines and trends like voter apathy where the number of people showing up to cast their vote dwindles each year is adding to the challenge of conducting fair elections. Electronic Voting Machines lack auditing capabilities and require sealing, police protection, and large logistic efforts to make sure they are not tampered with. This huge logistical and security overhead adds to the cost and variables of the already delicate electoral system.

A growing list of records that are linked using the cryptographic hash of previous blocks, is called a blockchain. Blockchain technology allows data to be captured in real-time in a secure and encrypted manner and ensures that the data is tamper-proof.

Bitcoin and other cryptocurrencies are the most common examples of blockchain behavior. This technology can be used to deploy a platform that provides maximum transparency and reliability to build a trustful relationship between voters and election authorities. The web-application will provide a framework that can be implemented to conduct voting activity digitally through blockchain. A voter will receive a single voting token that can be used to vote and will be unable to participate in the process again as the token has been utilized. Smart contracts are used to provide a secure connection between the user and the network while executing a transaction in the chain.

By providing an irrefutable and easy way to vote from one's phone or pc, the number of people voting will likely rise. Furthermore, it will decrease the logistics and security needed for various polling booths as the records will be stored on the blockchain in real-time, unlike traditional methods that need to be sealed and transported to secure locations for the counting of votes.

Chapter 1

Introduction to Project Topic

1.1 Overview

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system. A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralized database managed by multiple participants is known as Distributed Ledger Technology (DLT). Blockchain is a type of DLT in which transactions are recorded with an immutable cryptographic signature called a hash.

The project aims to implement a decentralized voting system, which will encourage a tamper-proof and anonymous election process. Each block will contain all the processed votes with anonymous tracking ability to maintain voter privacy. The candidates as well as the voters will have a wallet, and in those wallets, voting tokens will be assigned, these are non-fungible and will follow a single-direction flow. The flow of these tokens will be from the voter's wallet to the candidate's wallet. Candidates won't have access to their wallets and will only be allowed to access the polling ends. This process is defined and further described in the upcoming sections.

1.1.1 Project Overview

Blockchain technology is a very prominent technology of 2022, being an important part of Web3. The project focuses on taking full advantage of all features of blockchain technology and bringing them into effect and helping the currently a little shabbier election system by making it into a flawless system where votes would be cast digitally using the web application and will be delivered to the candidate wallet immediately using the cutting edge Hyperledger fabric technology provided by a blockchain. The system will consist of a Model-View-Controller Structure that will implement microservices to create modules and facilitate the separation of concerns. A front will be generated using static site generators such as Next JS which will interact with a middleware implemented in Node JS to interact with the Fabric network. The hydration of dynamic data will be completed using REST APIs. Each participant, i.e., Voters, and Candidates will have their own node hosted. These nodes will be used to process the transactions. These nodes will be hosted on AWS or Google Cloud in containerized servers.

1.2 Brief Description

Hyperledger fabric is used to implement a blockchain network. Hyperledger Fabric is intended as a foundation for developing applications or solutions with a modular architecture. Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play. Its modular and versatile design satisfies a broad range of industry use cases. It offers a unique approach to a consensus that enables performance at scale while preserving privacy.

Taking advantage of Byzantine Fault Tolerance-based algorithms, the consensus in Fabric runs in three steps, - Ordering, Endorsing, and Validation. This is not only faster than those suggested in Hyperledger Architecture Volume 1[10], but also provides better speed and finality. Thus, the aim is to create a service that does not throttle the election process while ensuring the scalability was satisfied by this technology. Storing the wallet for the user, using CouchDB (inbuilt functionality) will be a must as the intended user won't store the keys themselves.

1.3 Problem Definition

Conducting elections has been a very complex process right from the start to declaring the results. The proposed system aims to fill the enormous gap between a voter and the election process and make it a very smooth process for voters to cast their votes through a permissioned blockchain network, Hyperledger Fabric, with the help of smart contracts. The integrity of the votes is maintained and can be tracked if needed without exposing the voter. The system is able to process concurrent transactions without the possibility of a delay.

1.4 Applying Software Engineering Approach

After surveying research papers in Phase 1 of the Final Year Project, the technology for the problem statement has been decided. The project will be developed in a modular architecture so as to provide separation of concern and simultaneous development environments ensuring faster delivery. This modular approach will not only provide a better developmental experience but also aims to be scalable and more maintainable in future. The deployments will be summarized by using services such as Docker and Kubernetes. The main goal is to make this project deployable anywhere and not vendor locked on cloud.

Chapter 2. Literature Survey

As the block chain world is quite fresh and ripe, direct papers based on this problem statement were scarce. Although there are many other implementations of different distributed ledger solutions such as Kaleida, Ethereum, there were very few for Voting Mechanisms. The literature available is not much and mostly about analysis of existing blockchain. The first blockchain Paper published by Satoshi Nakamoto [12], defines the working of Bitcoin. It specifies the differences between distributed network and centralized network. It was the start for research related to decentralized solutions.

The Traditional Voting system [18] works as follows. An EVM is designed with two units: the control unit and the balloting unit. These units are joined together by a cable. The control unit of the EVM is kept with the presiding officer or the polling officer. The balloting unit is kept within the voting compartment for electors to cast their votes. This is done to ensure that the polling officer verifies your identity. With the EVM, instead of issuing a ballot paper, the polling officer will press the Ballot Button which enables the voter to cast their vote. The voter can press the button next to the candidate's name they wish to vote for.

According to Pawlak and Poniszewska-Marańda [14], a system with hybrid implementation of voting, where the voting machines need to be physically accessed, is suggested to bring up a huge change. The votes then are sent by the machines through an internet connection, which include a unique number for each voter. The approved transactions are processed onto the block and a ballot is printed for the voter, this ballot is then deposited into a box at the booth just for the purpose of extra security, which can be considered for validation of the blockchain transactions. The voters will receive a unique token, which they can use to validate their vote. The value of the vote will not be accessible but only its existence can be verified.

The paper motivates a hybrid way of conducting elections by utilizing the blockchain. It provides a brief background of bitcoin and then states the faults of a traditional voting system. It also lists the cons of existing solutions to e-voting such as voter identification and authentication. It takes in the implementation of the voting process in Poland and enhances it.

Muhammad and Usman [1], suggest that their framework provides a complete digital voting system with a flexible consensus algorithm. Their system included smart contracts that provided traceability of the vote, with a locking mechanism to prevent further transactions. The concept of UTXO, Unspent Transaction Outputs, was the main algorithm ensuring the voting process. This paper is the base for this implementation.

Case studies [17] concluded that through usage of Fabric for implementing block chain solutions, integrity of their transactions increased highly. Tech Mahindra implemented a land registry documentation for the government of Abu Dhabi which provided increased security and faster process with a single source of truth. All the records were re-collectable and trackable.

Fabric being a growing technology with increased community support, their adaptations into governmental solutions are rapid. In March 2019, it funded a project to create LACChain, a permissioned public blockchain infrastructure. LACChain implemented protocols for self-sovereign identity that are inclusive and respectful of personal data. The project is now building interoperable sidechains and layer-2 networks to be able to scale as demanded. After December 2023, when the IDB Lab project phase winds down, the LACChain networks will remain sustainable, strong and orchestrated by LACNet. LACChain won't be limited to the LAC region, either. Already operating nodes exist in the United States and Europe.

Table 1 summarizes the Literature survey of Voting Systems based on Blockchain technology.

Sr. No	Paper Title	Algorithm	Result	Observations
1.	Voting Using Blockchain Technology (2022) [16]	Uses a one-time password for authentication. One Voting coin (VC) is allotted to each voter	Ensuring the voter's anonymity and providing secure authentication.	Dependent largely on mobile technology.
2.	Blockchain: Research and Applications (2022) [15]	Uses ECC algorithm on Ethereum network	A novel blockchain-based voting scheme for IoE system	Not suitable for large-scale voting and parallel processing not supported.
3.	S K Geetha (2021) [5]	Simple and standard implementation of blockchain technology for e-voting	Reduced operational cost. Increased security of the voting process	Is suitable only for online voting. Using a centralized server creates a single point of attack against voting data.
4.	A Framework to Make Voting System Transparent Using Blockchain Technology [1]	Using blockchain and SUS score analysis	Prevent the spread of the Covid19 pandemic at a low cost by high system security and accuracy voting system	Blockchain public network to process every vote transaction that takes 10 to 20 seconds
5.	Polya's 2015.[7]	ECC	Online-Voting. High security and data protection	Suffer in scalability
6.	Luxoft's E-Voting Platform Enables First Consultative Vote based on Blockchain in Switzerland [8]	ECC/El Gamal	The first customized blockchain electronic voting system used by a significant industry.	Scalability and latency issues

Table 1. Literature survey

Chapter 3. Software Requirements Specification

3.1 Introduction

Implementation of blockchain technologies exists in varied ways, the most prominent one being crypto-currencies [19]. Ranging from Public based block chain to Privatized network, one can choose from a glut of available choices. The main purpose to keep track of votes while maintaining anonymity was fulfilled by Hyperledger Fabric. This implementation by the Linux Foundation provides a way to privatize the blockchain with heavy control over the network. It allows enforcing policies and rules over the network. It also provides permissioned flow of transactions. Not only it keeps the entire world state, but also provides private routes between two groups of nodes, known as Organizations, to ensure the translation integrity. This channel maintains a separate ledger of transactions that occurred within its limits.

The intended audience is a normal everyday user who does not understand any of the above. Thus, abstracting this all, the intent is to implement a web application that follows Web3 standards. This will promote the usage of digital voting allowing everyone to fully take part in the election process. The web application must consist of minimum dynamic content and maximum static pages, prebuilt during the hydration. This led us to choose Next JS for the implementation.

Fabric provides different SDKs to interact with the network, as it is written in Go, the languages supported are Java, JavaScript (Node JS), Go. Although most of the commands are supposed to be executed through a shell, these SDKs provide a great abstraction. Node JS is a JavaScript runtime that allows development of server-side applications in JavaScript. Express JS is a framework that aids in development of REST APIs which will be used to further expand the application.

For deployment, native docker support is provided by Fabric by providing all the necessary containers. IBM and AWS provide Fabric services with great ease. The deployment of the application is planned in a containerized mode which is independent of any cloud service provider. Using Docker, an open-source tool for containerizing applications, anyone should be able to host this application.

3.2 System Feature

The system consists of three tiers of entities:

1. Departments
2. Voters / Candidates
3. Channels

Departments are just grouped peers, i.e., Voters and Candidates. One department will only consist of Candidates. All the other departments will be location-based voting groups, in which voters will be organized into respective departments. To interact, each peer must be in one department.

The departments will be connected through a single channel. This channel is nothing but a ledger

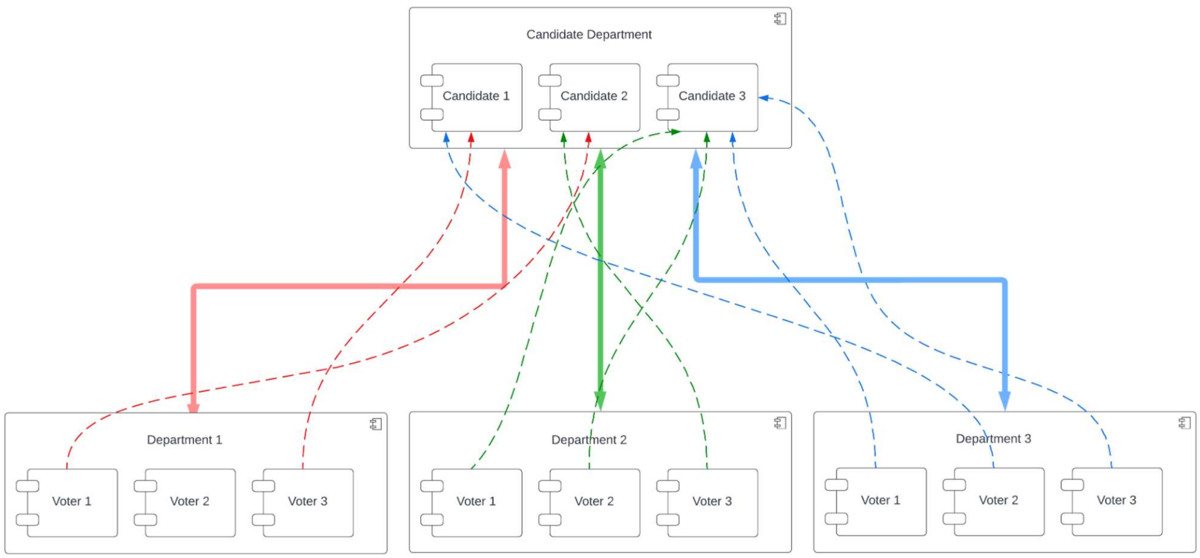

Figure 1. Proposed System

that records transactions between those two departments. This acts like a little private record, which
other departments in the network cannot access. Channels will consist of one Orderer node. The job of this Orderer peer is to queue the incoming transactions. This will ensure asynchronous processing of transactions in the network. Although a channel in Fabric is bidirectional, rules can be imposed to force unidirectional flow.

A view of the system is provided below. Here there are 3 Voter Departments, and as maintained, 1 Common Candidate Department. There are three different channels, red, green and blue for respective couplings. The voter tokens will pass through the specified routes only. But a common world state will record all of these transactions, which is immutable.

If some Candidate is eligible for a specific Voter Department, it will be handled outside of the network using metadata stored on databases to provide the voter with only those candidate options which were decided at the time of election nominations.

The Process of voting will start by nomination of the candidates. The Election Committee (EC) will be responsible for setting deadlines and administering the election. Creation of departments will also be under EC’s control. The voter will need to be registered with the election committee before

starting out on the platform. After the first login, voters will need to physically get verified. The physical locations will be defined in the metadata. During the verification process, once the moderator passes the checks, voting tokens will be transferred to the voter's wallet. These tokens will expire after the deadline for each election. After the nomination process ends, the polling duration will start.

During the polling, voters will be shown eligible elections and candidates that one can vote for. The wallet will be active from polling start till the election end. While casting the vote, a smart contract will be executed that will transfer the token from the voter's wallet to the intended candidate. This transaction will return a VIT (Vote Identification Token) that will allow the user to validate the vote's existence. The Smart Contract for results, will collect the tokens for each candidate while simultaneously going through the world state recollecting the transactions.

3.3 External Interface Requirements

3.3.1 User Interfaces

A user interface for three entities, Voter, Election Committee and Candidate (To View the result) will be implemented using Tailwind CSS. The intended design is given below.

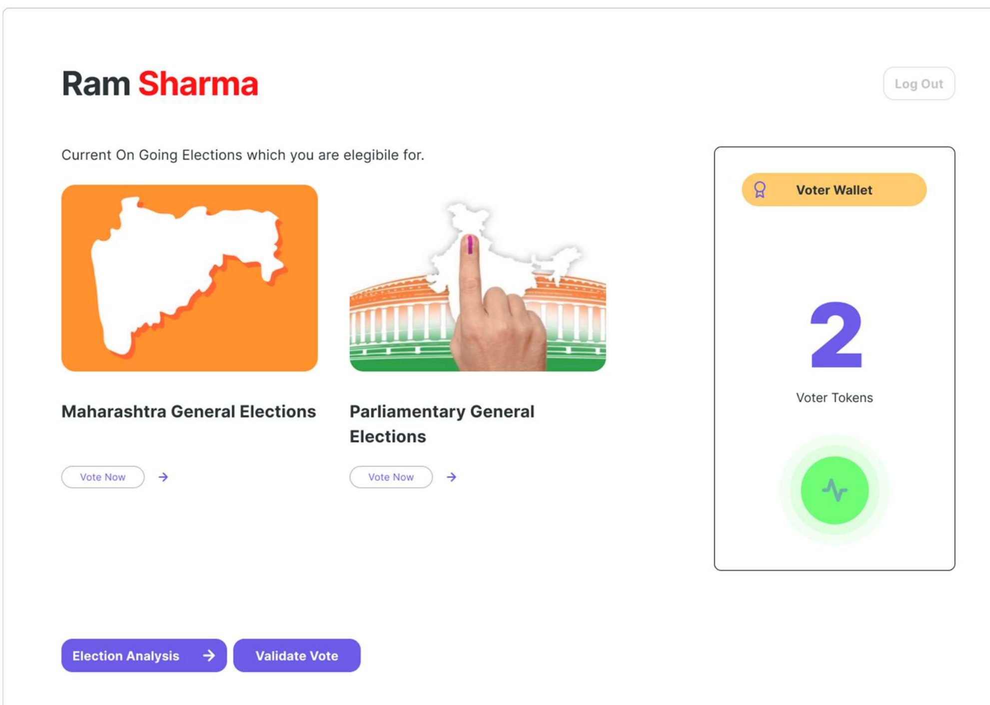

Figure 2. Voter Dashboard

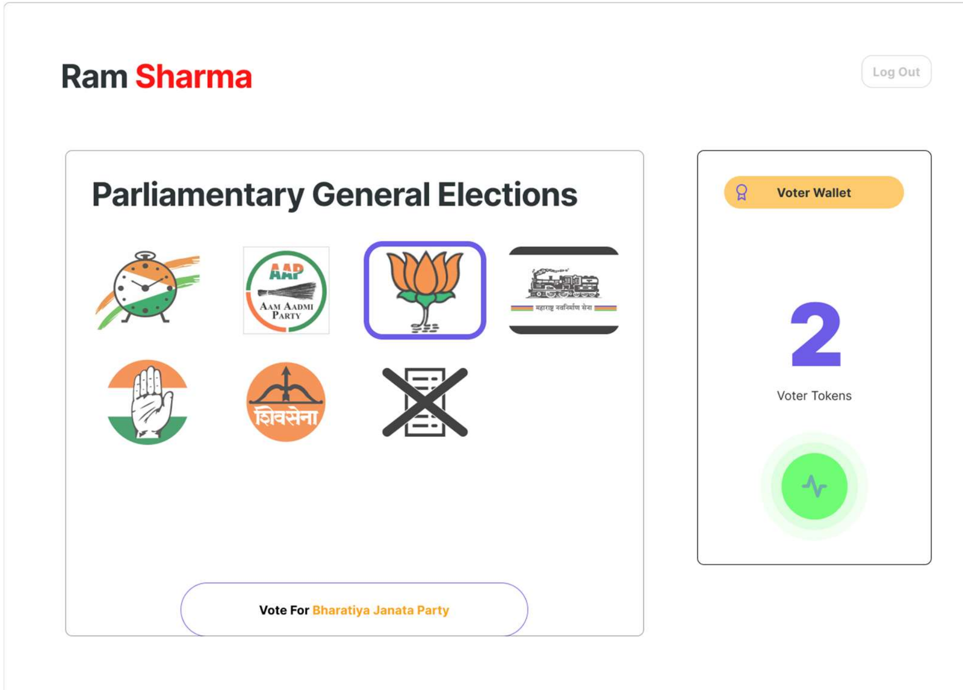

Figure 3. Candidates Choices

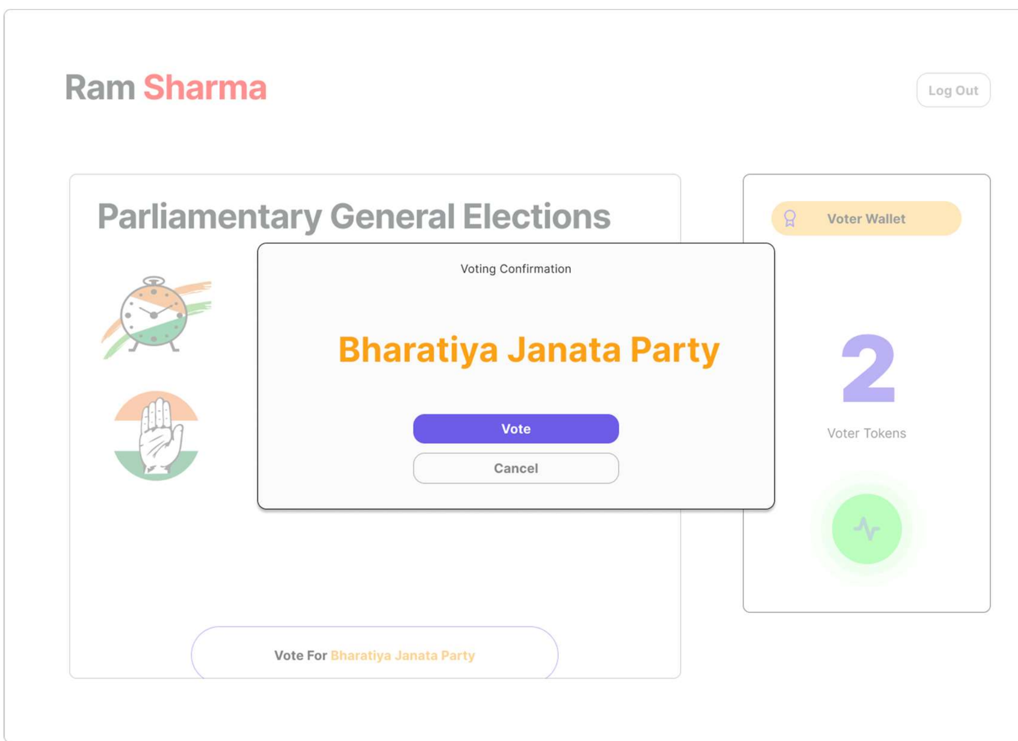

Figure 4. Vote Confirmation

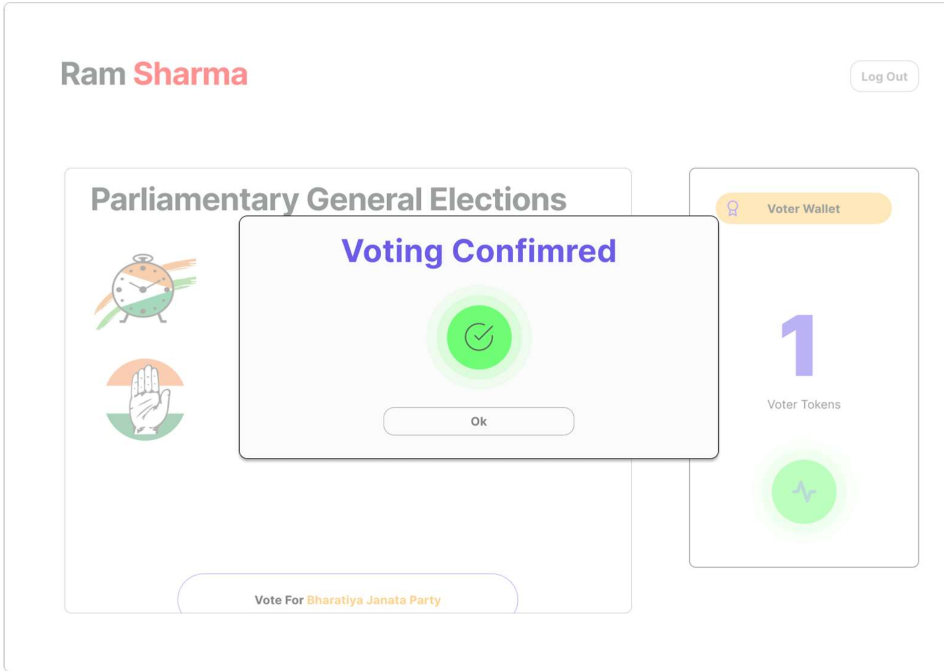

Figure 5. Vote Successful.

3.3.2 Hardware Interfaces

Voters, Candidates and the Election Commission will require nothing but a computer or a mobile device with internet connection and a web browser to use the portal. As the platform will be hosted on the cloud, it will support all types of devices which satisfy the above rules. This will allow each and every user to utilize the service.

3.3.3 Software Interfaces

All the stakeholders will be able to interact and partake in the process through a web app. The app can

3.3.4 Communication Interfaces

A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding. REST is a set of architectural constraints, not a protocol or a standard. API developers can implement REST in a variety of ways. In REST API, it transfers a representation of the state of the resource to the requester or endpoint.

When a client request is made via a RESTful information, or representation, is delivered in one of several formats via HTTP: JSON (JavaScript Object Notation), HTML, XLT, Python, PHP, or plain text. JSON is the most generally popular file format to use because, despite its name, it's language-agnostic, as well as readable by both humans and machines. Express JS will allow us to define REST APIs and process the network transactions

3.4 Non-Functional Requirements

3.4.1 Performance Requirements

Accuracy: A system is considered accurate if it is not possible to alter a vote, discount validated vote from the final tally or include an invalid vote in the final count.

Privacy (un-traceability): this is ensured by a system that prevents any agency from linking a specific voter with the ballot he cast, and does not allow voters to prove the way they voted.

Individual and universal verifiability: individual verifiability refers to the fact that any individual can verify that his vote was received properly while universal verifiability allows a voter to verify that all votes have been counted properly. Verifiability contributes to the public trust in e-voting systems.

3.4.2 Safety Requirements

Authentication: It requires voters to be uniquely identified in a way that unmistakably distinguishes them from other people.

Integrity: It means that a voter's intention shall not be affected by the voting system, or by any undue influence.

Privacy: It implies that each vote remains confidential and that the anonymity of the voters is preserved.

3.4.3 Security Requirements:

Voter Anonymity: Ensure that votes must not be associated with voter identity.

System Integrity: Ensure that the system cannot be re-configured during operation.

Data Integrity: Ensure that each vote is recorded as intended and cannot be tampered with in any
manner, once recorded (i.e., votes should not be modified, forged or deleted without detection).

Secrecy / Privacy: No one should be able to determine how any individual voted.

Non-coercibility and No Vote-selling: Voters should not be able to prove to others how they voted (which would facilitate vote selling or coercion).

Reliability: Election systems should work robustly, without loss of any votes, even in the face of
numerous failures, including failures of voting machines and total loss of network communication. The system shall be developed in a manner that ensures there is no malicious code or bugs.

Availability: Ensure that system is protected against accidental and malicious denial of service attacks. Also, setup redundant communication paths so that availability is ensured.

3.4.4 Software Quality Attributes

Integrity: Blockchains remain fully open and accessible to everyone. Thanks to the transparency of the blockchain, it is possible to easily track transaction flows. If the identity behind a wallet address is known, then the transactions made can be traced back and traced in the future. All these transactions can be viewed in detail. But as per our implementation, we would keep the voter completely anonymous, and make sure that the transaction will contain no data about the voter other than the voter hash. If in any case the transaction seems null or fake, the election committee, when provided the access, will be able to create a track of the vote and validate the voter.

Security: As the Fabric Network will imply heavily administered policies regarding the transaction flow, the voter will have a restricted access to the respective candidates. Each Department will have its own private channel to the Candidate Department, thus blocking the voters from the illicit department to vote for unpermitted candidates. The channels will contain private ledgers that will record all the transactions.

Concurrency: As there will be multiple channels, each channel will consist of an Orderer node. This will allow simultaneous transactions throughout the network. All these transactions will be recorded on their channel respective ledgers and also on a common world state. Different web applications can be permitted to transact on different channels, thus again reducing the server load for our clients. Networks can be localized to provide more safe and concurrent experience, which will later on report the status of all those networks combined to a master server.

3.5 Analysis Model

3.5.1 Data Flow Diagram

- Voting Transaction

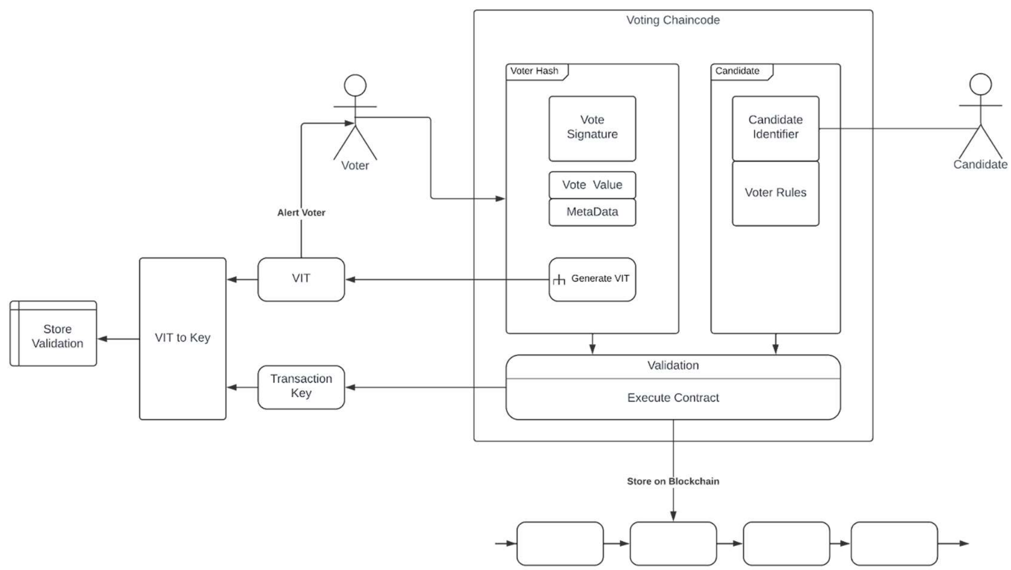

Figure 6. Vote Transaction Flow

- Vote Calculation

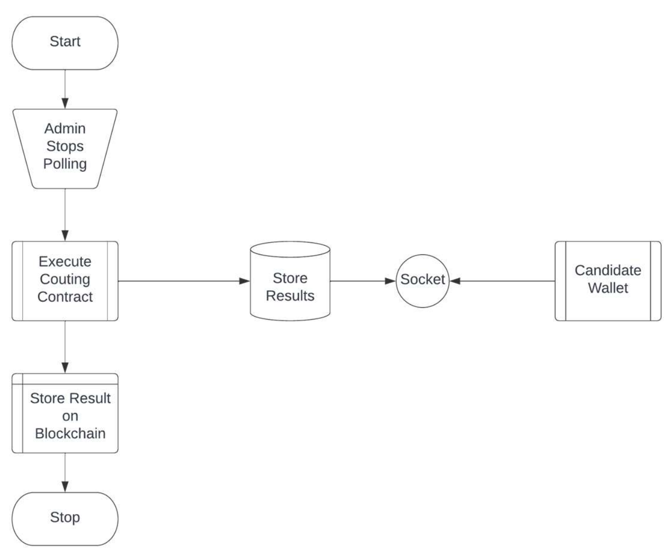

Figure 7. Voting Result Calculation

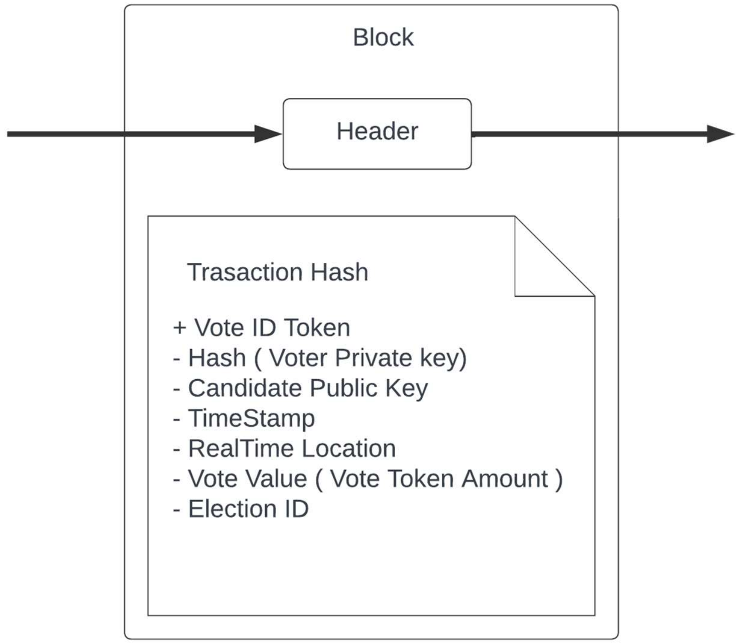

Figure 8. Transaction Data

3.5.2 Entity Relationship Diagram

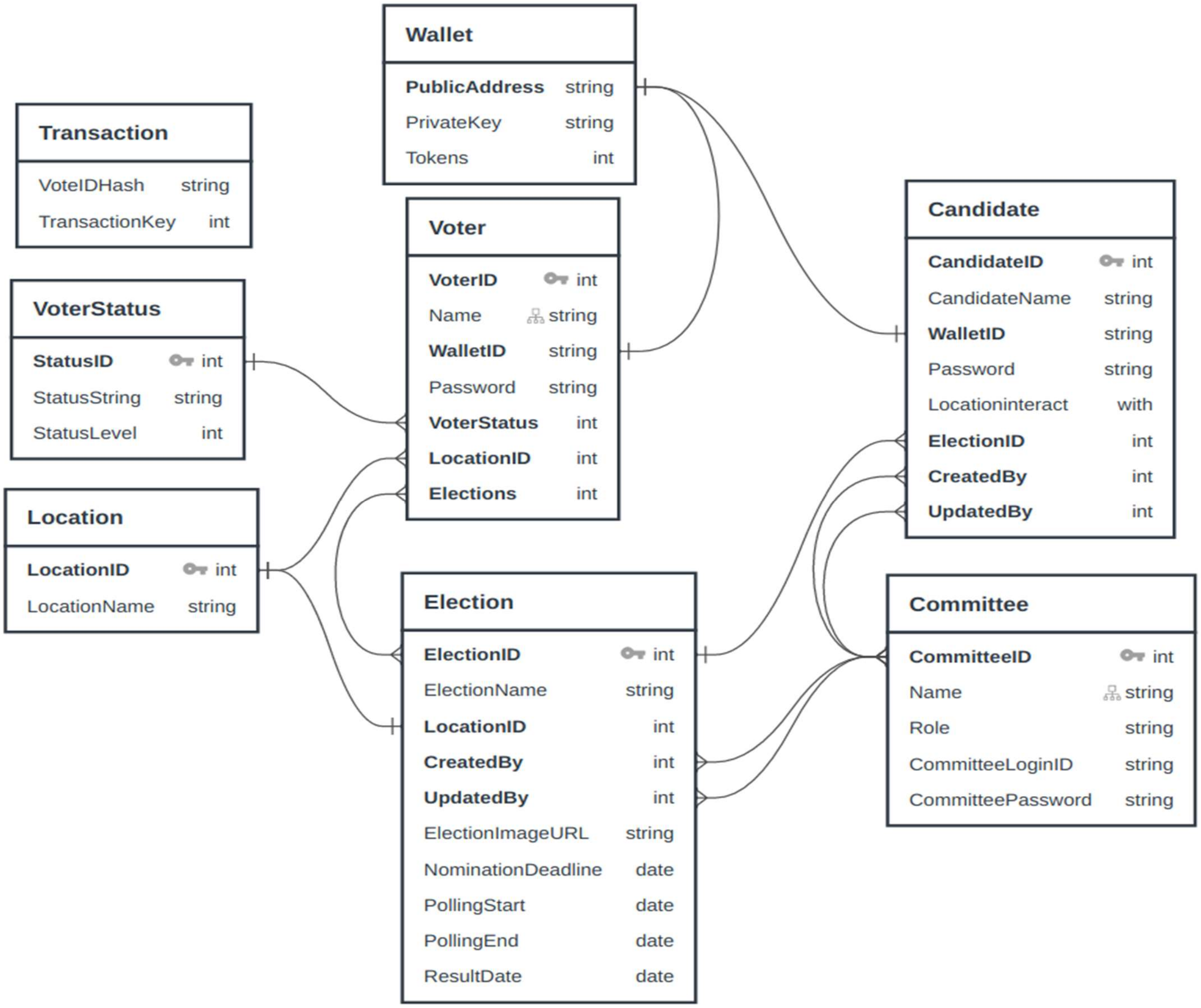

Figure 9. Database Design

The main stakeholders in the system are

1. Voters
2. Candidates
3. Administrators (EC)
4. Moderators (EC)

The location here is the department or grouping of peer nodes. The transaction ballot will be retrieved and stored on to the network

3.6 System Implementation Plan

3.6.1 Modules

The implementation items are divided into modules. These modules contain domain specific tasks and goals, which have to be completed in a specific timeline. There are four proposed modules:

1. Frontend
    a. Voter UI
    b. Election Committee UI
    c. Candidate UI
2. Backend
    a. Election API
    b. Voting API
    c. Voter API
3. Fabric Interface
    a. Network
    b. Certification Authority
    c. Organization (Membership Providers)
    d. Peers
    e. Smart Contracts (Voting)
4. Testing
    a. Deployment on Cloud
    b. Live College Elections

3.6.2 Implementation Phases

The implementation period consists of three phases.

1. Basic Crud Development

    This is the first stage in which the front-end interface will be developed along with the basic CRUD functionalities. These CRUD Functionalities will include the creation of an election process and candidate registration. The expected Timeline would be the first four weeks. The aim is to complete the frontend as well as backend module so as to focus on the interface module.

2. Hyperledger Network Interface Connection

    The interface connection phase will consist of connecting the network functionalities with the application. This is expected to take most of the time as this is a complex process and needs to be handled accurately. The Fabric interface module is to be completed in this phase. Node JS SDK provided by Hyperledger foundation allows us to interact with the network. It consists of middleware for Certification Authorities, Membership Providers, Peer Nodes Interaction, Smart Contract Execution, and much more.

3. <u>Testing and Execution</u>

   This last phase is the actual running phase in which the goal is to complete the application and test it on college level elections. The last Testing module will be completed.

3.6.3 Timeline

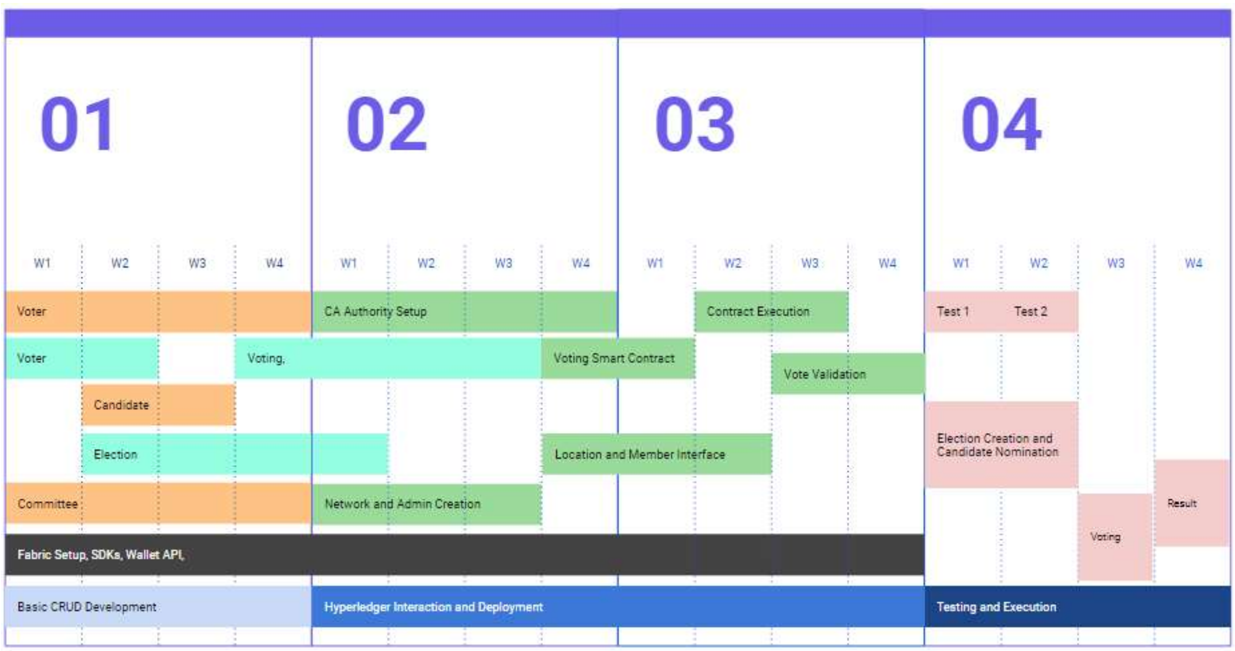

Figure 10. Implementation Timeline

Chapter 4. System Design

4.1 System Architecture

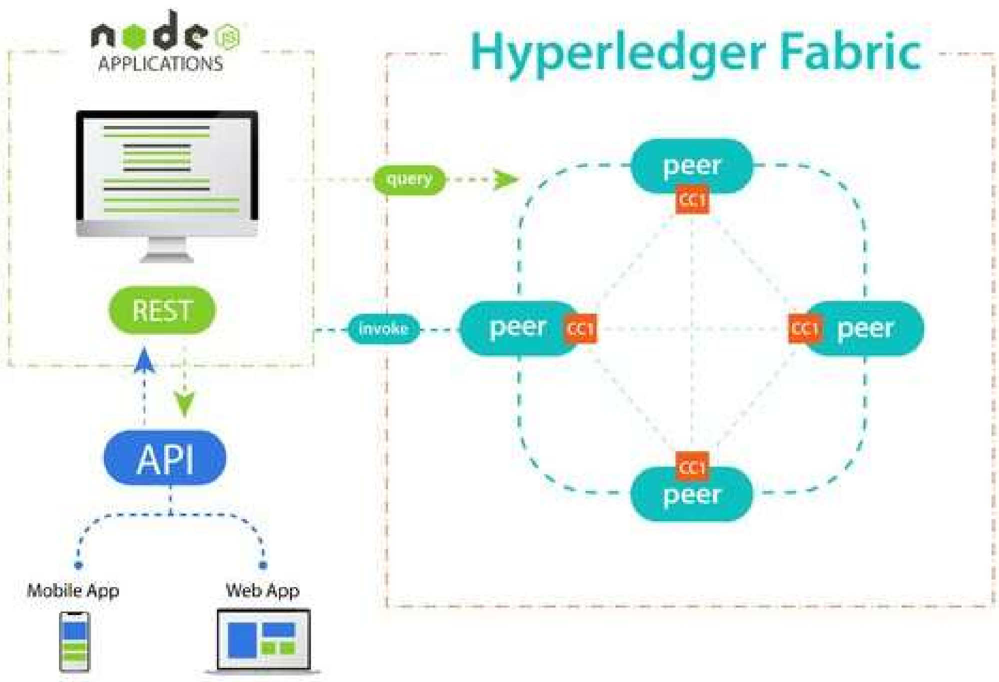

Figure 11. System Overview

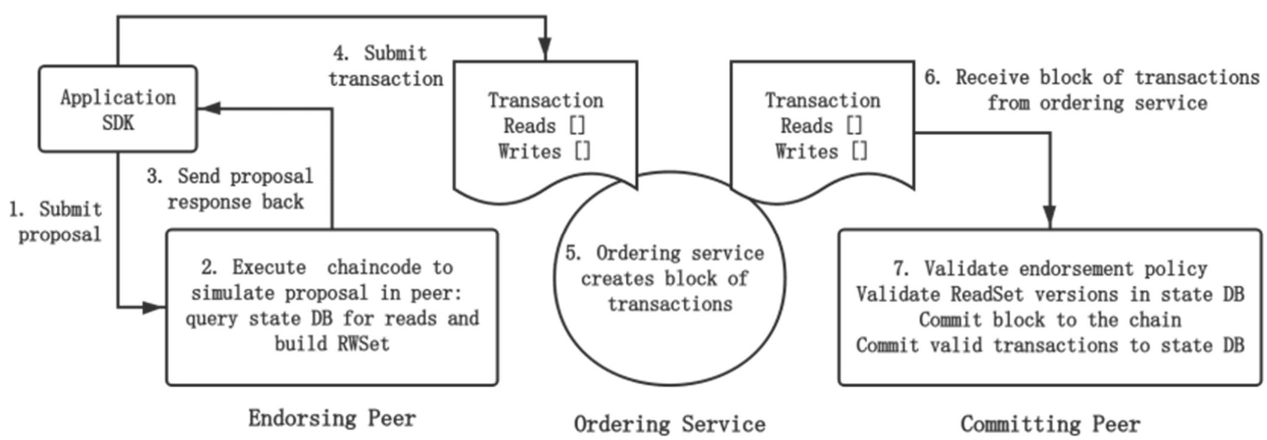

Figure 12. Fabric Architecture

4.2 UML Diagrams

4.2.1 Class Diagram

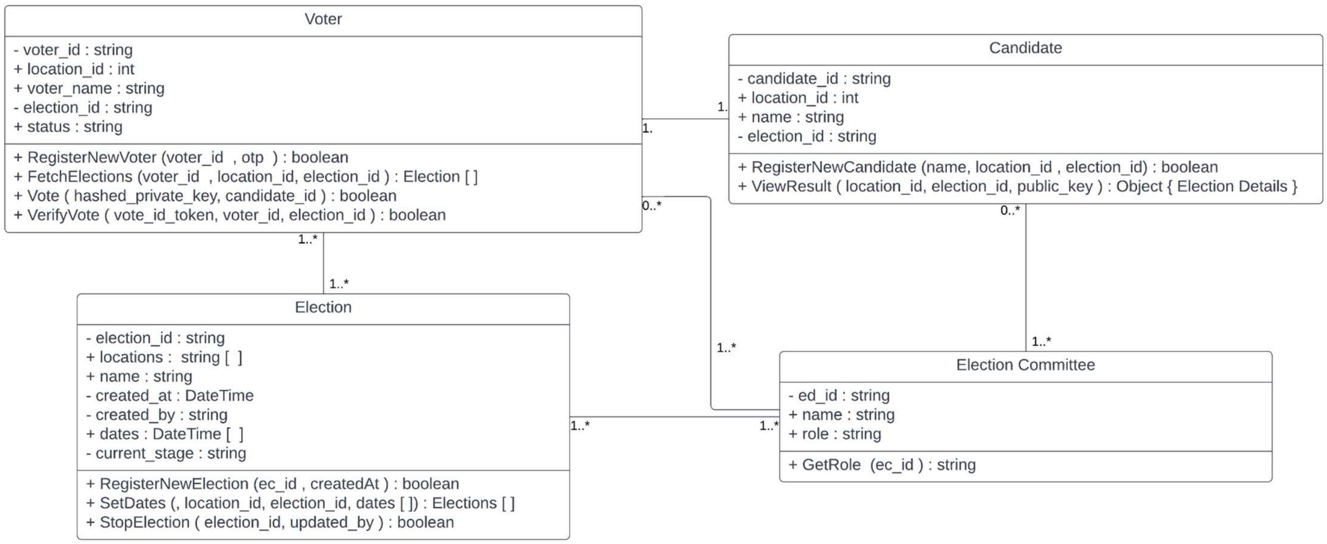

Figure 13. Class Diagram

The main classes are shown in the above diagram. The main stakeholders will be four main classes.

1. Voter
2. Election
3. Candidate
4. Election Committee

These four will be responsible for all of the main implementation. Although the proposed implementation will have a functional programming approach (Express JS), the goal is to transfer the same structure through hard coded models. The middleware will handle most of the validations, such as authentications and wallet retrievals, while these will be responsible for the core ideology of the voting process. The Hyperledger Fabric interface will interact with these models and abstract all the functions while keeping these easy to interact so there can be separation of concern for developers. This will not only facilitate better development experience but also allow a faster development of modules.

4.2.2 Use Case Diagrams

The system has two main use cases.

1. Voter

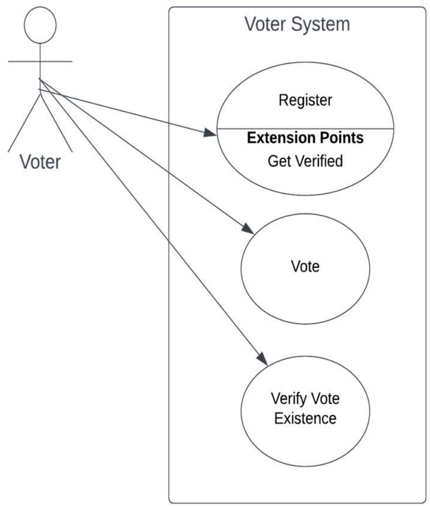

Figure 14. Voter Use Case Diagram

2. Election Committee

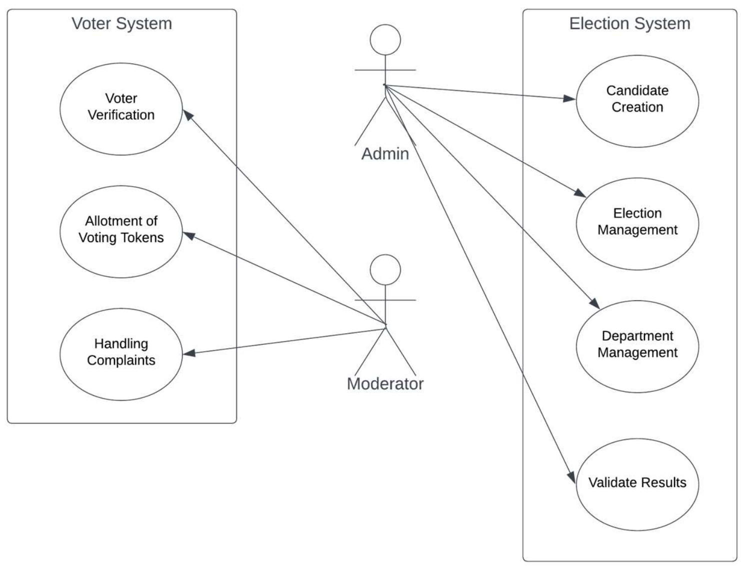

Figure 15. Election Committee (Admins and Moderators) Use Case

4.2.3 Activity Diagrams

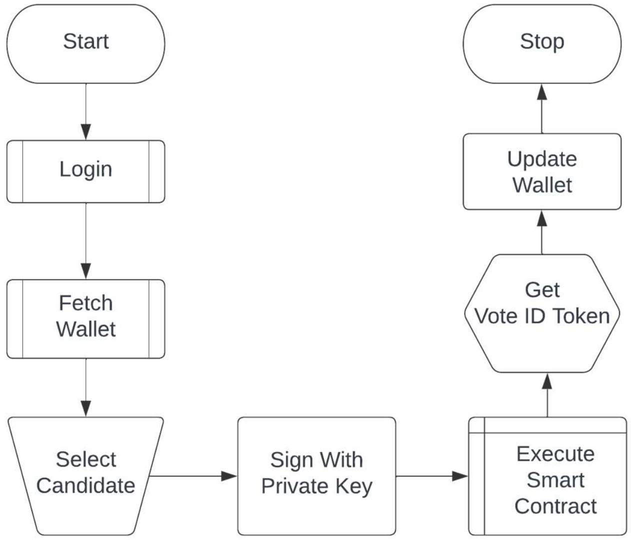

Figure 16. Voting Process

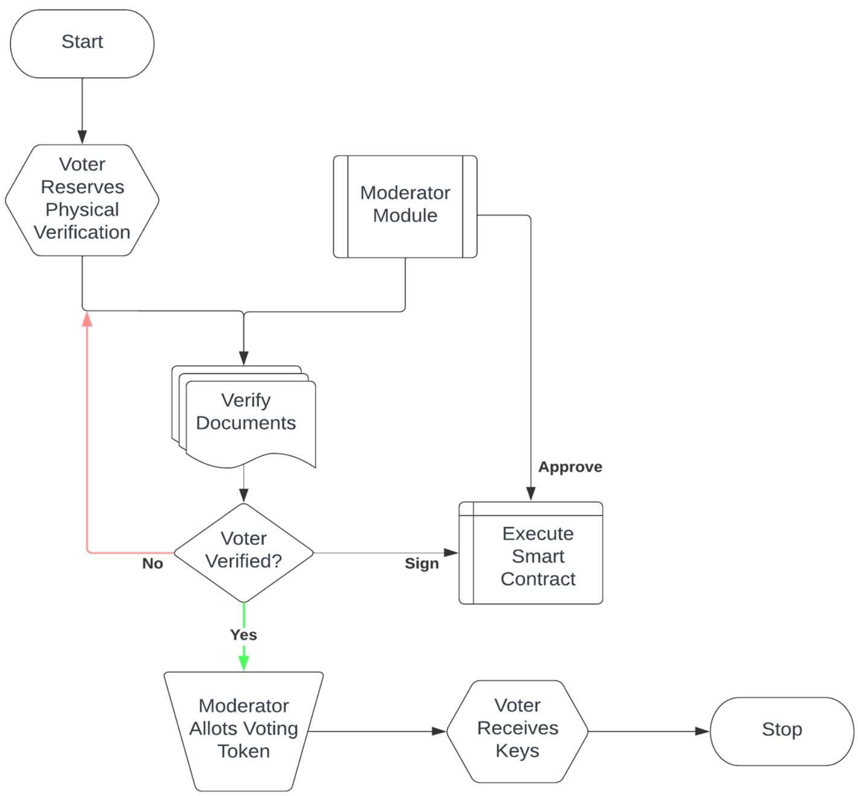

Fig 17. Voter Verification

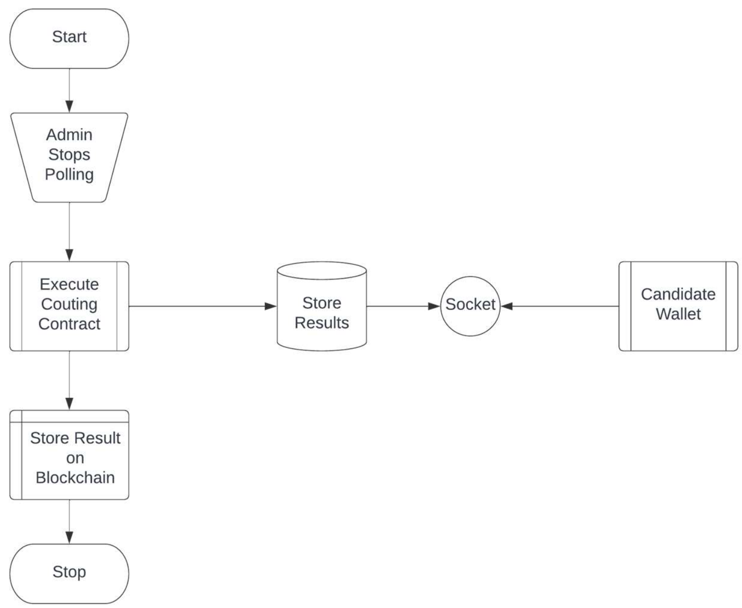

Figure 18. Result Calculation

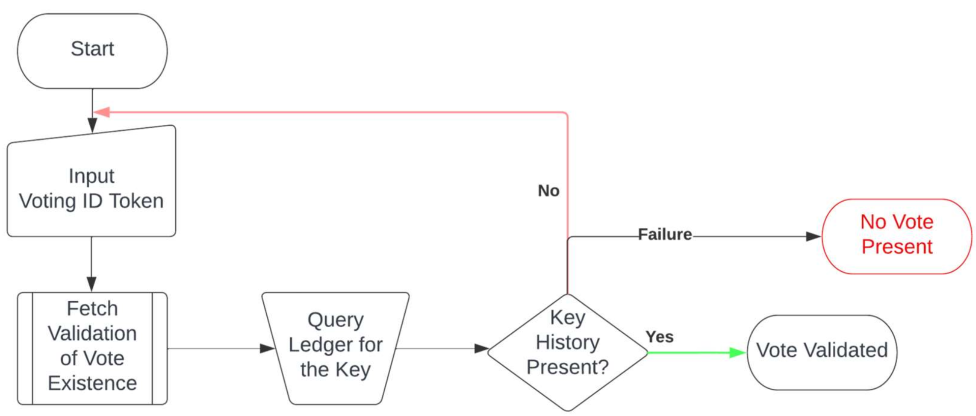

Figure 19. Vote Validation

Chapter 5. Technical Specification

5.1 Hyperledger Fabric

Hyperledger Fabric, an open-source project from the Linux Foundation, is the modular blockchain framework and de facto standard for enterprise blockchain platforms. Intended as a foundation for developing enterprise-grade applications and industry solutions, the open, modular architecture uses plug-and-play components to accommodate a wide range of use cases

Hyperledger Fabric is an open, proven, enterprise-grade, distributed ledger platform. It has advanced privacy controls so only the data you want shared gets shared among the "permissioned" (known) network participants.

Smart contracts document the business processes you want to automate with self-executing terms between the parties written into lines of code. The code and the agreements contained therein exist across the distributed, decentralized blockchain network. Transactions are trackable and irreversible, creating trust between organizations. This enables businesses to make more informed decisions quicker — saving time, reducing costs, and reducing risks.

Hyperledger Fabric is one of the blockchain projects within Hyperledger. Like other blockchain technologies, it has a ledger, uses smart contracts, and is a system by which participants manage their transactions.

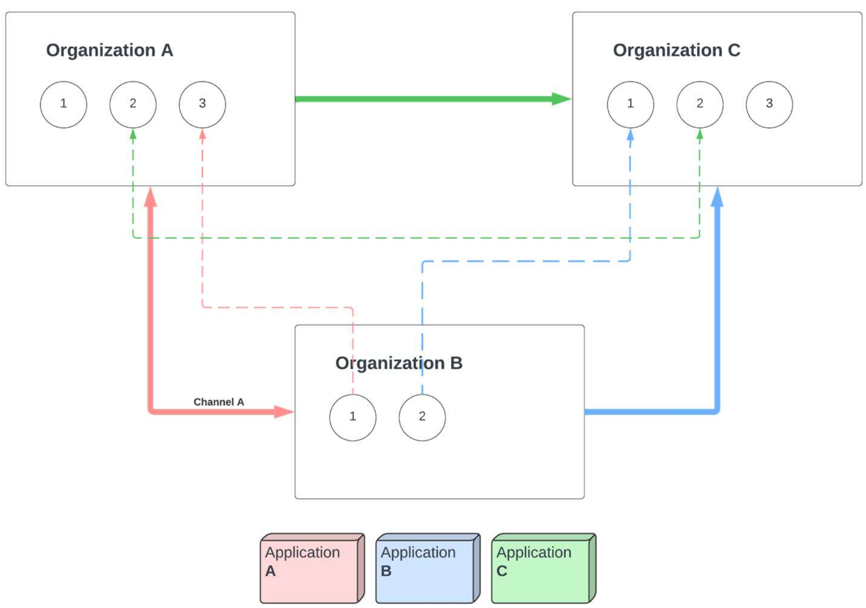

Figure 20. Hyperledger Architecture

Where Hyperledger Fabric breaks from some other blockchain systems is that it is private and permissioned. Rather than an open permissionless system that allows unknown identities to participate in the network (requiring protocols like “proof of work” to validate transactions and secure the network), the members of a Hyperledger Fabric network enroll through a trusted Membership Service Provider (MSP).

Hyperledger Fabric also offers several pluggable options. Ledger data can be stored in multiple formats, consensus mechanisms can be swapped in and out, and different MSPs are supported.

Hyperledger Fabric also offers the ability to create channels, allowing a group of participants to create a separate ledger of transactions. This is an especially important option for networks where some participants might be competitors and not want every transaction they make — a special price they’re offering to some participants and not others, for example — known to every participant. If two participants form a channel, then those participants — and no others — have copies of the ledger for that channel

5.2 Next JS and Node JS

Next.js is an open-source web development framework created by Vercel enabling React-based web applications with server-side rendering and generating static websites. React documentation mentions Next.js among "Recommended Toolchains" advising it to developers as a solution when "Building a server-rendered website with Node.js". Where traditional React apps can only render their content in the client-side browser, Next.js extends this functionality to include applications rendered on the server-side.

The main feature of Next.js is its use of server-side rendering to reduce the burden on web browsers and provide enhanced security. This can be done for any part of the application or the entire project, allowing for content-rich pages to be singled out for server-side rendering. It can also be done only for first-time visitors, to reduce the burden on web browsers that have yet to download any of the site's assets. The "hot reloading" feature detects changes as they are made and re-renders the appropriate pages so the server avoids the need to be restarted. This allows changes made to the application code to be immediately reflected in the web browser, though some browsers will require the page to be refreshed. The software uses page-based routing for developer convenience and includes support for dynamic routing. Other features include hot-module replacement so that modules can be replaced live, automatic code splitting, which only includes code necessary to load the page, and page prefetching to reduce load time.

Next.js also supports Incremental Static Regeneration and static site generation - a compiled version of the website is usually built during build time and saved as a “.next” folder. When a user makes a request, the pre-built versions which are static HTML pages are cached and sent to them. This makes the load time very fast, but it's not suitable for every website because interactive sites that change often and utilize a lot of user input will not be suitable.

5.3 Docker

Docker is an open-source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multi cloud environments.

Developers can create containers without Docker, but the platform makes it easier, simpler, and safer to build, deploy and manage containers. Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API.

5.4 Reference to Technology

1. Hyperledger Fabric - https://www.hyperledger.org/use/fabric
2. Next JS - https://nextjs.org/
3. Node JS - https://nodejs.org/en/
4. Docker - https://www.docker.com/

Chapter 6. Conclusion

Blockchain technology is relatively new to the field and there is a lot exploring remaining to be done to fully grasp its functionalities and capabilities. People associate blockchain with bitcoin and fail to recognize the fact that bitcoin is just an application of the former. Blockchain technology can be used in many diverse applications other than just crypto-currencies and improve their functionality.

The purpose of this project is to build trust between all the parties involved in the voting process. Blockchain based voting will make the process transparent and trustworthy. Large amount of money is spent every election to support the infrastructure, protect its integrity and handle logistics. Whereas the proposed system achieves the same in a cheaper manner.

The proposed system is to implement a method to conduct voting on Hyperledger fabric and take advantage of its characteristics like non-repudiation, immutability and tamper-proof nature. All stakeholders will be able to partake in the process through a simple web app. Privacy of all the parties will be ensured. By providing convenience to the voter, the number of people voting may drastically increase.

There have been previous implementations and the proposed system will overcome their shortcomings by implementing some changes. Scalability and response times will be improved by using a multi-chain approach which will make the blockchain faster.

Chapter 7. References

1. MUHAMMAD SHOAIB FAROOQ, USMAN IFTIKHAR, AND ADEL KHELIFI, "A Framework to Make Voting System Transparent Using Blockchain Technology," date of current version June 10, 2022. DOI: 10.1109/ACCESS.2022.3180168
2. S. S. Hossain, S. A. Arani, M. T. Rahman, T. Bhuiyan, D. Alam, and M. Zaman, "E-voting system using blockchain technology," in Proc. 2nd Int. Conf. Blockchain Technol. Appl., Dec. 2019, pp. 113–117, DOI: 10.1145/3376044.3376062.
3. B. Shahzad and J. Crowcroft, "Trustworthy electronic voting using adjusted blockchain technology," IEEE Access, vol. 7, pp. 24477–24488, 2019, DOI: 10.1109/ACCESS.2019.2895670.
4. F. P. Hjálmarsson, G. K. Hreiòarsson, M. Hamdaqa, and G. Hjálmtýsson, "Blockchain-based E-Voting system," in Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD), Jul. 2018, pp. 983–986.
5. S K Geetha," A Secure Digital E-Voting Using Blockchain Technology", 2021 J. Phys.: Conf. Ser. 1916 012197
6. Mustofa Kamila, Po Abas Sunaryab, Untung Rahardjab, Nuke Puji Lestari Santosob, Muhammad Iqbalb, "Covid-19: Implementation e-voting Blockchain Concept", 2021 Universitas Raharja, Jl. Jenderal Sudirman No.40, RT.002 / RW.006, Cikokol, Kota Tangerang, Indonesia
7. Polyas. Polya's. 2015. Available online: https://www.polyas.co
8. Luxoft. Luxoft. Available online: https://www.luxoft.com/
9. Voatz. Voatz—Voting Redefined. 2020. Available online: https://voatz.com
10. Hyperledger Architecture, Volume 1 - Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus (Journal)
11. Hyperledger Fabric : https://www.hyperledger.org/use/fabric
12. Bitcoin: A Peer-to-Peer Electronic Cash System , Satoshi Nakamoto satoshin@gmx.com , www.bitcoin.org
13. G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung and P. Verissimo, "Efficient Byzantine Fault-Tolerance," in IEEE Transactions on Computers, vol. 62, no. 1, pp. 16-30, Jan. 2013, doi: 10.1109/TC.2011.221.

14. Pawlak, M., Guziur, J., Poniszewska-Marańda, A. (2019). Voting Process with Blockchain Technology: Auditable Blockchain Voting System. In: Xhafa, F., Barolli, L., Greguš, M. (eds) Advances in Intelligent Networking and Collaborative Systems. INCoS 2018. Lecture Notes on Data Engineering and Communications Technologies, vol 23. Springer, Cham. https://doi.org/10.1007/978-3-319-98557-2_21

15. Blockchain : Research and Applications (2022) https://www.sciencedirect.com/journal/blockchain-research-and-applications/vol/3/issue/1

16. V. Lalitha, S. Samundeswari, R. Roobinee and L. S. Swetha, "Decentralized Online Voting System using Blockchain," 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022, pp. 1387-1391, doi: 10.1109/ICAAIC53929.2022.9792791.

17. Hyperlegder Fabric Case Studies : https://www.hyperledger.org/learn/case-studies

18. Working of traditional Voting System (EVM) : https://eci.gov.in/evm/ , https://eci.gov.in/about/about-eci/the-functions-electoral-system-of-india-r2/

19. Cryptocurrency : https://en.wikipedia.org/wiki/Cryptocurrency

Chapter 8. Plagiarism

Plagiarism Checker X Originality Report

Similarity Found: 8%

Date: Saturday, November 12, 2022

Statistics: 1122 words Plagiarized / 6088 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

/ A PROJECT REPORT ON "Enhancing Traditional Voting Systems Using Hyperledger Technology" SUBMITTED BY Om Lachake (BI36) Jitwar Kurhade(BI35) Pranay Malusare(BI42) Manas Tole(BI74)