

# **Blockchain-Based E-Voting System for Elections in Sri Lanka**

**R. A. T. L. Rathnayake**

**2021**



# **Blockchain-Based E-Voting System for Elections in Sri Lanka**

**A dissertation submitted for the Degree of Master of  
Computer Science**

**R. A. T. L. Rathnayake**

**University of Colombo School of Computing**

**2021**






## DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: R. A. T. L. Rathnayake

Registration Number: 2016/MCS/093

Index Number: 16440939

 28-Nov-2021

---

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr. ~~Mr.~~ R.A.T.L Rathnayake under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by,

Supervisor Name: P.V.K.G Gunawardana

 29-November-2021

---

Signature of the Supervisor & Date

I would like to dedicate this thesis to my parents and wife, who constantly supported me to allocate time and keep my focus on completing the research. Without their commitment and inspiration, this research project was not possible.

## **ACKNOWLEDGEMENTS**

First, I like to thank my supervisor Dr. Kasun Gunawardhana, for his continuous supervision, guidance, and encouragement. His commitment towards this research enabled me to continuously challenge and improve myself to complete this research.

I would also like to thank all the staff members of the University of Colombo School of Computing, for providing me the opportunity of this research and the content taught in the study course. The knowledge gained throughout the study course helped me immensely during this research.

I also thank my fellow students, who gave valuable advice whenever I needed it.

Finally, I like to thank my parents and my wife whose constant support kept me motivated and focused.

## ABSTRACT

Electronic voting systems have been an area of interest in a modern democracy for over a decade. With the advancement of technologies to enable the development of e-voting systems, multiple countries have already attempted to use e-voting for public elections with a certain degree of success. E-voting systems are identified as not only an approach to improve time and monetary efficiency, but also to improve voter participation for elections.

With the emergence of blockchain technology, it is identified as an advanced alternative technology for developing e-voting systems because of its highly secure decentralized nature with integrity and auditability by design. The Smart Contracts in blockchain provides an ideal approach to develop the core logic of e-voting systems as it is an integral part of the blockchain system itself. Most of the existing research literature is focused on public blockchains.

Private blockchains, also called permissioned blockchains, provides an alternative blockchain solution that addresses some of the limitations such as the performance, scalability, and legal concerns, identified through the researches on public blockchains. This research aims to design and develop an e-voting system for the context of Sri Lankan elections on a private blockchain to evaluate and understand the applicability of private blockchains for designing e-voting systems. For the scope of the research, the context of the Sri Lankan election is considered and the Hyperledger Fabric is selected as the private blockchain platform.

This research presents a detailed architectural design with a reference implementation on the Hyperledger Fabric private blockchain platform. This proposed solution utilizes the blockchain features to implement e-voting system logic through Smart Contracts and consists of a REST API layer to provide access to consumers. This system is capable of scaling to handle a large number of voters and to satisfy desired properties of e-voting systems. The results obtained through the evaluations confirm the adherence of the system with desired properties of e-voting systems and the capability of handling large voter counts.

This research successfully concludes the applicability of private blockchain to develop e-voting systems for general elections.

# TABLE OF CONTENTS

<b>DECLARATION</b> .....	<b>I</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>III</b>
<b>ABSTRACT</b> .....	<b>IV</b>
<b>TABLE OF CONTENTS</b> .....	<b>V</b>
<b>LIST OF FIGURES</b> .....	<b>VII</b>
<b>LIST OF TABLES</b> .....	<b>VIII</b>
<b>CHAPTER 1 INTRODUCTION</b> .....	<b>1</b>
1.1. MOTIVATION.....	1
1.2. BLOCKCHAIN TECHNOLOGY FOR E-VOTING SYSTEMS .....	3
1.2.1. <i>Elections in Sri Lanka</i> .....	5
1.3. STATEMENT OF THE PROBLEM .....	6
1.4. RESEARCH AIMS AND OBJECTIVES .....	6
1.4.1. <i>Aim</i> .....	6
1.4.2. <i>Objectives</i> .....	7
1.5. SCOPE.....	7
1.6. STRUCTURE OF THE THESIS .....	8
<b>CHAPTER 2 LITERATURE REVIEW</b> .....	<b>10</b>
2.1. ELECTRONIC VOTING FOR PUBLIC ELECTIONS.....	10
2.2. RESEARCHES ON IMPROVING ELECTRONIC VOTING.....	11
2.2.1. <i>Cryptographic Algorithm Based Researches</i> .....	11
2.2.2. <i>Blockchain-Based Researches</i> .....	12
2.3. CONCLUSION.....	17
<b>CHAPTER 3 METHODOLOGY</b> .....	<b>18</b>
3.1. RESEARCH METHODOLOGY .....	18
3.2. SELECTION OF THE RESEARCH PROBLEM .....	18
3.3. ANALYSIS OF THE RESEARCH PROBLEM AND DOMAIN.....	19
3.3.1. <i>Private Blockchain Platform Selection</i> .....	19
3.3.2. <i>Electoral System and Demography of Sri Lanka</i> .....	21
<b>CHAPTER 4 DESIGN AND IMPLEMENTATION</b> .....	<b>23</b>
4.1. USER FLOW OF THE PROPOSED E-VOTING SYSTEM.....	23
4.2. SYSTEM COMPONENTS AND ARCHITECTURE.....	25
4.2.1. <i>Hyperledger Fabric Network Design</i> .....	25
4.2.2. <i>Application API Layer Design</i> .....	26
4.2.3. <i>Deployment Aspects of the E-voting System</i> .....	28
4.2.4. <i>Smart Contract Design</i> .....	28
4.2.5. <i>REST API Design</i> .....	31
4.3. ACHIEVING DESIRED PROPERTIES OF AN E-VOTING SYSTEM .....	38
4.3.1. <i>Anonymity and Privacy</i> .....	38
4.3.2. <i>Coercion Resistance</i> .....	38
4.3.3. <i>Receipt-freeness</i> .....	39
4.3.4. <i>Accessibility</i> .....	39
4.3.5. <i>Accuracy</i> .....	39
4.3.6. <i>Eligibility</i> .....	40
4.3.7. <i>Verifiability</i> .....	40



4.3.8. Robustness of the election .....	40
<b>CHAPTER 5 EVALUATION AND RESULTS .....</b>	<b>41</b>
5.1. EVALUATION CRITERIA .....	41
5.2. EVALUATION METHODOLOGY AND APPROACH .....	42
5.2.1. Data for Evaluation.....	42
5.2.2. Simulation Scenarios.....	43
5.2.3. Prototype Deployment Specifications.....	45
5.3. EVALUATION RESULTS .....	46
5.3.1. Voting Phase Results.....	47
5.3.2. Post Vote Validation Results .....	52
5.3.3. Non-Functional Results.....	55
5.4. LIMITATIONS OF THE EVALUATION .....	61
5.5. SUMMARY .....	61
<b>CHAPTER 6 CONCLUSION AND FUTURE WORK .....</b>	<b>62</b>
6.1. RESEARCH OUTCOME AND CONTRIBUTIONS.....	62
6.2. LIMITATIONS.....	63
6.3. FUTURE WORK.....	63
6.4. CONCLUSION.....	64
<b>APPENDICES .....</b>	<b>I</b>
APPENDIX A: JSON DATA OBJECTS IN SMART CONTRACTS.....	I
APPENDIX B: SECTION FROM SAMPLE TEST DATA.....	V
APPENDIX C: CANDIDATE LEVEL VOTE DATA VS E-VOTING RECORDED DATA FOR SCENARIO THREE .....	VI
APPENDIX D: RESPONSE TIME VARIATION OF ALL REQUESTS FOR SCENARIO ONE .....	XIV
<b>REFERENCES.....</b>	<b>XVI</b>

## LIST OF FIGURES

Figure 1: User Flow of the Proposed E-voting System.....	24
Figure 2: Architecture of the Proposed E-voting System.....	27
Figure 3: Architecture of the Prototype Network Used for Evaluation.....	46
Figure 4: Bar Chart Representing Used Voting Data Against Recorded Valid Votes.....	47
Figure 5: Bar Chart Representing Candidate Level Voting Data Against Recorded Valid Votes for Election ‘Party A’ .....	48
Figure 6: Bar Chart Representing Voting Data Against Recorded Valid Votes .....	49
Figure 7: Bar Chart Representing Candidate Level Voting Data Against Recorded Valid Votes For Election ‘Party A’ .....	49
Figure 8: Bar Chart Representing Voting Data Against Recorded Valid Votes .....	50
Figure 9: Bar Chart Representing Candidate Level Voting Data Against Recorded Valid Votes for Election ‘Party A’ .....	51
Figure 10: CPU Consumption for Scenario One with Three Concurrent Voters.....	57
Figure 11: Network Consumption for Scenario One with Three Concurrent Voters.....	57
Figure 12: CPU Consumption for Scenario One with Six Concurrent Voters.....	57
Figure 13: Network Consumption for Scenario One with Six Concurrent Voters.....	57
Figure 14: CPU Consumption for Scenario Two with Three Concurrent Voters .....	58
Figure 15: Network Consumption for Scenario Two with Three Concurrent Voters .....	58
Figure 16: CPU Consumption for Scenario Three with Three Concurrent Voters .....	59
Figure 17: Network Consumption for Scenario Three with Three Concurrent Voters .....	59

## LIST OF TABLES

Table 1: Hyperledger Fabric Terminology .....	20
Table 2: Data Structures for E-voting in Smart Contracts .....	28
Table 3: Data Elements and Generation Methods for Evaluation .....	42
Table 4: Simulation Scenario One Details .....	44
Table 5: Simulation Scenario Two Details.....	44
Table 6: Simulation Scenario Three Details.....	45
Table 7: Simulation Scenario One Vote Validation Results .....	53
Table 8: Simulation Scenario Three Vote Validation Results.....	53
Table 9: Simulation Scenario Three Vote Validation Results.....	54
Table 10: Simulation Scenario One ‘Validate Voter’ Step Response Times in Milliseconds with Different Concurrent Voters.....	56
Table 11: Simulation Scenario One ‘Vote’ Step Response Times in Milliseconds with Different Concurrent Voters .....	56
Table 12: Simulation Scenario Two Response Times in Milliseconds with Three Concurrent Voters .....	58
Table 13: Simulation Scenario Three Response Times in Milliseconds with Three Concurrent Voters .....	59

# CHAPTER 1

## INTRODUCTION

The purpose of this chapter is to provide an overview of the project and the domain. This chapter elaborates on the motivation for the research along with the domain of elections and e-voting systems, security concerns, followed by a justification on the requirement of using blockchain and *Smart Contracts* for developing an e-voting system for the context of Sri Lankan elections.

In addition, this chapter provides a description of the selection of an applicable blockchain platform with support for *Smart Contracts*, along with the research aims and objective, and the scope of the research project. Finally, the chapter provides an overview of the rest of the chapters contained in this document.

### 1.1. Motivation

In the context of modern democracy, national elections are considered to be an integral component. Elections enable the public to select their preferred representatives to make decisions on behalf of them. Thus, holding fair and peaceful elections on time with maximum security ensures the democracy of a country. Usually, national elections incur a significant cost. When considering a country like Sri Lanka, the cost of an election has a significant impact on the economy. Also, with the traditional paper ballot-based elections, publishing results requires time and manpower. Additionally, there are always concerns raised over the results, due to the lack of visibility.

With computer technologies becoming more advanced over time, there have been proposals to replace the existing paper-based traditional election model with online and computer-based solutions. The main advantages of using e-voting systems are identified as,

- Increased efficiency of the overall election process in terms of time, required manpower,
- Cost-effectiveness,
- Increased availability, accuracy, and reliability (Bokslag and de Vries, 2016), (Stenbro, 2010).

There are several instances where countries including Switzerland (The Swiss authorities online, n.d.) and Estonia (e-Estonia, n.d.) have enabled electronic-based voting. However, security and integrity have always been a concern over the computer-based solutions for

elections (Li et al., 2015). For example, an independent security analysis published on the online voting system of Estonia highlights several critical security vulnerabilities where a “state-level attacker, sophisticated criminal, or dishonest insider” is able to manipulate the election results (Springall et al., 2014).

In an e-voting system, security risks can be categorized into three high-level categories (Okediran O. O et al., 2011), (Franke and Darmstadt, n.d.).

- Risks at the client-end,
- Risks in the communication link between the client and server and
- Risks at the service/storage layer.

Out of these three categories, the most significant security threat in terms of the impact is the risks at the e-voting service layer (Okediran O. O et al., 2011), (Franke and Darmstadt, n.d.). If a security breach happens at the e-voting service/storage layer, then the attackers may be able to alter the complete election results by either manipulating the calculations or tampering with the voting storage. In comparison, a security breach at the client layer or in the communication layer between a client and the server will mostly impact only those specific scenarios within a limited scope (Franke and Darmstadt, n.d.). In addition to these primary security threats, there can be other security threats such as social engineering, which are focused on the weaknesses of human users (Okediran O. O et al., 2011).

For an e-voting system to be accepted, there are several key properties that the system should satisfy (Novotný, 2009), (Yao and Houston, 2002). These properties include,

- Anonymity and Privacy: a ballot cannot be linked to a voter and sensitive voting information should not be accessible to unwanted parties.
- Coercion resistance: a coercer cannot extract information on what way a voter has voted.
- Receipt-freeness: a voter cannot prove his/her selection by creating a receipt.
- Accessibility: eligible voters should be able to access the system and vote conveniently.
- Robustness of the election: the system should be robust against active/passive attacks and faults.
- Accuracy: the system should guarantee that all ballots are accurately counted.
- Eligibility: voters must be registered according to a predefined criterion and voting is allowed only for such registered users.
- Verifiability: voters should be able to verify that his/her vote was accounted for in the final result.

For any e-voting system to be successful and accepted, the above-mentioned properties must be satisfied. Thus, the solution proposed during this research focuses on satisfying these properties.

## **1.2. Blockchain Technology for E-Voting Systems**

The adaptation of blockchain technology, which was the underlying technology of the success story of Bitcoin, offers an alternative technological stack for developing computer-based e-voting solutions. The blockchain provides a highly secure decentralized, distributed system with integrity by design. The core components of a blockchain include distributed ledger, transactions, and consensus mechanisms (Puthal et al., 2018). The distributed ledger is the data storage component of the blockchain system where the blocks are stored. The ledger has a unique significant property, of the stored blocks being immutable. The transactions are the operations that take place in the blockchain system and a set of transactions are collected together to form a block which is then stored in the ledger. The consensus mechanism is used to ensure new blocks are added to the blockchain only upon the agreement of all (or majority of) the nodes in the network. The consensus algorithm acts as the fault tolerance mechanism which enables the blockchain to function correctly even with failed or malicious nodes. With these components, blockchain technology is capable of achieving multiple desirable properties including decentralized and peer-to-peer communication, distributed ledger, enhanced security, and immutability of records.

With the emergence of Blockchain and related technologies, Smart Contracts which is a term coined by Nick Szabo in 1994 (“The Idea of Smart Contracts | Satoshi Nakamoto Institute,” n.d.), has gained significant attention in the recent past. By definition, Smart Contracts refer to computer programs that can self-execute upon agreements between involved parties according to agreed terms and conditions. Smart Contracts in blockchain systems are capable of exchanging currencies and assets, and maintain/access blockchain records once added to the blockchain system. The Smart Contract is a computer program written in a programming language and usually consists of terms, conditions, and various logics which can be executed automatically. These Smart Contracts can provide the means of implementing the core functionalities of the various application domains securely, within the boundaries of the blockchain ecosystems.

With such preferable properties and capabilities, blockchain technology has been adapted to various problem domains through researches. Some of the most popular application domains of blockchain included cryptocurrency, asset management, finance, healthcare, insurance, and

copyright management (Chen et al., 2018).

Similarly, for the domain of e-voting, blockchain technology can be considered as a solution to provide a more secure approach when compared with the traditional database-oriented approaches (Taş and Tanrıöver, 2020). As discussed above, with the functionality of blockchains, a block entry can be added to the blockchain only after getting the consent of other nodes. Once a block is written to the blockchain, those blocks are immutable meaning it is impossible to alter written blocks. Also, when required depending on permissions, users can audit the blockchain to determine the progress (Zheng et al., 2017). These inherent properties of blockchain technology address several limitations and vulnerabilities at the service layer of the existing e-voting system and thus make it an ideal candidate technology for implementing an election voting system.

Blockchain systems are classified into three categories as public blockchains, private blockchains, and consortium blockchains (Puthal et al., 2018).

The public blockchains provide an open platform for anyone to join the network and perform transactions. Thus, public blockchains are also referred to as ‘permissionless’ blockchains. Since its open to the public, anyone can see the transactions and entries in the ledger. Due to the highly distributed nature, commonly used consensus algorithms are proof-of-work and proof-of-stake.

Private blockchains are permissioned blockchains where the blockchain is owned and controlled by a single organization and used by a set of identified and authorized users. When compared to public blockchains, private blockchains are less decentralized but have advantages of higher performance, higher scalability, and less resource consumption.

The consortium blockchains are also permissioned blockchains but are owned and controlled by multiple organizations.

Several researches were conducted using blockchain to solve the security concerns of e-voting systems. These researches and some of their drawbacks will be discussed in detail in the ‘Chapter 2: Literature Review’ chapter. Based on the literature review, a research gap is identified on the lack of researches on e-voting systems using private blockchains and evaluation of consensus algorithm applicability. Therefore, during this research, we evaluate the usage of private blockchain to implement the proposed e-voting system. Thus, selecting the appropriate blockchain platform is a critical decision for the research. The main criteria for the selection of a blockchain platform include the ability to support private blockchains, support for smart contracts, support for multiple consensus algorithms, stability, and community

support. For the selection, the following popular private permissioned blockchain platforms were considered.

1. Ethereum: Even though Ethereum is more popular as a public blockchain, still since it is open-source, it is possible to set up Ethereum as a private blockchain as well. However, compared to specifically designed private blockchains, Ethereum offers fewer functionalities for permission handling. Also, only “Proof of Work” and “Proof of Authority” consensus algorithms are supported.
2. Quorum: This is a fork of Ethereum, and provides more features for setting up a private blockchain. It also supports multiple consensus algorithms including RAFT and IBFT.
3. Hyperledger Fabric: This is one of the projects under the Hyperledger umbrella which is supported by the Linux foundation. Hyperledger Fabric is one of the most popular private blockchain platforms and offers functionalities to configure permissions on the blockchain with multiple organizations and advanced authorization features. Also, it has numerous performance enhancements to overcome the issues faced in other blockchain platforms. Hyperledger Fabric supports a pluggable consensus algorithm architecture with PBFT, RAFT, and Kafka consensus algorithms.
4. MultiChain: This is an open-source blockchain and is a fork of the BitCoin. MultiChain supports the features including permission setups and privacy setups.

Out of these considered blockchain platforms, the Hyperledger Fabric platform is selected as the blockchain platform for this research. The main reasons for selecting this platform are the rich set of features it provides for Smart Contract developments and user permission control, along with the multiple consensus algorithm support with the pluggable architecture, and the stability and establishment of the Hyperledger projects, and the support of the Linux Foundation and the community. In addition, as per the evaluation that has been performed by Julien Polge et al. (Julien Polge et al., 2020), the Hyperledger Fabric platform has outperformed other considered platforms.

### **1.2.1. Elections in Sri Lanka**

Sri Lanka has been using the traditional paper-based election system. As highlighted previously, for each election, the government has to bear a significant cost. The cost of the presidential elections in 2019 was about 7.5 billion rupees (Centre for Monitoring Election Violence, 2019) and the 2020 general election was estimated at around 9-10 billion rupees.



Usually, with manual ballot counting, it takes around 24 hours to publish the official results. This entire process can be improved in a cost-efficient and transparent manner if a blockchain-based electronic voting system can be introduced. The proposals presented during this research can act as the basis and guide for designing the service layer of a cost-effective and secure electronic voting system for Sri Lankan elections.

Thus, in this research we investigate the applicability of the private blockchain as the service/storage layer for processing and storing of votes in an e-voting system in the context of Sri Lankan elections, with the expectation of performing public elections in Sri Lanka through a secure computer-based solution, thus reducing the repeatedly incurred costs and improving the overall quality of elections. For developing this solution, the 'Hyperledger Fabric' blockchain platform and Smart Contracts will be used as the core technologies.

### **1.3. Statement of the problem**

Electronic voting has been considered as a solution to reduce recurrent costs and engage more voters. However, there have been several concerns over the security, privacy, auditability, and transparency of e-voting systems. The blockchain technology provides an advanced alternative for designing electronic voting systems with higher security, availability, and auditability. The usage of public blockchains for developing e-voting systems introduces limitations on the performance and the scalability of the system due to the size and participants of the blockchain. The alternative approach is to use private blockchains and there is a lack of researches on the usage of private blockchains for designing e-voting solutions at present.

To fully evaluate the applicability of private blockchains for e-voting systems, it is required to design and develop an e-voting system on private blockchain satisfying all desirable properties of an e-voting system and evaluated the outcome. Such a system will enable us to overcome the limitations inherited on public blockchains as well as the concerns over security, privacy, auditability, and transparency of traditional systems.

### **1.4. Research Aims and Objectives**

#### **1.4.1. Aim**

This research aims to develop an electronic voting system for the context of Sri Lankan elections on a private blockchain and to evaluate and understand the applicability of private blockchains for designing e-voting systems.

## 1.4.2. Objectives

In order to achieve the above-mentioned aim, the objectives of this research are,

- Define the user flow of the proposed e-voting system,
- Design the architecture of the e-voting system for the demography of Sri Lanka to ensure availability, robustness, and high performance,
- Define and develop Smart Contracts to enable required properties of an e-voting system which are anonymity, privacy, coercion resistance, receipt freeness, accessibility, accuracy, eligibility, and verifiability,
- Define a process to enable auditing of the election,
- Evaluate the performance of the e-voting system with reference to the selected consensus algorithm.

## 1.5. Scope

An electronic voting system can be segregated into three components,

- The service layer, which is responsible for the data processing and storage,
- The client layer, which is the interface with the voter and
- The communication link, connecting the client and the service layers.

Within the scope of the research, only the service layer of an e-voting system will be considered and this layer will be designed using blockchain technology for data processing and storage. The security risks at the consumer end and the communication link layer will not be considered for the scope. Thus, the solution proposed by this research will not cover the end-to-end election process.

The election process of a country is a complicated and substantial process involving numerous departments and organizations. For the scope of this research, only the stages of vote casting and vote tallying processes in an election will be considered. Election management processes such as registration of voters and candidates, will not be considered for the scope of this research.

Also, the scope of the proposed e-voting system solution will be focused on the context of the Sri Lankan elections. Since the procedures and legislation are different across different countries, using the proposed system directly in a different country will be limited. Also, the architecture of the proposed solution will be based on the demography and electoral structure

of Sri Lanka.

Within the scope of this research, the Hyperledger Fabric blockchain platform will be used. Thus, the proposing solution will have dependencies on the design and features of the Hyperledger Fabric blockchain platform. Generalizing the proposed solution for other platforms is not considered for the scope of this research.

The Hyperledger Fabric supports a pluggable architecture for consensus algorithms. For the scope of this research, the RAFT consensus algorithm (Ongaro and Ousterhout, n.d.) will be used. The performance of the proposed solution will be evaluated against the RAFT algorithm. This research will not focus on any lottery-based or ‘proof of work’ related consensus algorithms.

## **1.6. Structure of the Thesis**

This section provides an overview of the research paper and the rest of the chapters.

### Chapter 2 – Literature Review

The Literature review chapter will provide a detailed review of the existing solutions on existing e-voting solutions focusing on the researches related to blockchain-based e-voting solutions. This section will further describe the strengths and weaknesses of various approaches, common problems faced by all types of these approaches. Finally, the existing research gaps will be discussed.

### Chapter 3- Methodology

This chapter will provide details on the research methodology and the design of the solution.

### Chapter 4- Design and Implementation

This chapter will describe the architectural design of the proposed system and justifications on the design decisions, along with how certain properties are achieved with this proposed system design.

### Chapter 5- Evaluation and Results

The evaluation chapter will provide details on the evaluation. This chapter will further describe the approaches taken for evaluation purposes and the outcomes from these evaluations.

### Chapter 6 – Conclusion and Future Work

The Conclusion chapter will provide an analysis of how the objectives were successfully

achieved, the problems encountered, and the limitations of the proposed solution. Finally, future enhancements and concluding remarks will be described.

## **CHAPTER 2**

### **LITERATURE REVIEW**

The purpose of this chapter is to provide an in-depth analysis of the existing literature and solutions on the topic of electronic voting systems and the usage of blockchain technology for designing electronic voting systems for elections.

In this chapter first we discuss and review some of the existing electronic voting systems used for public elections to highlight the importance of such electronic voting systems and to review the drawbacks of these existing systems. Then the researches performed to improve the security of such electronic voting systems is evaluated by classifying researches under two categories as non-blockchain and blockchain-based researches. Since this research is focused on the usage of blockchain technology, researches under the blockchain-based category are discussed further by categorizing those under two sections as researches based on public blockchains and researches based on private blockchains.

#### **2.1. Electronic Voting for Public Elections**

With the advancement of computer technologies, there has been a significant interest in the usage of electronic-based systems for holding public elections. At the time of this research, over ten countries have conducted public elections with the usage of either full or partial support for electronic voting including the United States, United Kingdom, Brazil, Estonia, Norway, India, and Switzerland.

Among these countries, Estonia has held multiple national elections using e-voting from the year 2005 onwards with 43.75% of electronic voting recorded for the parliamentary election held in 2019 (e-Estonia, n.d.). However, in the evaluation performed by Springall et al., a number of security weaknesses of the system have been identified along with the critical security gaps of the operational process (Springall et al., 2014). This evaluation highlights several critical security vulnerabilities where a “state-level attacker, sophisticated criminal, or dishonest insider” is able to manipulate the election results, emphasizing the importance of the security and ability of auditing for e-voting systems.

Switzerland can be identified as another country actively focusing on the usage of electronic voting from 2009 (The Swiss authorities online, n.d.). The e-voting system used by Switzerland has not caused significant concerns over security aspects, however, in an evaluation performed

by ‘Franke and Darmstadt’, concerns over the trust of the system has been raised due to the lack of visibility and auditing has been highlighted (Franke and Darmstadt, n.d.)

There have been several scenarios where the adaptation of e-voting has failed and the initiatives of using e-voting systems have been discontinued in several countries. For example, Finland has temporarily stopped progressing with e-voting systems after an internal evaluation recommended against the usage of e-voting for general elections (“Working group,” n.d.). The e-voting system trialed in Germany was criticized for lack of transparency and later was determined as unconstitutional by the Constitutional Court of Germany in 2009 (Weiler, 2016). Similarly, usages of e-voting projects in Norway, France, Ireland, Kazakhstan have been discontinued mainly due to security concerns and the lack of auditing and visibility.

Thus, taking above mentioned use cases into consideration, it can be concluded that even though there is a great interest shown by many countries in adapting e-voting for general elections, there have been many challenges in the aspects of security, auditability, and visibility. These challenges have become a bottleneck for the evolution of e-voting systems. There have been many researches performed to overcome these challenges in recent years and the rest of this chapter focuses on the strengths and drawbacks of these researches and to identify research gaps.

## **2.2. Researches on Improving Electronic Voting**

During this section, the most significant researches on improving e-voting systems are discussed with strengths and limitations on such researches. When considering the literature regarding e-voting system improvements, those can be classified under two main categories as ‘Cryptographic algorithm-based approaches’ and ‘Blockchain-based approaches’ for the scope of this research.

### **2.2.1. Cryptographic Algorithm Based Researches**

There have been many researches on improving the security aspects of e-voting systems using cryptographic approaches. Some of the significant researches on this category include the proposal by Weber, 2008 on creating a coercion-resistant cryptographic voting protocol using Zero-knowledge proof and Threshold cryptosystems as the basis (Weber, 2008). In another research, Ibrahim et al., 2003, proposes an e-voting system using the blind signature technique (Ibrahim et al., 2003). The research by Hao et al., 2010, proposes an automated vote tallying mechanism using the two-round anonymous veto protocol (Hao et al., 2010).

Since the focus of this research is on blockchain-based researches for the service layer of an e-voting system, each of these research approaches is not discussed in detail. In addition, there are a large number of researches performed on cryptographic algorithms for e-voting systems, which are not explored during this research.

Even though the cryptographic algorithm-based approaches significantly improve the security aspects during voting especially at the communication layer and the client layer, still the e-voting system depends on trusted centralized parties at the storage layer, which still can be compromised. Additionally, most of such researches focus mainly on security aspects, however, as highlighted in the previous section lack of visibility and lack of auditability are also critical aspects that need to be addressed if e-voting systems are to be accepted.

## **2.2.2. Blockchain-Based Researches**

With the evolution of cryptocurrencies, blockchain technology has been proven and established as a technology to provide highly secure, distributed architectural solutions for storing critical information, while providing visibility on transactions. Thus, there have been multiple researches on using the blockchain technology to develop highly secure e-voting systems while preserving the transparency of transactions and enabling auditing of transactions. The available literature can be grouped under five categories for the scope of this study. These categories are,

- Researches on general blockchain-based e-voting solutions,
- Researches focused on public blockchain-based e-voting solutions,
- Researches focused on private blockchain-based e-voting solutions,
- Researches on consensus algorithms for blockchain-based e-voting solutions,
- Researches focusing on electoral demography for blockchain-based e-voting solutions.

### **2.2.2.1. Researches on General Blockchain-Based E-voting Solution**

There have been many researches published on evaluating the applicability of blockchain technology and Smart Contracts as the basis of developing improved e-voting solutions. The researches discussed under this section are focused on establishing the importance of blockchain as a secure, highly available, and cost-effective solution for e-voting systems that enable overcoming limitations of existing systems.

Ben Ayed, proposed one such design for a blockchain-based e-voting system where the voter registry is stored in a relational database and the voting is performed through a blockchain system (Ben Ayed, 2017). Through this author claims that the proposed system enables public verifiability and high availability with blockchain. However, due to the dependency of a centralized database, this proposed system still inherits limitations of traditional e-voting systems and does not fully utilize the features of blockchains. Srivastava et al. propose a solution with customized blockchain to achieve security, reliability with high throughput (Srivastava et al., 2018). This solution depends on custom block sizes, special protocols thus adapting to existing blockchain platforms is challenging. Hjálmarsson and Hreiðarsson also proposed a similar customized blockchain to achieve higher security and throughput (Hjálmarsson and Hreiðarsson, n.d.).

Kshetri and Voas, describe the advantages and opportunities of using blockchain for e-voting and concludes the limitations outweigh the advantages (Kshetri and Voas, 2018). In addition, publications by Krimmer, Adiputra, et al., highlight the importance and advantages of using blockchain for e-voting systems in terms of privacy, transparency, and availability (Adiputra et al., 2018) (Krimmer, n.d.).

These researches emphasize the importance and advantages of using blockchain technology for e-voting systems, even though there are still areas to be improved.

#### **2.2.2.2. Researches Focused on Public Blockchain-Based E-Voting Solution**

A public blockchain is a permissionless, highly decentralized blockchain where any user can read and write to the blockchain. Two of the most popular public blockchains are Bitcoin and Ethereum at the time of this research. There have been numerous researches performed on the usage of public blockchains for developing e-voting systems. This section study and review the most significant researches performed on public blockchains for e-voting systems.

The proposal by Zhao and Chan, on using BitCoin for a simple and limited ‘Yes/No’ type of e-voting system can be considered as one of the earliest adaptations of public blockchain for e-voting (Zhao and Chan, 2016).

When considering the literature on blockchain-based e-voting systems, Ethereum has been the most preferred blockchain platform (Taş and Tanrıöver, 2020).

Yavuz et al. proposed an adaptation of Ethereum blockchain and Smart Contracts for a basic e-voting system utilizing the integrity, verifiability, transparency, and distributed nature of the



blockchains (Yavuz et al., 2018). This research focuses on providing only a high-level adaptation of Ethereum. In another research, Khoury et al. proposed the usage of the Ethereum blockchain to achieve auditability and transparency with the additional functionality of mobile number based voting validations (Khoury et al., 2018). Hardwick et al. proposed a system with auditability and the additional feature of updating casted votes on Ethereum (Hardwick et al., 2018). Also, this research provides statistics on the cost and performance of the proposed system which can be used as benchmark values.

Similarly, there have been more researches performed on the Ethereum public blockchain with minor differences and the common goal of these researches is to adapt the transparency, distributed nature, auditability, and integrity of blockchain systems to design e-voting systems (Bartolucci et al., 2018; Hjálmarsson and Hreiðarsson, n.d.).

Even though the researches under this taxonomy addresses most of the critical functionalities of e-voting systems such as high availability, robustness, voter privacy, verifiability, resisting coercion when considering a combination of proposals, there are several limitations imposed by the usage of public blockchains. Since the public blockchains are accessible by anyone, there is an associated risk of exposing unnecessary data as well as legal restrictions as this data can reside in any place in the world without any restrictions. Also due to the size of the public blockchains (Bitcoin and Ethereum main blockchains), there is a concern over the scalability and the performance of the systems (Khan et al., 2020; Taş and Tanrıöver, 2020).

#### **2.2.2.3. Researches Focused on Private Blockchain-based E-voting Solution**

A private blockchain is a permissioned blockchain where only a selected and verified parties are allowed to access and are usually owned by single or multiple organizations. Due to permissioned nature and restricted ownership, private blockchains address the trust and privacy concerns, performance, and scalability concerns of using a public blockchain for e-voting systems. However, this also restricts the transparency and special design considerations needed to achieve transparency. Some of the most popular private blockchains include Hyperledger Fabric and Quorum.

There have been a limited number of researches on the usage of private blockchains for e-voting systems when compared to public blockchains. One such research is performed by authors Zhang et al., with their proposal for a protocol with end-to-end privacy with a reference implementation in the Hyperledger Fabric blockchain (Zhang et al., 2018). The proposed solution is driven completely on blockchain and is capable of preserving privacy and the ability

to detect and correct dishonest usages. One major limitation of this proposed system is that the voter eligibility is determined by the voter being a peer of the network. This introduces a scalability limitation and is impractical to be used with a considerable number of voters. Additionally, the proposed protocol does not provide verifiability, robustness, and anonymity.

In the research performed by Kirillov et al., the authors proposed a Hyperledger Fabric based design that enables a combination of e-voting as well as paper-based voting (Kirillov et al., 2019). Similar to the above-discussed approach, this design also relies on the x509 certificates to determine the eligibility and thus has the limitation of scaling to a large number of users. This design uses blind signature at registration as a security measure, however, it has the limitation where some users might get restricted from using both e-voting and traditional approaches. The proposed solution is capable of satisfying privacy, anonymity, verifiability. However, it has concerns over accessibility, eligibility, and accuracy of results.

Another solution was proposed by authors Mukherjee et al., in their research on designing a Hyperledger Fabric based e-voting FaaS (Framework as a Service) system (Mukherjee et al., 2020). In this paper, the authors propose a layered architecture with four layers as the data layer, blockchain layer, microservices layer, and API layer. This research discusses only the high-level architecture of the system and specific security measures are not discussed. Also, electoral operations such as voter authentication, vote validation, vote tallying, vote auditing are done at the micro-services layer instead of at Smart Contracts, which means the blockchain capabilities are not optimally utilized. The proposed architecture is capable of providing a reusable and scalable solution with a certain level of security. However, since the solution is described at a high level, it is difficult to determine the attributes such as privacy, anonymity, verifiability, and accuracy of results.

As per the evaluated existing literature, the number of researches performed on using private blockchains for e-voting systems is limited. And as discussed above, there are limitations in the existing proposals which need to be addressed and solved. In the research review by Taş and Tanrıöver, authors have identified the importance of the usage of private blockchains for e-voting systems while ensuring transparency as a research gap to be addressed (Taş and Tanrıöver, 2020).

#### **2.2.2.4. Researches on Consensus Algorithms for Blockchain-based E-voting**

The consensus algorithm of a blockchain determines the fault tolerance mechanism in which multiple nodes in the blockchain agree on data blocks. The most commonly used consensus algorithms in blockchain frameworks include Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA). When an e-voting system is designed to use a public blockchain, then the e-voting system must adhere to the consensus algorithm used in the particular public blockchain. However, for e-voting systems based on private blockchains, researchers have the opportunity of selecting a suitable consensus algorithm supported by the blockchain framework.

When evaluating most of the literature on blockchain-based e-voting systems, there is little information available on the consensus algorithms used. In the research by Luo et al., usage of DPoS consensus algorithms is proposed to improve the security and efficiency of e-voting systems (Luo et al., 2018). Li et al. proposed a consensus algorithm named Proof of Vote (PoV) which is targeted to improve the efficiency in using blockchains for e-voting systems (Li et al., 2017).

In the research review by Taş and Tanrıöver, the authors highlight the lack of information on the consensus algorithms used for blockchain-based e-voting solutions and emphasize the importance of publishing the consensus information and stats with research publications.

#### **2.2.2.5. Focusing on Electoral Demography for Blockchain-based E-voting**

When developing a blockchain-based e-voting system for national elections, the solution will have a dependency on the demography and the electoral and legal systems of the country. Thus, there are researches performed focusing on elections of particular countries.

Bulut et al. proposed a blockchain-based e-voting system with a layered architecture, which is optimized for Turkey considering aspects of the electoral system, demography, internet availability, etc. (Bulut et al., 2019). Daramola and Thebus proposed a blockchain-based e-voting focusing on the context of South Africa (Daramola and Thebus, 2020).

The existence of such researches emphasizes the importance of adapting the blockchain solution considering the demography, legislation, and other factors when designing an e-voting system for a country.

When considering the context of Sri Lanka and blockchain-based e-voting, Sirimanna, and

Jinasena, have proposed an e-voting system considering the scope ‘Grama Sewa officers’ (Sirimanna and Jinasena, 2019). This system proposes a high-level design of the system, without defining the implementation of the system.

### **2.3. Conclusion**

This chapter discussed and evaluated the existing literature on e-voting systems, focusing mainly on blockchain-based e-voting systems.

Based on this literature review, it can be identified that the usage of private blockchain-based solutions provides advantages on performance and scalability when compared with public blockchain-based e-voting systems. It is also identified that there is only a limited number of researches performed on the usage of private blockchains for e-voting systems. Also, the available proposals on private blockchains still have certain limitations, especially on scalability aspects, which need to be further improved to satisfy all the desired properties of an e-voting system. In addition, this review identifies the importance of defining and evaluating the consensus algorithms used when designing a blockchain-based e-voting solution. Then this review highlights the requirement of designing e-voting systems by focusing on the demography and electoral systems of countries, to optimize the overall e-voting system. These identified gaps will be addressed through this research.

## **CHAPTER 3**

### **METHODOLOGY**

This chapter describes the research methodology followed during this research, the reasoning behind using the selected research method, the manner in which the research is conducted and steps taken during the research, and how the research questions are addressed through this research method. This chapter also provides a comprehensive description of the proposed solution.

#### **3.1. Research Methodology**

The constructive research approach is followed during this research to guide the research through the lifecycle. Lukka describes the constructive research approach as “*a research procedure for producing innovative constructions, intended to solve problems in real-world and to make a contribution to the theory of discipline in which it is applied*” (Lukka, 2003). In the constructive research approach, research is started with a real-world problem and trying to solve it through a new or refined solution and implementation, with both practical and theoretical contributions. The constructive research approach has been successfully used in the field of computer science, especially due to the fact that this approach assists and guides the process of developing solutions to existing practical problems.

There are several reasons for selecting the constructive research approach for this research. First, the problem addressed in this research requires a solution to a practical problem with a novel approach and since the constructive research approach is highly focused on developing novel solutions, the constructive approach is well suited. Also, since this research is focused on developing and implementing a new solution for the research problem while addressing the research gap, the process and the phases defined in the constructive approach are deemed suitable. Finally, the constructive research approach is a proven method in the domain of computer science and thus this research is also basing the constructive research approach.

#### **3.2. Selection of the Research Problem**

With the evolution of cryptocurrencies, blockchain technology which is the foundation of crypto-currencies also gained significant research interest. There have been a number of researches on applying blockchain technology in various domains. One such domain is

electronic voting. Electronic voting systems have raised concerns over security, privacy, auditability, and transparency. Since blockchain technology addresses most of these concerns by design, multiple researches have been performed on applying blockchain technology for designing e-voting systems. However, when evaluating the existing literature, it can be identified that most researches have been focused on using public blockchains. The drawback of using public blockchains for e-voting systems is the negative impact on the performance and the scalability of the system, as highlighted in the literature review chapter. The alternative approach is to use private blockchains and the number of researches performed on this aspect is limited. Thus, designing and developing an e-voting system on a private blockchain satisfying all desirable properties of an e-voting system is considered as the research problem. The desirable properties of an e-voting system are privacy, anonymity, coercion resistance, receipt freeness, accessibility, robustness, accuracy, eligibility, and verifiability.

The main hypothesis of this research is that introduction of a private blockchain-based e-voting system will satisfy all desirable properties of an e-voting system and improve time efficiency while reducing costs.

### **3.3. Analysis of the Research Problem and Domain**

Since elections span over multiple departments and involve a vast operation set, it is important to define the scope for this research. An electronic voting system consists of three major components as the service/storage layer, the client end layer, and the communication layer. For the scope of this research, only the service layer will be considered. From the numerous stages in the election process, only the voting stage and vote tallying stage are considered for the scope of the research. In addition, this research will only consider the elections of Sri Lanka.

#### **3.3.1. Private Blockchain Platform Selection**

The selection of the appropriate private blockchain platform is one of the critical decisions for this research. For this selection decision, the main consideration criteria include supported features, smart contract support, stability, community support, performance, and security aspects of the private blockchain platform. After evaluating several platforms, the Hyperledger Fabric was selected as the blockchain platform for this research.

The Hyperledger Fabric is a private permissioned blockchain infrastructure (Androulaki et al., 2018), which is supported by the Linux foundation. Some of the important features of the Hyperledger Fabric platform influencing this selection include the rich and well-designed

permission model, high-performance optimizations, stability and usage across multiple domains, pluggable consensus algorithm architecture, support for powerful Smart Contract development, scalability support with docker based deployment architecture.

Following are descriptions of some ‘Hyperledger Fabric’ specific terminology that will be used during this document.

Table 1: Hyperledger Fabric Terminology

Peer Node	Peer is the basic component of the Hyperledger network and a peer node is responsible for storing its own copy of the ledger and storing and executing ChainCodes.
Orderer Node/ Ordering Service	Orderer is responsible for maintaining the order of transactions and delivering transactions to peer nodes. Orderer uses a consensus mechanism to achieve this functionality.
Membership Service Provider (MSP)	MSP is responsible for the authentications and authorization of the members of the network. MSP in Hyperledger Fabric is based on Certificate Authority where certificates are issued to members and those certificates are used to identify and control access based on definitions.
Organization	The organization is a conceptual concept which maps to a particular party involved in the blockchain network. An organization owns a set of peers and a single MSP. Multiple organizations can use the same Hyperledger Fabric network while sharing some information and while maintaining private information as well.
Channel	Channel is responsible for partitioning the network into sub-sections, where a set of peers will be connected. In a Hyperledger Fabric network, there can be multiple channels and members connecting to a particular channel can access the data shared in that channel.
ChainCode	ChainCode contains the Smart Contracts in Hyperledger Fabric context. A ChainCode can be packaged and deployed to the Hyperledger Fabric blockchain.

Ledger	In Hyperledger Fabric the Ledger consists of two parts as the blockchain and a database called world state. This world state database holds the latest values of the blockchain and acts as a faster data access method. There are two types of word state databases supported by Hyperledger Fabric as LevelDB and CouchDB.
Hyperledger Fabric Gateway SDK	Hyperledger provides SDKs to interact with the blockchain and access Smart Contracts in multiple programming languages.

The Hyperledger Fabric platform supports a pluggable consensus algorithm architecture. The consensus algorithm in a Blockchain platform is responsible for generating an agreement on transactions among the nodes to generate a valid block. In the Hyperledger Fabric platform, the component called the ‘Ordering service’ is responsible for this process.

With the current version of the Hyperledger Fabric platform of 2.2, the recommended algorithm for the ‘Ordering service’ is RAFT. Thus, for this research, the RAFT consensus algorithm is used. However, there is no direct impact on the proposed system with the selection of this consensus algorithm as the ‘Ordering service’ is completely independent of the functionalities of the proposed e-voting system.

### **3.3.2. Electoral System and Demography of Sri Lanka**

As identified in the literature review chapter, the electoral system and the demography of a country has a dependency on the final design of e-voting systems. Since this research is performed in the context of Sri Lankan elections, the functionality and the architecture of the e-voting system must be based on the context of Sri Lanka.

The proposed solution in this research still requires the element of voting centers. Due to the limitations of internet and technology accessibility and literacy in Sri Lanka, it is not possible to enable e-voting directly through devices owned by voters. Instead, as per the proposed solution, voters are required to vote in a voting center similar to the current system.

The electoral system of Sri Lanka consists of twenty-two electoral districts. Thus, when designing the architecture of the proposed system, this segregation is considered.



This chapter provided details on the research methodology followed for this research. The next chapter describes the proposed solution of the e-voting system while providing information on the reference implementation.

## **CHAPTER 4**

### **DESIGN AND IMPLEMENTATION**

This chapter provides an in-depth description of the design and the architecture proposed for a private blockchain-based e-voting solution for Sri Lanka with reference implementation details.

#### **4.1. User Flow of the Proposed E-voting System**

As previously mentioned, according to the proposed e-voting system, voters are required to cast their votes at the voting centers. Also as described previously, the scope of this research will mainly focus on the voting stage and the vote tallying stage of the elections. Thus, it is assumed that all eligible voters are added to the e-voting system previously. However, at the implementation level, necessary functionalities for adding eligible voters to the e-voting system are provided.

Figure 1 illustrates the user flow of the proposed e-voting system at the voting stage.

Once a voter enters the voting center, the voter is expected to provide unique identification. In the Sri Lankan context, this can be the national identity card or the passport. As a prerequisite to the elections, all eligible voters will be added to the e-voting system with this unique reference (national identity card number is used for the prototype design).

The election officer at the voting center will validate the identity provided and enter this unique reference to the e-voting system. The system will then validate the entered unique reference with the data in the ledger to ensure the validity of the provided reference. The provided reference will be considered valid if and only if the reference is available as an eligible voter and a vote is not cast yet. If invalid, the system will return a warning and the user will not be allowed to vote. If valid, then the system will return a temporary token which will be then assigned to the voter at the client end. Then the voter will cast the vote and this information will be sent to the e-voting system along with the temporary token.

The e-voting system will validate the temporary token and if valid then the vote will be stored in the ledger and the vote tallies will be updated. Finally, the system will generate and return a unique token. The voter can store this unique token and later use it to validate whether the vote has been considered.

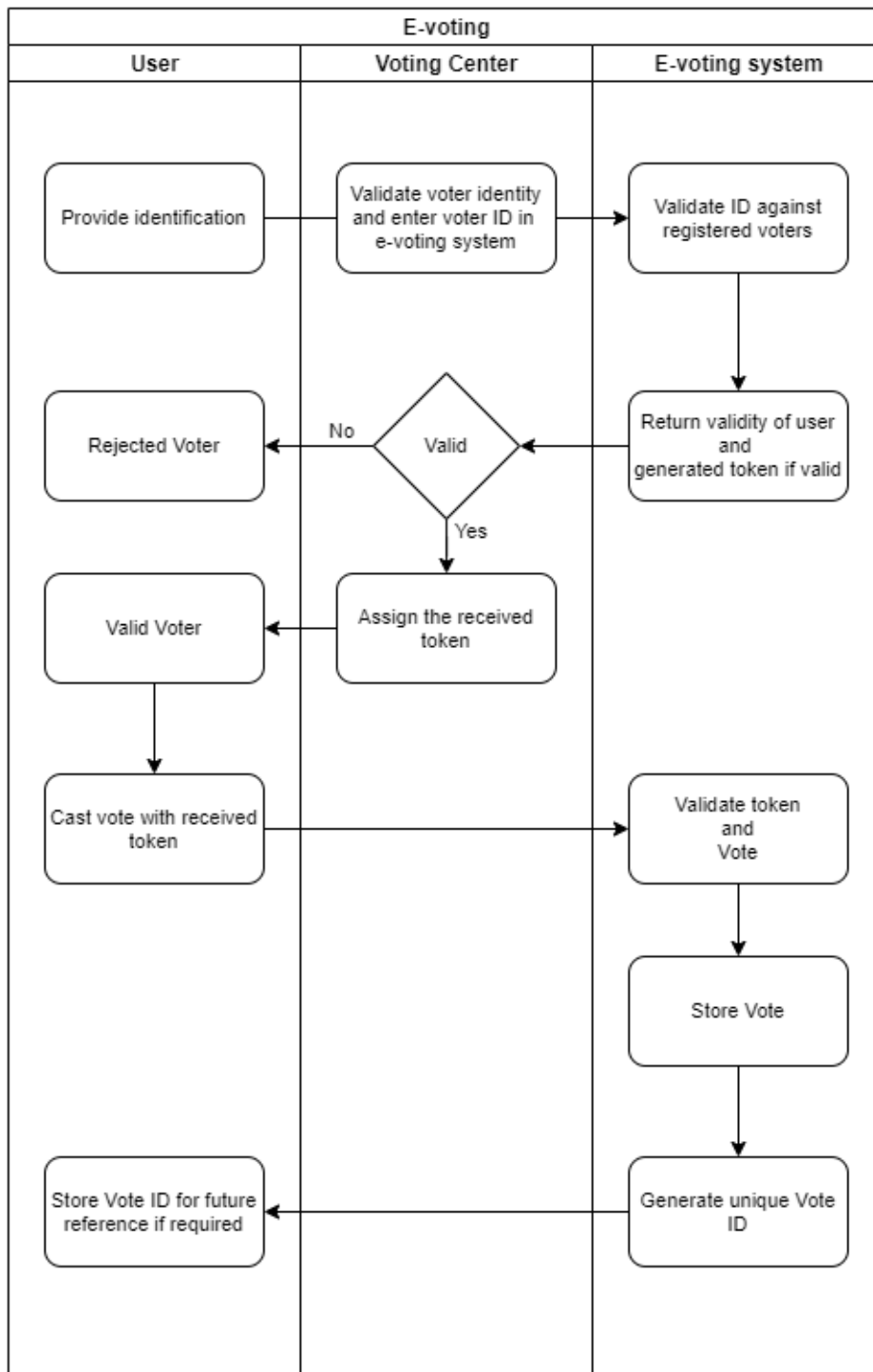


Figure 1: User Flow of the Proposed E-voting System

This is the basic flow at the voting stage and how the system achieves the required properties of an e-voting system is described later in this chapter.

## 4.2. System Components and Architecture

This section describes system components and the architecture of the proposed system, along with the justification for selections.

### 4.2.1. Hyperledger Fabric Network Design

The Hyperledger Fabric blockchain is configured under a single ‘Organization’ for the e-voting system. Since the e-voting system is a single system where all participating parties have access to common information in the system, this can be modeled as a single ‘Organization’ in the blockchain.

This ‘Organization’ consists of multiple ‘Peer’ nodes to handle the voter base. If necessary, the number of peers can be scaled vertically to handle the required voter load. Each ‘Peer node’ will have a ‘Ledger’ copy and a ‘World State’ DB instance as per the design of Hyperledger Fabric. The ‘CouchDB’ database is used for the ‘World State’ DB as it is the recommended choice for production-grade systems with Hyperledger Fabric. The e-voting ‘ChainCode’ consisting of multiple Smart Contracts, which contain all the logic of the e-voting system will be deployed at each of these ‘Peer’ nodes. This enables all the Peer nodes to participate in e-voting functionalities.

An ‘Ordering Service’ with multiple nodes will be configured to use RAFT as the consensus algorithm. The selection of the RAFT consensus algorithm is influenced by the recommendation from the Hyperledger Fabric documentation as it is the recommended algorithm at the time of this research.

A single ‘Channel’ will connect all the ‘Peer’ nodes and ‘Ordering Service’ nodes. Multiple Channels can be used to partition the network into subsections if required. However, for this proposed design, all the ‘Peer nodes’ should participate in the e-voting functionalities and the information needs to be shared across all these ‘Peers’. Thus a single ‘Channel’ is used to connect all the ‘Peer nodes’ and maintain the information flow.

There will be two ‘Certificate Authority’ instances acting as the ‘Membership Service Providers (MSP)’ of the ‘Organization’ nodes and the ‘Ordering Service’ nodes. This is as per the design of the Hyperledger Fabric platform where independent ‘Certificate Authorities’ are required for each of the ‘Organizations’ and the ‘Orderer Service’. These ‘Certificate Authority’ instances are used for authentication and authorization of the members in the network.

### **4.2.2. Application API Layer Design**

There is a service application layer that consists of REST web services, which act as the access points to the services provided by the Smart Contracts deployed in the Hyperledger Fabric network. These REST web services are developed using the Hyperledger Fabric SDK to interact with the blockchain. The Hyperledger Fabric SDK provides the features to communicate with the Smart Contracts in the network securely and efficiently. The REST web services enable simple integrations at the consumer end.

The REST services are grouped under two categories as admin services and client services. This separation enables the scaling of client services independently of admin services. Admin services provide the access to administrative functionalities such as election definition, voter registration, start and end voting periods, publishing results, etc. through REST endpoints. Client services provide access to functionalities related to the voting stage such as voter ID validation, storing votes, generating vote reference, etc. through REST endpoints. Administrative clients connect with the admin service cluster and the voting center clients connect with the Client service cluster.

These REST services are developing NodeJS. The main reason for selecting NodeJS is because the default SDK of the Hyperledger Fabric is provided for NodeJS.

It is important to understand that this application layer does not contain any functionalities or logic related to the e-voting system, and only act as a communication layer with the blockchain and Smart Contracts.

The following figure illustrates the high-level architecture of the proposed e-voting system.

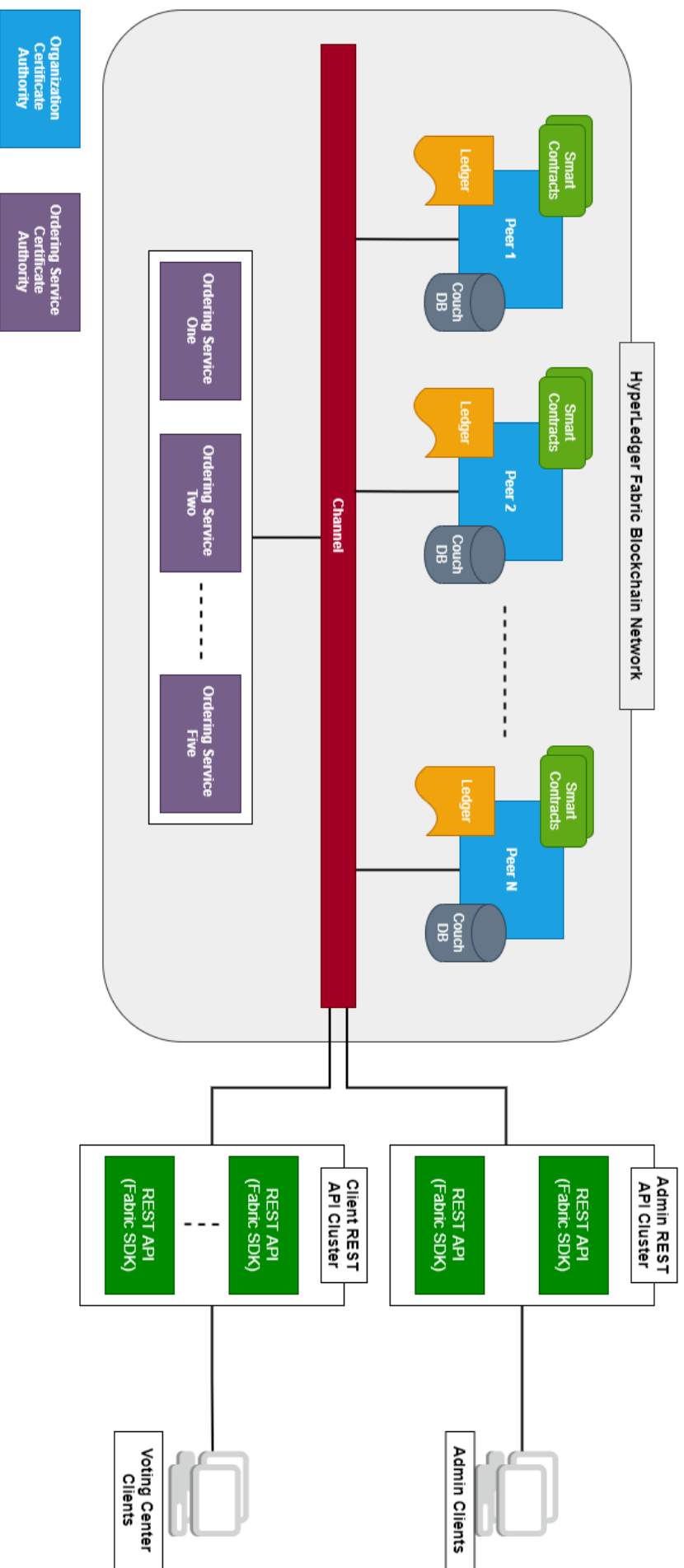


Figure 2: Architecture of the Proposed E-voting System

### 4.2.3. Deployment Aspects of the E-voting System

The Hyperledger Fabric blockchain platform is designed as deployment-ready docker images. For each 'Peer' two docker containers are required as one for the 'Peer' and the other for the 'CouchDB'. The 'Ordering Service' instances will be also deployed as docker containers. For the two 'Certificate Authorities (MSP)', another two docker containers will be created.

Additionally, the admin REST services and client REST services will be deployed in separate docker containers.

Since a large number of docker containers are required for this proposed system, for managing purposes a container orchestration tool like Docker Swarm or Kubernetes can be used.

### 4.2.4. Smart Contract Design

This section describes the design of the e-voting Smart Contracts of the proposed system. These Smart contracts contain complete logic related to voting processes.

In Hyperledger Fabric, the information to be persisted into the ledger is defined as JSON objects. The following table contains the JSON objects which are persisted in this e-voting system. The complete information on these objects is added under 'Appendix A: JSON Data Objects in Smart Contracts'.

Table 2: Data Structures for E-voting in Smart Contracts

ElectionType	Represent types of elections in Sri Lanka.
Election	Represent an election in Sri Lanka.
PollingDivision	Represent a polling division in Sri Lanka.
PollingSection	Represent a polling section in Sri Lanka.
VotingCenter	Represent a leaf-level voting center.
PoliticalParty	Represent a political party in the country.
PoliticalPartyElective	Represent a political party for an electoral section. Elective code refers to a unique segment where candidates will

	campaign for.
Candidate	Represent an election candidate.
RegisteredVoter	Represent a registered voter.
VoterMetaRecord	Represent a voting status of a voter.
CandidateTallyVC	Hold vote counts against candidate number at the voting center level.
CandidateTallyPollSec	Hold vote counts against candidate number at the Polling section level.
CandidateTallyPollDiv	Hold vote counts against candidate number at Polling Division level.
PoliticalPartyTallyVC	Hold vote counts against the political party at the voting center level.
PoliticalPartyTallyPollSec	Hold vote counts against the political party at the Polling Section level.
PoliticalPartyTallyPollDiv	Hold vote counts against the political party at the Polling Division level.
PollingSectionVoteTally	Hold vote counts at a polling section level.
PollingDivisionVoteTally	Hold vote counts at a polling section level.

The Smart Contracts functions are implemented to perform the required functionalities to achieve the desired properties of an e-voting system. The following is the voter validation function in the Smart Contract that is used to verify the provided voter Id to be eligible and to generate the temporary token for eligible voters.



```

//Validate the voter to be registered and valid then return a temp token -
voterId is transient

func (t *EVotingSmartContract) ValidateVoter(ctx
contractapi.TransactionContextInterface, voterId string, electionCode
string, votingCenterCode string) (*VoteValidateResult, error) {
    voterMeta, err := t.QueryVoterMetaRecordByVoterId(ctx, voterId,
electionCode, "ENABLED")
    validateResponse := &VoteValidateResult{
        Status: "INVALID",
    }
    if err != nil || voterMeta == nil {
        return validateResponse, nil
    }
    if voterMeta.VotingCenterCode != votingCenterCode {
        fmt.Println("Valid voter incorrect vote center: " +
voterMeta.VoterId + " " + voterMeta.VotingCenterCode)
        return validateResponse, nil
    }

    fmt.Println("Valid voter meta record found: " + voterMeta.VoterId)

    //create time based hash
    txTime, err := ctx.GetStub().GetTxTimestamp()
    if err != nil {
        return validateResponse, nil
    }
    currentTime := time.Unix(txTime.Seconds, int64(txTime.Nanos))
    tempToken := GetCryptoHash(currentTime.String() + voterId)

    voterMeta.VotingStatus = "VOTE_PENDING"
    voterMeta.TempToken = tempToken

    voteMetaByte, err := json.Marshal(voterMeta)
    if err != nil {
        return nil, err
    }
    voterMetaKey := voterId + "_" + electionCode
    err = ctx.GetStub().PutState(voterMetaKey, voteMetaByte)
    if err != nil {
        return nil, err
    }

    validateResponse.Status = "VALID"
    validateResponse.VotingToken = tempToken

    return validateResponse, nil
}

```

## 4.2.5. REST API Design

This section describes the design of the REST API of the e-voting system, which provides the access to e-voting functionalities. This API service layer is only responsible for invoking the relevant functions in Smart Contracts, and there is no business logic related to e-voting implemented at this layer.

There are two main REST API endpoints used for the voting stage.

### Voter Validation

Endpoint	
GET	/evote-api/{Election_Code}/{Voting_Center_Code}/voters/{Voter_ID}
URL Parameters	
Election_Code	Unique code identifying election
Voting_Center_Code	Unique voting center code
Voter_ID	Unique voter identity (National Identity number in prototype)
Response	
data.status	Validity of the request
data.votingToken	Temporary token to be passed with vote request
Sample Response	
<pre>{   "data": {     "status": "VALID",     "votingToken": "2a690492a9c185ac3f65f23b3de794c37269b14b"   } }</pre>	

### Cast Vote

Endpoint	
POST	/evote-api/votes
Request Body	

votingToken	Temporary token received through voter validation
partyCode	Unique code of the election party
candidateNos	List of candidates numbers
Sample Request Body	
<pre>{   "votingToken": "2a690492a9c185ac3f65f23b3de794c37269b14b",   "partyCode": "AAA",   "candidateNos": [     "24"   ] }</pre>	
Response	
data.status	Status of the request
data.voteToken	UUID token generated for the vote
Sample Response	
<pre>{   "data": {     "status": "SUCCESS",     "votingToken": "ab659712a594ccde6ea39b0fba0cbeaf89adae11"   } }</pre>	

The following set of REST endpoints provides post-election functionalities of vote verification and election results.

### **Verify Vote**

Endpoint	
GET	/evote-api/votes/{Vote_Token}
URL Parameters	
Vote_Token	UUID token generated for the vote received at vote casting
Response	

data.status	Validity of the vote token
data.voterId	Unique voter identity used for voting
data.electionCode	Election code
data.token	Vote token
data.votingCenterCode	Voting center code
data.votedTimestamp	Voted time
<b>Sample Response</b>	
<pre> {   "data": {     "status": "VALID",     "voterId": "198000505000",     "electionCode": "PARLMNT_2025",     "token": "ab659712a594ccde6ea39b0fba0cbeaf89adae11",     "votingCenterCode": "01_A_0",     "votedTimestamp": "2021-09-04T09:51:16Z"   } } </pre>	

### **Election Party Level Results for Polling Division**

<b>Endpoint</b>	
GET	/evote-api/election-results/{ <i>Election_Code</i> }/{ <i>Division_Code</i> }
<b>URL Parameters</b>	
Election_Code	Unique code identifying election
Division_Code	Unique code of the polling division
<b>Response</b>	
data.status	Status of the request
data.voteToken	UUID token generated for the vote
data.partyResults[*].partyCode	Party Code

data.partyResults[*].votes	Number of votes
Sample Response	
<pre> {   "data": {     "status": "VALID",     "electionCode": "PARLMNT_2025",     "partyResults": [       {         "partyCode": "AAA",         "votes": 153       },       {         "partyCode": "BBB",         "votes": 226       },       {         "partyCode": "CCC",         "votes": 186       }     ]   } } </pre>	

### **Election Party Level Results for Polling Section**

Endpoint	
GET	/evote-api/election-results/{ <i>Election_Code</i> }/{ <i>Division_Code</i> }/{ <i>Section_Code</i> }
URL Parameters	
Election_Code	Unique code identifying election
Division_Code	Unique code of the polling division
Section_Code	Unique code of the polling section
Response	
data.status	Status of the request
data.voteToken	UUID token generated for the vote
data.partyResults[*].partyCode	Party Code

data.partyResults[*].votes	Number of votes
Sample Response	
<pre> {   "data": {     "status": "VALID",     "electionCode": "PARLMNT_2025",     "partyResults": [       {         "partyCode": "AAA",         "votes": 63       },       {         "partyCode": "BBB",         "votes": 129       },       {         "partyCode": "CCC",         "votes": 98       }     ]   } } </pre>	

**Election Candidate Level Results for Polling Division**

Endpoint	
GET	/evote-api/election-results/{Election_Code}/{Division_Code}/{Party_Code}
URL Parameters	
Election_Code	Unique code identifying election
Division_Code	Unique code of the polling division
Party_Code	Unique code of the election party
Response	
data.status	Status of the request
data.voteToken	UUID token generated for the vote
data.partyResults[*].partyCode	Party code

data.partyResults[*].votes	Number of votes
data.partyResults[*].candidateResults[*].candidateNo	Candidate number
data.partyResults[*].candidateResults[*].votes	Number of votes for the candidate
<b>Sample Response</b>	
<pre> {   "data": {     "status": "VALID",     "electionCode": "PARLMNT_2025",     "partyResults": [       {         "partyCode": "BBB",         "votes": 2,         "candidateResults": [           {             "candidateNo": "3",             "votes": 41           },           {             "candidateNo": "5",             "votes": 28           }         ]       }     ]   } } </pre>	

**Election Candidate Level Results for Polling Section**

<b>Endpoint</b>	
GET	/evote-api/election-results/{ <i>Election_Code</i> }/{ <i>Division_Code</i> }/{ <i>Section_Code</i> }/{ <i>Party_Code</i> }
<b>URL Parameters</b>	
Election_Code	Unique code identifying election
Division_Code	Unique code of the polling division

Section_Code	Unique code of the polling section
Party_Code	Unique code of the election party
<b>Response</b>	
data.status	Status of the request
data.voteToken	UUID token generated for the vote
data.partyResults[*].partyCode	Party Code
data.partyResults[*].votes	Number of votes
data.partyResults[*].candidateResults[*].candidateNo	Candidate number
data.partyResults[*].candidateResults[*].votes	Number of votes for the candidate
<b>Sample Response</b>	
<pre> {   "data": {     "status": "VALID",     "electionCode": "PARLMNT_2025",     "partyResults": [       {         "partyCode": "BBB",         "votes": 2,         "candidateResults": [           {             "candidateNo": "3",             "votes": 41           },           {             "candidateNo": "5",             "votes": 28           }         ]       }     ]   } } </pre>	



In addition to the above main API endpoints, the e-voting system consists of the following set of REST endpoints to provide access to admin functionalities.

GET	/evote-api/electionTypes	Get election types
GET	/evote-api/pollingDivisions	Get polling divisions
GET	/evote-api/electionParties	Get election parties
GET	/evote-api/{Party_Code}/candidates	Get candidates of a party

### **4.3. Achieving Desired Properties of an E-voting System**

The proposed e-voting system in this research attempts to achieve the desired properties of an e-voting system, which were discussed earlier in this document. Each of these properties and how those properties are addressed in the proposed solution is discussed below.

#### **4.3.1. Anonymity and Privacy**

This property of e-voting means that a ballot cannot be linked to a voter and sensitive voting information should not be accessible to unwanted parties.

In this proposed system, a connection between the personal identification information of the voter and the voted ballot will not be maintained. Even at the vote casting stage, the voted ballot will be passed along with a token and this token will not contain information for tracking ballots back to the voters. Thus, there it will not be possible to link a ballot with its voter, ensuring the anonymity property.

In this proposed system, once the vote information is at the service layer, access to this information will be controlled through a permission model, ensuring undesired access and exposure are prevented. However, this research only focuses on the service layer and the privacy breaches at the client layer and the communication layer need to be addressed separately. With the proposed solution, since the existing voting center structure is still maintained, we can assume that privacy breaches at the client layer will be at a minimum.

#### **4.3.2. Coercion Resistance**

This property of e-voting means a coercer cannot extract information on what way a voter has voted.

As mentioned in the previous section, this proposed system, will not maintain a connection between the personal identification information of voters and respective voted ballots. Also, it is not possible to extract voter information from the token used at the voting stage. The ‘Vote ID’ provided to the voter at the end of the process, will only allow users to ensure that the respective vote is accounted, but will not provide information on the voting selection. Thus, a coercer cannot extract on the voting selection of a voter ensuring the coercion resistance property.

### **4.3.3. Receipt-freeness**

This property of e-voting means that a voter cannot prove his/her selection by creating a receipt.

In this proposed system, once the voting is completed the voter will only be presented with a unique ‘Vote ID’. This ‘Vote ID’ enables the user to ensure the respective vote has been accounted for in the system by receiving meta-information regarding the vote. However, this will not contain information on the voting selections. Thus, users will not be able to prove their selections by creating receipts ensuring the receipt-freeness of the system.

### **4.3.4. Accessibility**

This property of e-voting means that eligible voters should be able to access the system and vote conveniently.

In this proposed system, once the eligible voters are added to the system as a prerequisite, then such voters will be able to cast their votes at respective voting centers. The system will only validate the unique ID of the voter and ensures any registered user is able to vote, thus satisfy the accessibility property.

### **4.3.5. Accuracy**

This property of e-voting means that the system should guarantee that all ballots are accurately counted.

In this proposed system, once the ballot information is received at the Smart Contract layer, it will ensure the vote is accurately counted. For this, the calculation logic should be correctly implemented.

#### **4.3.6. Eligibility**

This property of e-voting means that the voters must be registered according to a predefined criterion and voting is allowed only for such registered users.

In this proposed system, once the eligible voters are added to the system as a prerequisite, then such voters will be able to cast their votes at respective voting centers. At the time of the voting, the system will validate the voter against the registered users in the system and also validate whether the unique id (National Identity Card number) representing the voter is not voted already. The officials at the voting center are still responsible for validating the provided unique id belong to the voter. This process ensures the eligibility property of an e-voting system.

#### **4.3.7. Verifiability**

This property of e-voting means that voters should be able to verify that his/her vote was accounted for in the final result.

In this proposed system, once the voter casts the vote, a unique 'Vote ID' will be provided to the user. With this 'Vote ID', users will be able to retrieve meta-information about their vote to verify that the vote is accounted into the counting process. This meta-information will include the data such as the voting time, voting center, voter unique id. However, the actual voting information cannot be retrieved by this 'Vote ID'. With this functionality, this system satisfies the verifiability property of an e-voting system to a certain extent.

#### **4.3.8. Robustness of the election**

This property of e-voting means that the system should be robust against active/passive attacks and faults.

This research is only focused on the service layer of the e-voting system. Thus, satisfying the robustness of the e-voting system is not feasible within the defined scope. However, the robustness of the service layer is satisfied in the proposed system through the usage of blockchain technology.

## CHAPTER 5

### EVALUATION AND RESULTS

This chapter describes the evaluation criteria, process, and results derived through these evaluations. This chapter also critically analyzes the derived results to determine the strengths and weaknesses of the proposed system.

#### 5.1. Evaluation Criteria

The section presents the criteria identified for the evaluation considering the research problem, aim, and objects of this research. This includes evaluations against some of the applicable and measurable desired properties of e-voting systems as described in the Introduction chapter.

- **Performance of the proposed system architecture with reference to the prototype:**  
The performance of the prototype system needs to be evaluated with reference to the hardware and consensus algorithm to determine the throughput of the system.
- **Scalability of the system to handle the voter base of Sri Lankan elections:**  
With reference to the evaluated performance of the system, it should be evaluated that the proposed architecture can be scaled to manage the voting requirements of Sri Lankan elections.
- **Satisfying the Accuracy property of the prototype:**  
The prototype should be evaluated to ensure that such a system can satisfy the Accuracy property of an e-voting system, where the results of elections are correct.
- **Satisfying the Eligibility property of the prototype:**  
The prototype should be evaluated to ensure that such a system can satisfy the Eligibility property of an e-voting system, where only eligible users are allowed to vote.
- **Satisfying the Accessibility property of the prototype:**  
The prototype should be evaluated to ensure that such a system can satisfy the Accessibility property of an e-voting system, where all eligible users can cast their votes.
- **Satisfying the Anonymity property of the prototype:**  
The prototype should be evaluated to ensure that such a system can satisfy the Anonymity property of an e-voting system, where voters and their votes cannot be linked.

- **Satisfying the Verifiability property of the prototype:**

The prototype should be evaluated to ensure that such a system can satisfy the Verifiability property of an e-voting system, where later voters can ensure their votes are accounted for the final result.

- **Applicability of private blockchains for designing e-voting systems:**

Finally, it should be evaluated that the private blockchain can be used for designing e-voting systems, based upon the results from all of the above-mentioned evaluation criteria.

## 5.2. Evaluation Methodology and Approach

The evaluation method used for this research is experimental evaluation. During the evaluation, the developed prototype and the architecture will be tested to ensure the desired properties are satisfied and to determine the performance of the system to set as a baseline for future researches.

### 5.2.1. Data for Evaluation

In order to evaluate the prototype system, the system should be populated with data before the simulation process. The required data elements and the generation methods are described in the following table.

Table 3: Data Elements and Generation Methods for Evaluation

<b>Data Element</b>	<b>Description</b>
Electoral district	Electoral districts of Sri Lanka pre-populated into the system.
Polling division	Polling divisions under each electoral district of Sri Lanka pre-populated into the system.
Voting center	Voting centers under each polling division pre-populated with random values based on the scenario simulated.
Election	A dummy parliamentary election is defined in the system.

Political parties	Ten dummy political parties are registered across all electoral districts.
Candidates of each party	For each political party, twenty-five dummy candidates are registered for each electoral district.
Eligible Voters	A set of dummy eligible voters pre-populated of the dummy election under each polling division. This number of voters is selected based on the scenario simulated.

### 5.2.2. Simulation Scenarios

For evaluating the criteria identified under the ‘Evaluation Criteria’ section, a set of identified election scenarios are simulated on the developed prototype system. This section illustrates the three scenarios used in the evaluation.

These scenarios are modeled considering the electoral statistics in Sri Lanka published on the website of the ‘Election Commission of Sri Lanka’ for the year 2019 (Election Commission of Sri Lanka, 2021). The total number of registered voters is 16,263,885 according to these reports.

For simulation of voting attempts, the ‘Apache JMeter’ tool is used to trigger requests to the REST API of the E-voting system. For each simulation, the number of eligible voters and their voting sections is pre-defined in a configuration CSV file.

For this simulation, a total of ten political parties are added to the system, named by the first ten letters of the English alphabet. For each party, a total of twenty-five candidates are defined. The voting data for each scenario is generated using random values, while a higher weight is given to ‘Party A’, ‘Party B’, and ‘Party C’. For the generation of these CSV feeds, Google App Scripts is used.

These values in the CSV file can later be used to compare against the results derived from the E-voting system. A part of the values of such CSV file is added under ‘Appendix B: Section from Sample Test Data’.

The following sections will describe the three simulation scenarios used during the evaluation. These three scenarios are identified to simulate different voting scenarios while considering the limitations on hardware resources available.

#### 5.2.2.1. Simulation Scenario One

The first simulation is performed considering a single electoral district of Colombo. The eligible voters and voting centers are as follows.

Table 4: Simulation Scenario One Details

<b>Data Element</b>	<b>Count</b>
Number of Electoral districts	1
Polling divisions	15
Voting Centers per Section	3
Total Voting Centers	45
Eligible voters	450
Malicious voters	50

#### 5.2.2.2. Simulation Scenario Two

The second simulation is performed considering ten electoral districts. The eligible voters and voting centers are as follows.

Table 5: Simulation Scenario Two Details

<b>Data Element</b>	<b>Count</b>
Number of Electoral districts	10
Voting Sections	49
Voting Centers per Section	5
Total Voting Centers	245

Eligible voters	1225
Malicious voters	175

### 5.2.2.3. Simulation Scenario Three

The third simulation is performed considering twenty-two electoral districts. The eligible voters and voting centers are as follows.

Table 6: Simulation Scenario Three Details

Data Element	Count
Number of Electoral districts	22
Voting Sections	160
Voting Centers per Section	3
Total Voting Centers	480
Eligible voters	2400
Malicious voters	300

### 5.2.3. Prototype Deployment Specifications

For the evaluation, the prototype E-voting system is deployed on a local machine with an i7-7500 CPU operating at 2.70 GHz with 8GB of RAM and Ubuntu 16.04 as the operating system.

For the evaluation, the developed E-voting Smart Contracts are deployed on a Hyperledger Fabric network scaled down to match the available hardware specifications of the testing machine.

For configuring the required Hyperledger Fabric network, the ‘test network’ provided by the Hyperledger Fabric framework is used. The used Hyperledger Fabric network consists of a single peer node and two ordering nodes along with certificate authorities of both organization



and the ordering service. All of these nodes were deployed in the above-mentioned local machine using Docker Compose.

The REST APIs which are developed using NodeJS with Node Express are also deployed on the same local machine, with a single instance of the system.

The following figure illustrates the Hyperledger Fabric network along with the API services used for the evaluation.

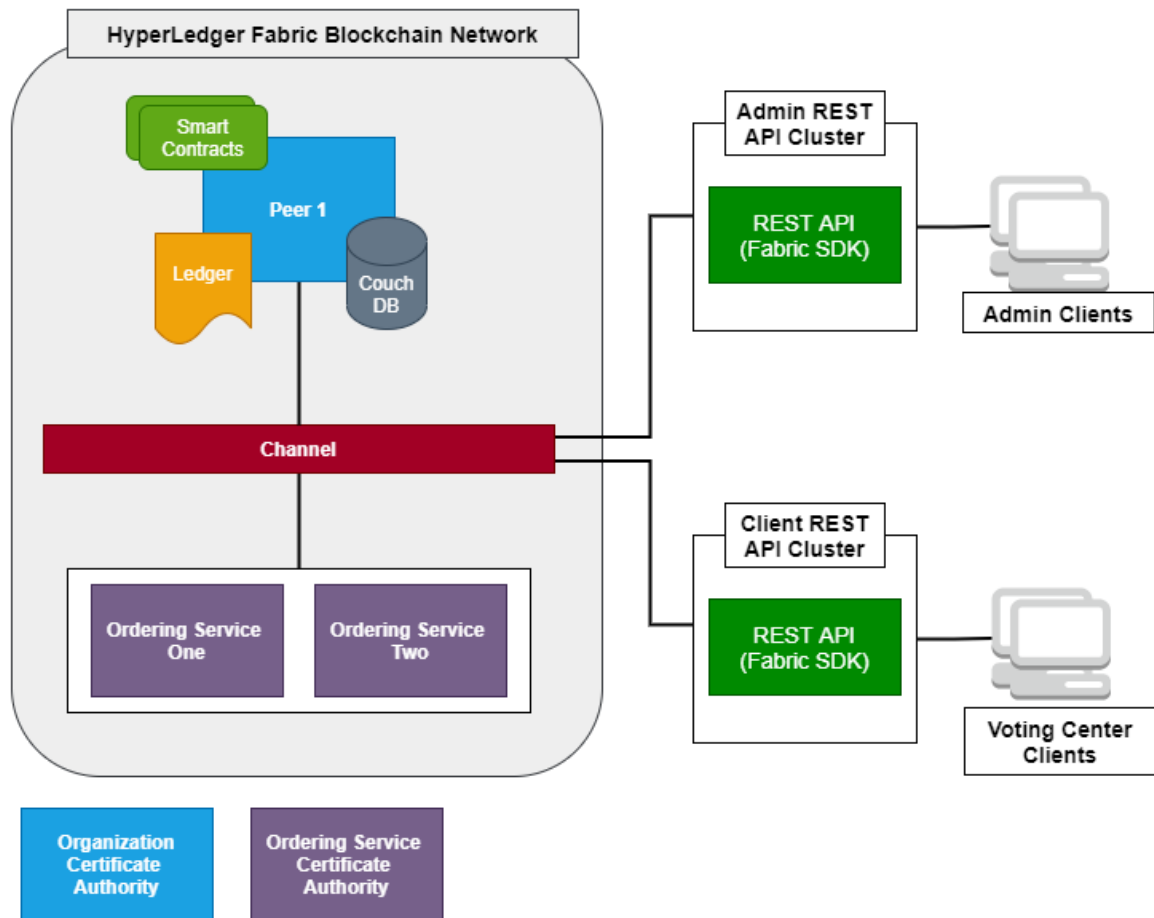


Figure 3: Architecture of the Prototype Network Used for Evaluation

### 5.3. Evaluation Results

This section describes the results collected through the evaluation. Additionally, under this section, these derived results will be critically analyzed to determine the suitability of the proposed system. The evaluation results are categorized under three main sections as the 'Voting Phase Results', 'Post Vote Validation Results', and finally 'Non-functional Results'.

The pre-vote stage actions such as setting up elections, an eligible voter for the election, eligible candidates, and political parties are not directly evaluated as per the scope defined for this research.

### 5.3.1. Voting Phase Results

This section describes the results obtained during the three voting scenarios described previously.

#### 5.3.1.1. Simulation Scenario One

After the simulation of scenario one, the total recorded valid vote count was 450. All fifty malicious votes were rejected by the system.

The following bar chart illustrated the distribution of the recorded votes by the system in red color and the random data used as the source in the color blue. This outcome confirms the e-voting system has correctly recorded all the valid voted simulated in this simulation scenario.

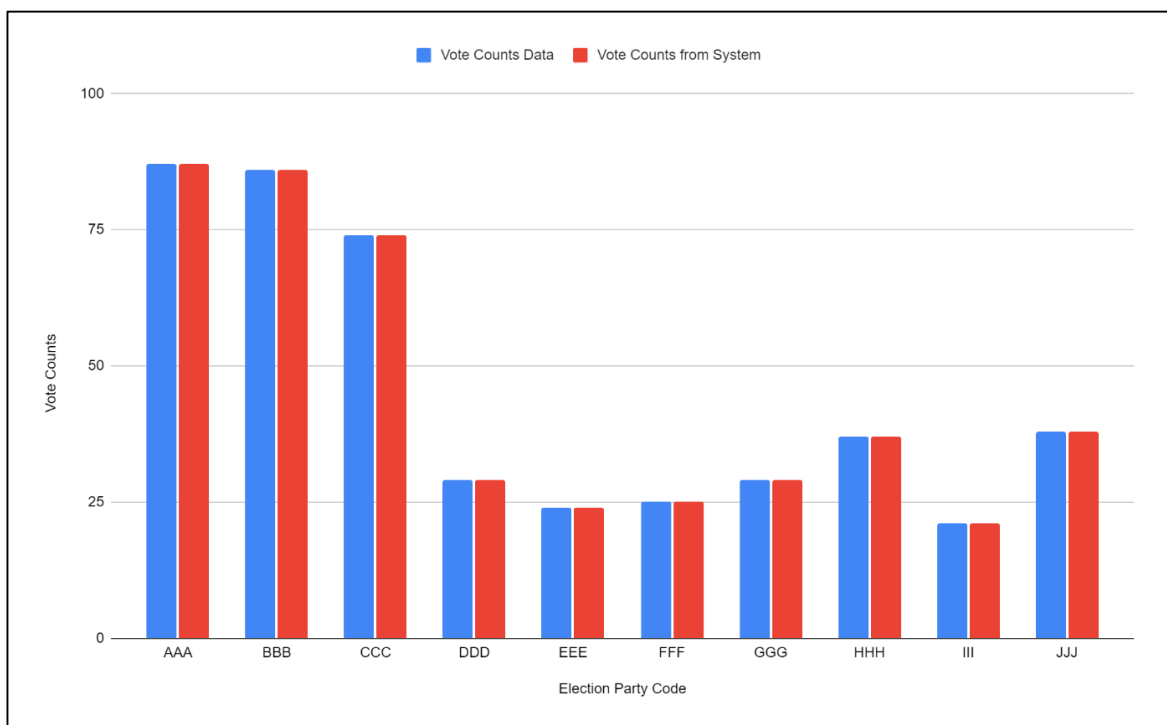


Figure 4: Bar Chart Representing Used Voting Data Against Recorded Valid Votes

The same evaluation can be performed at the candidate level and the results obtained at the candidate level also showed a hundred percent accuracy for this simulation scenario. The

following chart illustrates the votes in the data source in color blue with the votes recorded in the e-voting system in color red for the election ‘Party A’.

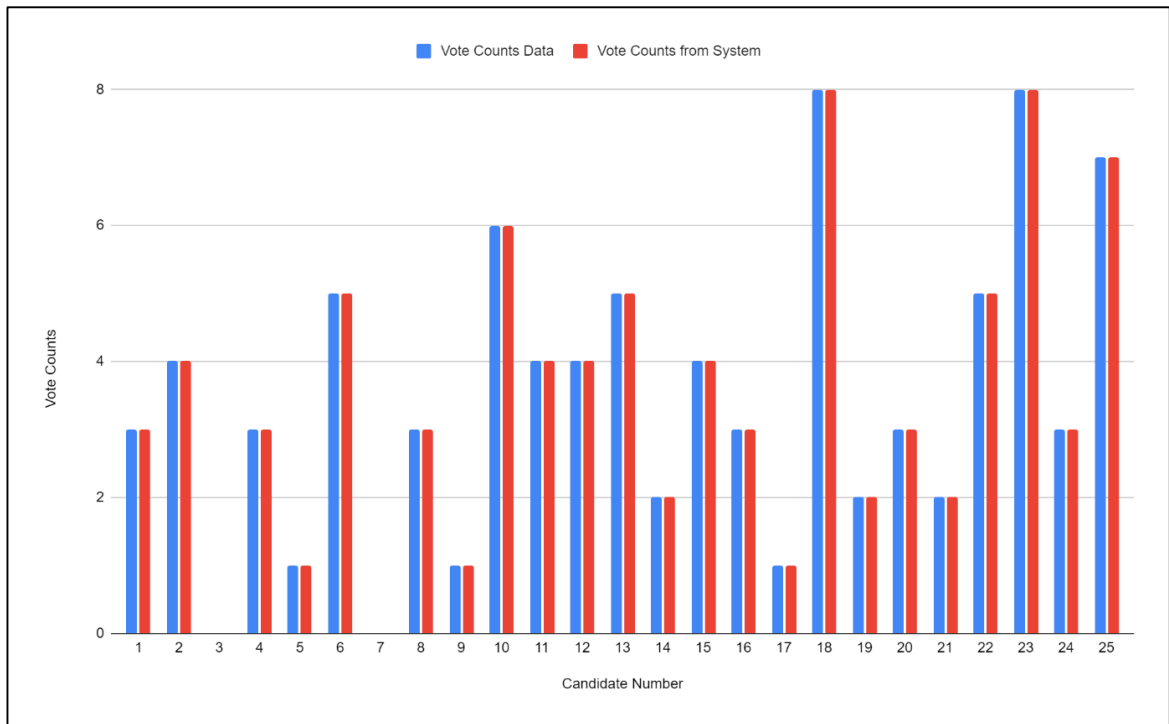


Figure 5: Bar Chart Representing Candidate Level Voting Data Against Recorded Valid Votes for Election ‘Party A’

### 5.3.1.2. Simulation Scenario Two

After the simulation of scenario two, the total recorded valid vote count was 1225. All 175 malicious votes were rejected by the system.

The following bar chart illustrated the distribution of the recorded votes by the system in red color and the random data used as the source in the color blue. This outcome confirms the e-voting system has correctly recorded all the valid voted simulated in this simulation scenario.

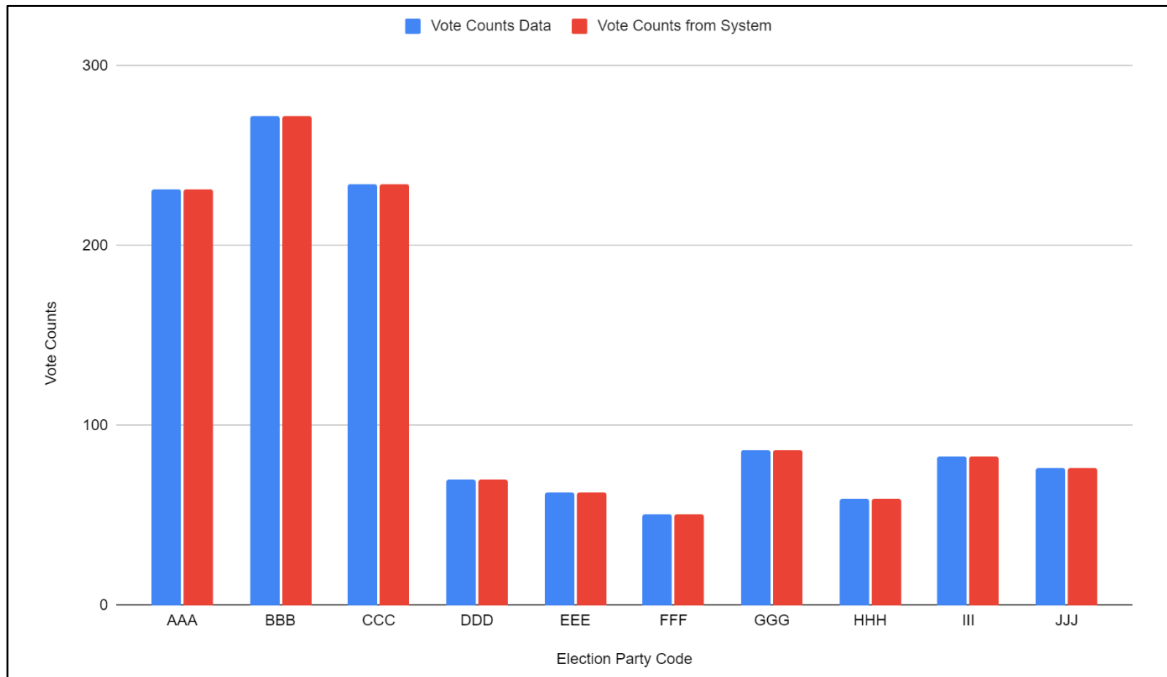


Figure 6: Bar Chart Representing Voting Data Against Recorded Valid Votes

The same evaluation can be performed at the candidate level and the results obtained at the candidate level also showed a hundred percent accuracy for this simulation scenario. The following chart illustrates the votes in the source data in color blue with the votes recorded in the e-voting system in color red for the election ‘Party A’.

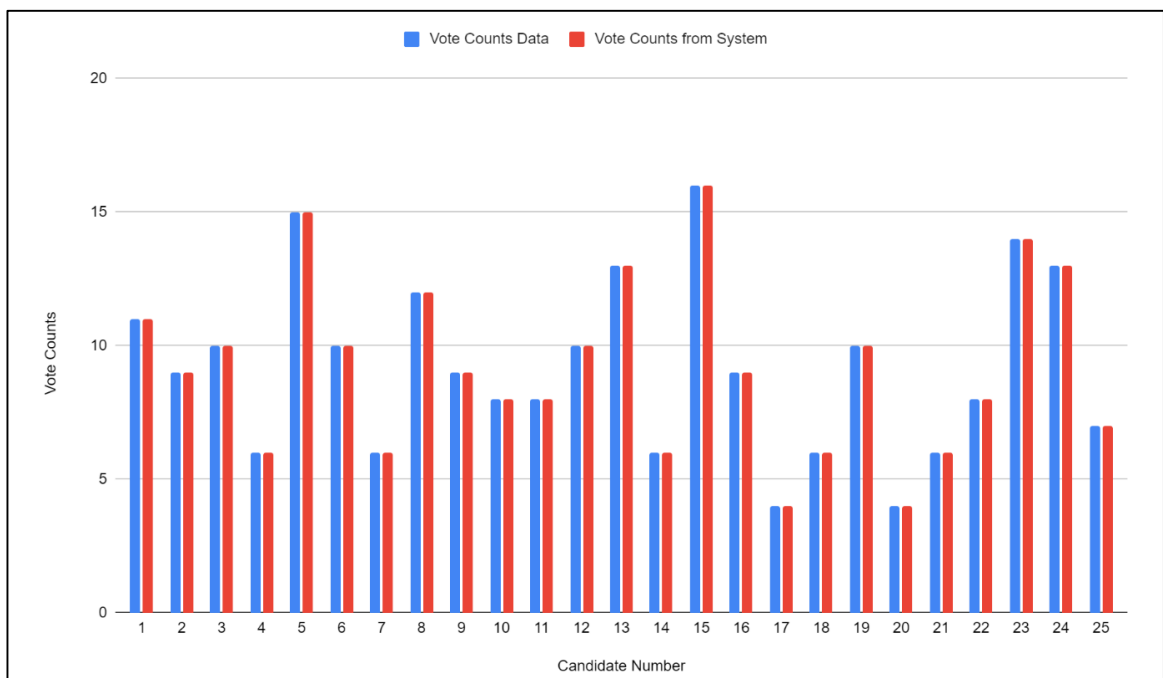


Figure 7: Bar Chart Representing Candidate Level Voting Data Against Recorded Valid Votes For Election ‘Party A’

### 5.3.1.3. Simulation Scenario Three

After the simulation of scenario three, the total recorded valid vote count was 2400. All 300 malicious votes were rejected by the system.

The following bar chart illustrated the distribution of the recorded votes by the system in red color and the random data used as the source in the color blue. This outcome confirms the e-voting system has correctly recorded all the valid voted simulated in this simulation scenario.

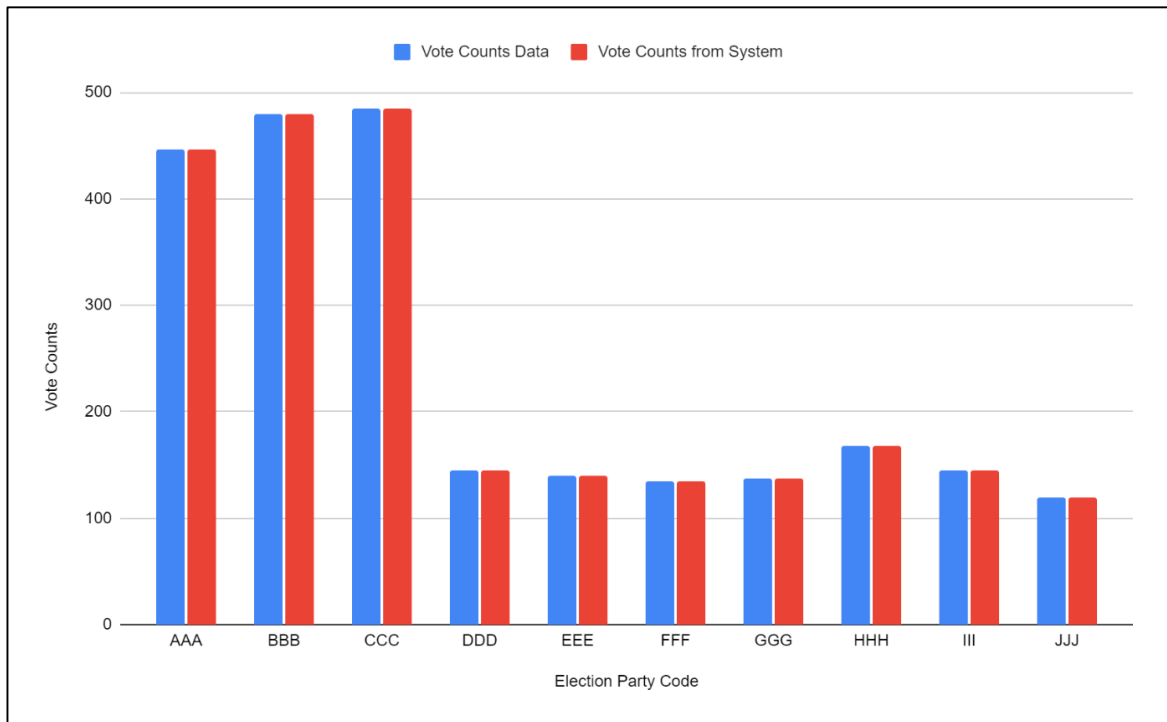


Figure 8: Bar Chart Representing Voting Data Against Recorded Valid Votes

The same evaluation can be performed at the candidate level and the results obtained at the candidate level also showed a hundred percent accuracy for this simulation scenario. The following chart illustrates the votes in the data in color blue with the votes recorded in the e-voting system in color red for the election 'Party A'.

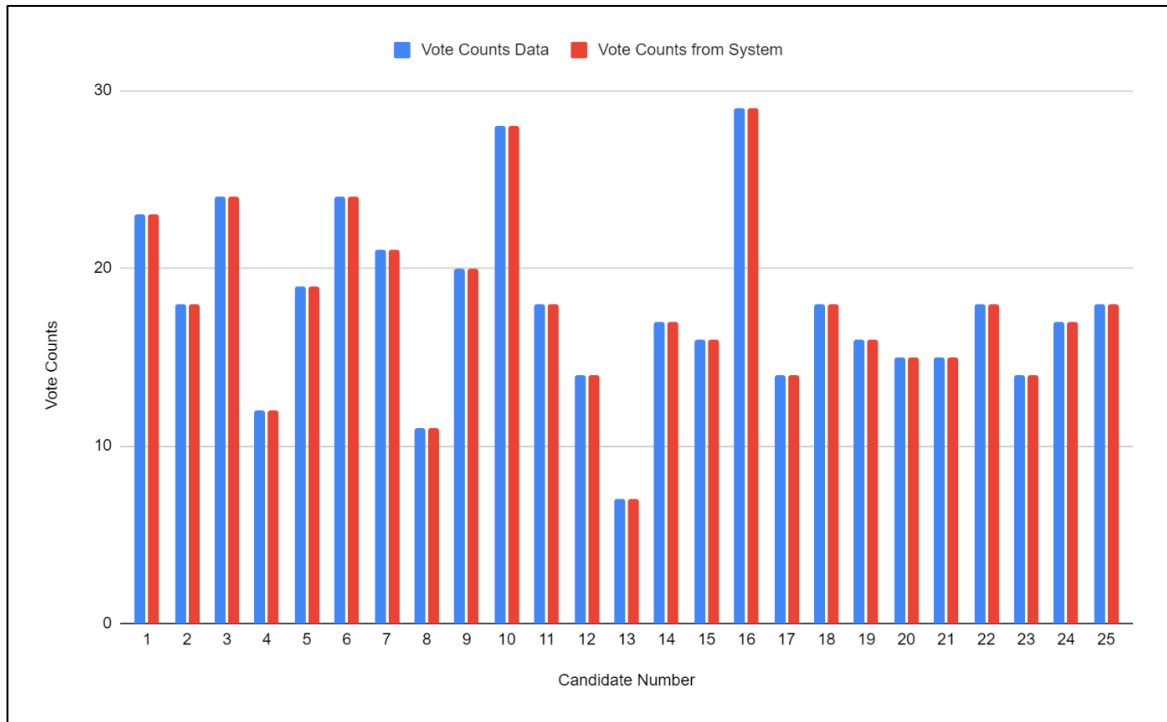


Figure 9: Bar Chart Representing Candidate Level Voting Data Against Recorded Valid Votes for Election ‘Party A’

The complete data set of recorded votes at the candidate level for scenario three is added under ‘Appendix C: Candidate Level Vote Data vs E-Voting Recorded Data for Scenario Three’.

#### 5.3.1.4. Analysis of Results

This section analyzes the derived voting results during the above described three simulation scenarios.

In all three simulation scenarios, the prototype system was able to achieve a hundred percent accuracy in the recorded votes with reference to the used sample data as illustrated in the above three sections. In all three simulation scenarios, the vote counts at both the election party level and the candidate level were exact matches against the voted counts returned from the e-voting system at the end of the simulation. Also in all three scenarios, all the malicious voter attempts were not considered for the results.

This behavior of the e-voting system ensures that the prototype was able to satisfy the ‘Accuracy’ property of an e-voting system.

As per the evaluation results, all the eligible voters were able to cast votes successfully in all three scenarios. Also, all the non-eligible voting attempts resulted in failures. Thus, it can be concluded that the prototype e-voting system satisfied the ‘Accessibility’ and ‘Eligibility’ properties of an e-voting system.

However, during the evaluation, when the concurrent user load was increased to eight, it was observed that the available hardware resources were not sufficient to continue the testing and the system started to fail. The CPU consumption of the host machine increased to a hundred percent. Thus, evaluating the robustness of the system with the prototype system with limited resources would not be feasible. In the ideal scenario, the components of the Hyperledger Fabric network system should be deployed in separate nodes and REST API services should also be deployed in separate nodes. Also, system orchestration tools such as Docker Compose or Kubernetes should be used to evaluate the Robustness property of the e-voting system.

### **5.3.2. Post Vote Validation Results**

This section describes the results obtained by invoking the vote verification endpoint of the e-voting system for the three voting scenarios described previously.

During the three voting process simulations, the ‘Vote ID’ tokens returned at the end of each vote casting are saved into a CSV file using ‘JSR223 Test Elements’ and Groovy language supported by the Apache JMeter tool. These tokens are saved against the Voter Identity number, voting center code from the test data, and the voted time.

These saved vote tokens are later played against the vote verification API endpoint of the e-voting system. Then the returned Voter Identity number, voting center code, and the voted time are compared against the values in the CSV file. When comparing the voted time, a two-second time tolerance has been introduced as the time recorded at the Smart Contract layer may differ slightly from the value saved in the CSV feed through the Groovy script upon receiving the response.

#### **5.3.2.1. Simulation Scenario One**

When playing the data collected from the vote process simulation of scenario one, 50 invalid ‘Vote ID’ tokens were added to the data source to evaluate the behavior of the system with malicious usage. The following table provides the collected results of this simulation on the prototype system.

Table 7: Simulation Scenario One Vote Validation Results

<b>Data Element</b>	<b>Count</b>
Number of valid Vote IDs used	450
Number of invalid Vote IDs used	50
Number of API responses with the correct metadata	450
Number of API responses with invalid status	50

As mentioned in the above table, for all the valid Vote ID tokens, the E-voting prototype system returned the response with the correct metadata. And for all invalid Vote ID tokens, the E-voting prototype system returned a response with failed status.

### 5.3.2.2. Simulation Scenario Two

When playing the data collected from the vote process simulation of scenario two, 300 invalid ‘Vote ID’ tokens were added to the data source to evaluate the behavior of the system with malicious usage. The following table provides the collected results of this simulation on the prototype system.

Table 8: Simulation Scenario Three Vote Validation Results

<b>Data Element</b>	<b>Count</b>
Number of valid Vote IDs used	1225
Number of invalid Vote IDs used	175
Number of API responses with the correct metadata	1225
Number of API responses with invalid status	175



As mentioned in the above table, for all the valid Vote ID tokens, the E-voting prototype system returned the response with the correct metadata. And for all invalid Vote ID tokens, the E-voting prototype system returned a response with failed status.

### 5.3.2.3. Simulation Scenario Three

When playing the data collected from the vote process simulation of scenario one, 300 invalid ‘Vote ID’ tokens were added to the data source to evaluate the behavior of the system with malicious usage. The following table provides the collected results of this simulation on the prototype system.

Table 9: Simulation Scenario Three Vote Validation Results

<b>Data Element</b>	<b>Count</b>
Number of valid Vote IDs used	2400
Number of invalid Vote IDs used	300
Number of API responses with the correct metadata	2400
Number of API responses with invalid status	300

As mentioned in the above table, for all the valid Vote ID tokens, the E-voting prototype system returned the response with the correct metadata. And for all invalid Vote ID tokens, the E-voting prototype system returned a response with failed status.

### 5.3.2.4. Analysis of Results

This section analyzes the derived voting results during the above described three simulation scenarios.

As highlighted above in all three scenarios, the prototype system achieved a hundred percent accuracy in all the three simulation scenarios for vote verification. Also, the system was able to identify the all-invalid requests and return a failed response correctly. This ensures that the e-voting system satisfies the ‘Accuracy’ property at the vote verification stage as well.

Also, this evaluation illustrates that the e-voting system is capable of returning the defined meta-information correctly. This behavior ensures that the e-voting system satisfies the ‘Verifiability’ property. Additionally, this ensures the compliance of the ‘Receipt-freeness’, ‘Coercion Resistance’ and the ‘Anonymity’ properties as only vote meta information is exposed through the endpoints of the e-voting system.

### **5.3.3. Non-Functional Results**

This section describes the non-functional results obtained during the voting simulation scenarios described previously. This evaluation is mainly focused on the resource consumption and the performance aspects of the system.

#### **5.3.3.1. Simulation Scenario One**

When simulating scenario one, multiple executions were performed by increasing the number of concurrent voters starting from one voter. This was done with the expectation of understanding the system behavior with a different number of concurrent users concerning the existing hardware configurations. When the voter count is increased beyond six, the existing resources were not sufficient and the system started to consume a hundred percent CPU and displayed slow response times. Thus this experiment was carried up to six concurrent voters only.

#### **Response Times of the System**

These six simulation executions were performed with summary response time tracking enabled in the JMeter tool to evaluate the probable response times with the increasing number of concurrent voters.

The following table contains the response times for the ‘Voter Validation’ step in milliseconds for different concurrent voters. As per this data, the average voting time increases with the number of concurrent voters.

Table 10: Simulation Scenario One ‘Validate Voter’ Step Response Times in Milliseconds with Different Concurrent Voters

Concurrent Voters	# Samples	Average (ms)	Min (ms)	Max (ms)	Std. Dev.	Throughput
1	500	3244	2945	3843	138.76	0.30826
2	500	3504	3340	4135	150.44	0.30874
3	500	3526	3262	4437	140.31	0.31339
4	500	3754	3378	4770	197.29	0.41984
5	500	4601	3647	6143	297.72	0.49475
6	500	4786	3589	6237	300.72	0.57853

The following table contains the response times for the ‘Vote’ step in milliseconds for different concurrent voters. As per this data, the average voting time increases with the number of concurrent voters.

Table 11: Simulation Scenario One ‘Vote’ Step Response Times in Milliseconds with Different Concurrent Voters

Concurrent Voters	# Samples	Average (ms)	Min (ms)	Max (ms)	Std. Dev.	Throughput
1	450	3341	2204	4317	181.63	0.29931
2	450	3359	2489	4703	174.86	0.30876
3	450	3572	2349	4728	191.29	0.31306
4	450	3904	2907	4880	192.84	0.42633
5	450	4834	4327	6389	278.97	0.49431
6	450	4857	4485	6471	290.06	0.56706

Additionally, a full response time recording for all the requests was performed for the scenarios of three concurrent voters and six concurrent voters to monitor response time variation over time. For this recording, three concurrent voters scenario was selected as it represents the median scenario and six concurrent voters scenario was selected as it is the maximum amount possible with available hardware resources. Also, this was only performed for scenario one voting as it was observed that the number of voters and voting centers doesn’t have a significant impact on the response times of the system. These charts as available under “Appendix D: Response Time Variation of All Requests for Scenario One”.

## Resource Consumption Statistics

The following figures illustrate the CPU utilization of the deployed machine with three concurrent users. There was no significant impact on the memory consumption of the system.

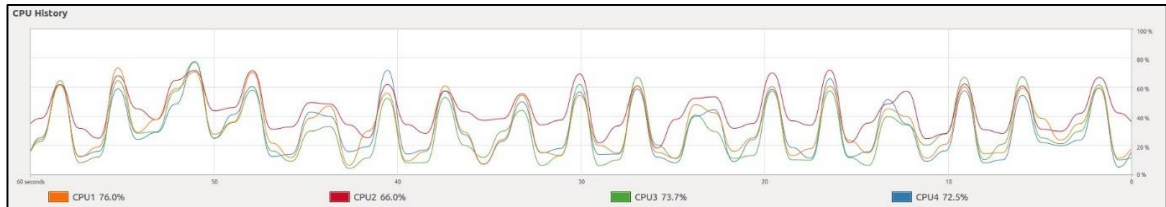


Figure 10: CPU Consumption for Scenario One with Three Concurrent Voters



Figure 11: Network Consumption for Scenario One with Three Concurrent Voters

The following figures illustrate the CPU and network utilization of the deployed machine with six concurrent users. There was no significant impact on the memory consumption of the system.

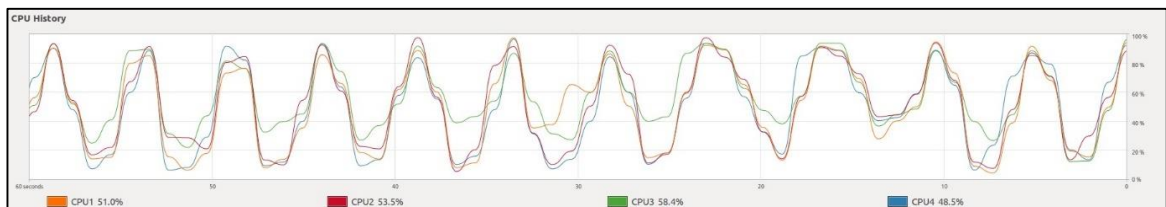


Figure 12: CPU Consumption for Scenario One with Six Concurrent Voters

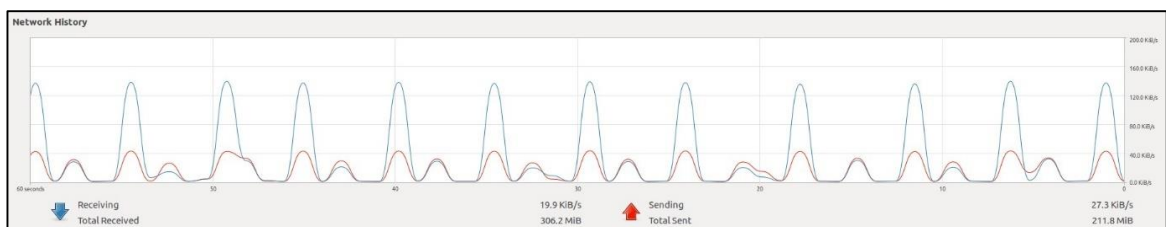


Figure 13: Network Consumption for Scenario One with Six Concurrent Voters

The CPU consumption has increased when the number of concurrent votes has increased, as highlighted in the following CPU graph. Also, the system couldn't handle a load of seven

concurrent users as mentioned previously. However, there was no significant impact on the memory and network consumptions during these simulations.

### 5.3.3.2. Simulation Scenario Two

The simulation was performed with three concurrent users as previous simulations covered the probably concurrent user handling capabilities of the prototype system. Usage of three concurrent voters was selected as it represents the median voter count the system is capable of handling with existing hardware configurations.

#### Response Times of the System

The following table contains the summary of the response times and throughput of the system for this simulation.

Table 12: Simulation Scenario Two Response Times in Milliseconds with Three Concurrent Voters

Label	# Samples	Average (ms)	Min (ms)	Max (ms)	Std. Dev.	Throughput
Validate	1400	3453	3284	4740	154.93	0.30567
Vote	1225	3665	2998	4911	173.82	0.31065

#### Resource Consumption Statistics

The following figures illustrate the CPU and network utilization of the deployed machine with these three concurrent users. There was no significant impact on the memory consumption of the system.

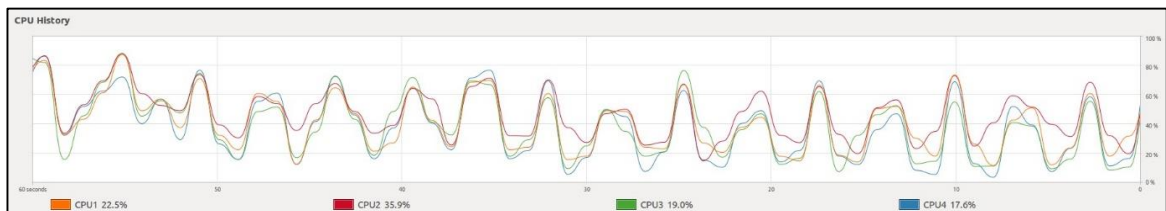


Figure 14: CPU Consumption for Scenario Two with Three Concurrent Voters

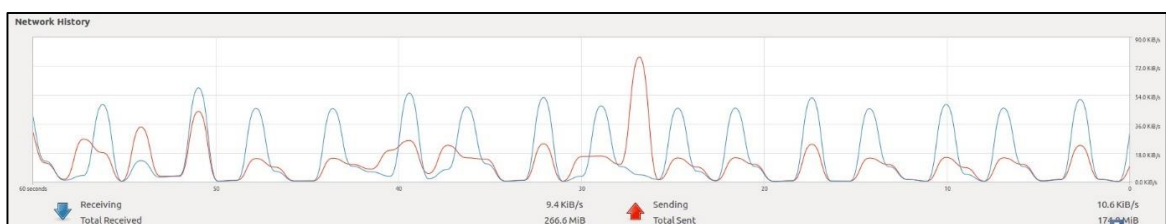


Figure 15: Network Consumption for Scenario Two with Three Concurrent Voters

### 5.3.3.3. Simulation Scenario Three

The simulation was also performed with three concurrent users as previous simulations covered the probably concurrent user handling capabilities of the prototype system. Usage of three concurrent voters was selected as it represents the median voter count the system is capable of handling with existing hardware configurations.

#### Response Times of the System

The following table contains the summary of the response times and throughput of the system for this simulation.

Table 13: Simulation Scenario Three Response Times in Milliseconds with Three Concurrent Voters

Label	# Samples	Average (ms)	Min (ms)	Max (ms)	Std. Dev.	Throughput
Validate	2700	3651	3096	4952	155.84	0.29891
Vote	2400	3711	3161	5035	170.98	0.30265

#### Resource Consumption Statistics

The following figures illustrate the CPU and network utilization of the deployed machine with these three concurrent users. There was no significant impact on the memory consumption of the system.

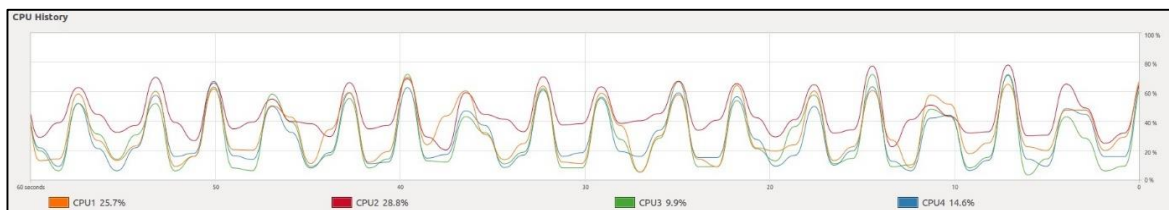


Figure 16: CPU Consumption for Scenario Three with Three Concurrent Voters



Figure 17: Network Consumption for Scenario Three with Three Concurrent Voters

#### **5.3.3.4. Analysis of Results**

When considering the response times of the prototype system during the simulation of the first scenario of five hundred voters, it can be observed that the response times varied with the number of concurrent users consuming the system. These statistics are heavily dependent on the limited hardware resources used for the evaluation of this system. However, considering the scenario of three concurrent voters it can be observed that the average response times of voting stages varies between three to four seconds.

When observing the response times of scenario two and scenario three with three concurrent voters, it can be noticed the response times and the throughput has not varied significantly based on the number of voters available in the system. Thus, it can be assumed that the system is capable of performing steadily under constant usage irrespective of the running duration, the total number of eligible voters, and voting centers.

When the number of concurrent voters was eight, the system started to display a hundred percent CPU consumption and unacceptable response times over ten seconds. Thus, to evaluate the scalability of the system to Sri Lankan voters based, the prototype system needs to be deployed with sufficient resources with separation of services. In an ideal scenario, the prototype should be deployed by separating the Hyperledger Fabric network nodes and the REST API services in a Kubernetes or a Docker Swarm cluster with horizontal scaling. However, the statistics collected during this evaluation can be used to forecast the feasibility of the Hyperledger Fabric blockchain network for the Sri Lankan context as per the below.

In the prototype setup, two peer nodes, two CouchDB nodes, two orderer nodes, two certificate authority nodes, and one REST service shared a single machine with four core CPUs and 8 GB memory. Even with the worst-case response time, assuming it takes 10 seconds for a single voter, the prototype system can process 450 voters within thirty minutes with only three concurrent users. It can be assumed that when these nodes are deployed separately with dedicated resources and with more nodes, these response times would be smaller. Considering it still takes 10 seconds, still, the system is capable of completing the 18000000 votes within 6.5 hours when there are 8000 concurrent users. With the current structure, there are over 2500 voting centers for the 160 voting sections in Sri Lanka. Thus, considering these stats the proposed e-voting should be able to scale to the context of Sri Lankan elections.

With this, the evaluation results satisfy evaluation criteria of the performance of the prototype system and the scalability of the system for the context of Sri Lankan elections.

## **5.4. Limitations of the Evaluation**

There are two main limitations of the evaluation approach used during this research.

The first limitation is the limited resources used in the evaluation. As described previously, the prototype system is deployed in a single machine using Docker Compose. This deployment introduces the limitations on the inability to allocate dedicated hardware resources to different nodes of the Hyperledger Fabric networks as well as REST services. While this limits the evaluation of scalability aspects, it also restricts monitoring the resource consumption of different components in the overall system. Thus, as the next step of the evaluation, the prototype can be deployed in a Kubernetes or a Docker Swarm cluster, with sufficient worker nodes with appropriate monitoring tools.

Then the second limitation is the lack of opinions from real users of the system. This evaluation is performed only by using randomly generated data. However, another aspect to capture would be the expert opinions on the architecture and the practical usage of the system.

## **5.5. Summary**

This chapter described the evaluation of the proposed private blockchain-based E-voting system. The evaluation criteria used for evaluation, consist of the satisfaction of desired properties of an e-voting system along with the performance and scalability of the proposed solution to the context of Sri Lanka.

As per the different simulation scenarios used to evaluate the prototype system and the derived results, the proposed e-voting system satisfies the desired e-voting properties of ‘Accuracy’, ‘Accessibility’, ‘Eligibility’, ‘Verifiability’, ‘Anonymity’, ‘Receipt-freeness’ and ‘Coercion Resistance’. Also, the performance and the scalability of the system can be considered to be sufficient for the Sri Lankan elections as per the analysis of the non-functional results.

Thus, this evaluation confirms that the Hyperledger Fabric can be used to develop a private blockchain-based e-voting system.



## **CHAPTER 6**

### **CONCLUSION AND FUTURE WORK**

This chapter focuses on discussing the conclusions of the research by elaborating how the research questions are addressed and limitations with recommendations on possible future research opportunities in the same domain.

#### **6.1. Research Outcome and Contributions**

This research focuses on evaluating the applicability of private blockchains for developing an electronic voting system. For this purpose, this research aimed to design and develop an optimized architecture for an e-voting system and then evaluated the feasibility of achieving all the required properties of an e-voting system. For the scope of the research, the context of Sri Lankan general elections was used and the Hyperledger Fabric blockchain platform was used as the private blockchain platform of the system.

This research provides a detailed architecture of the proposed private blockchain-based e-voting system, along with a reference implementation, which satisfies most of the desired properties of an e-voting system. As described during the evaluation section, this proposed system successfully achieved the properties of Accuracy, Eligibility, Accessibility, Anonymity, Verifiability. Also, the proposed system is capable of scaling to handle the Sri Lankan voter base and the evaluation of the prototype exhibited sufficient performance aspects. Finally, the results observed through the evaluation, confirm that the private blockchains are also suitable for implementing e-voting systems.

The scope of the research was focused on the service layer of an e-voting system which is being replaced with a private blockchain. Within this scope and the limited hardware resources used during the evaluation, the ‘Robustness’ property of the e-voting system cannot be evaluated fully.

As identified during the literature review chapter, there is a lack of researches on the usage of private blockchains for developing e-voting systems. The existing proposals mainly have the limitations of scaling for handling a large number of voters. Thus this research focuses on evaluating the feasibility of using a private blockchain platform for developing a scalable e-voting system. This research proposes an architectural design that is capable of handling a large

number of voters while satisfying all of the desired properties of an e-voting system apart from the 'Robustness'. This design along with the e-voting Smart Contracts is the main research contribution of the research. Also, the prototype e-voting system developed during the research is the main artifact of the research. The statistics gathered on the performance and other non-functional attributes are another research outcome, which will help future researches as a benchmark.

## **6.2. Limitations**

Even though this research proposes a scalable solution for developing e-voting systems using a private blockchain platform while conforming to most of the desired e-voting properties, there are several limitations to the work carried out during the research.

As per the scope considered for this research, only the service layer of an e-voting system was considered. However, to use practically use the proposed design, the other components need to be developed to facilitate the full lifecycle of an election process.

The proposed e-voting still requires the voters to be physically present in a voting center. However, the proposed design and workflow can be used as the basis and it can be enhanced to support fully virtual voting.

The proposed e-voting system architecture is bound to the Hyperledger Fabric blockchain framework. So, to use a different blockchain framework, some analysis is required on adapting the architecture.

Finally, this research only focused on the voting processes used in Sri Lanka. Even though most of the proposed solutions can be adapted, there can be some differences when adapting this to different countries.

## **6.3. Future Work**

The outcome of this research can be used as the basis for further adaptations of private blockchains for e-voting solutions.

One enhancement is to develop an end-to-end e-voting solution using the proposed solution as the basis. Such a system will ensure the proposed system can be used in practical e-voting scenarios.

Evaluating the feasibility of using the proposed solution to enable fully virtual voting and to enable mobile voting is considered as another area that can be explored.

From the evaluation perspective, the proposed system needs to be deployed in a distributed network with multiple worker nodes, to simulate a production-grade deployment and to evaluate the system capabilities.

Another area of interest is to further enhance this proposed solution by making it abstract which can be applied across multiple private blockchain platforms.

## **6.4. Conclusion**

This research establishes that private blockchains can be used to develop e-voting systems while preserving desired properties of e-voting systems while scaling to handling large voter bases. The proposed solution architecture and the prototype system along with the published statistics are expected to serve as a basis for future researches on private blockchains for e-voting systems.

## APPENDICES

### Appendix A: JSON Data Objects in Smart Contracts

```
//ElectionType representing types of elections in Sri Lanka. Need to be added prior to election starts
type ElectionType struct {
    ObjectType string `json:"docType"` //docType is used to distinguish the various types of objects in state database
    Code       string `json:"code"`
    Name       string `json:"name"`
}

//Election representing an election in Sri Lanka. Need to be added prior to election starts
type Election struct {
    ObjectType string `json:"docType"` //docType is used to distinguish the various types of objects in state database
    Code       string `json:"code"`
    Name       string `json:"name"`
    Type       string `json:"type"` //maps to code of ElectionType
    Year       string `json:"year"`
    Status     string `json:"status"` //values: NOT_STARTED, VOTING, PROCESSING, COMPLETED
    VoteStartTime string `json:"voteStartTime"` // in the format of 2021-04-23T18:25:43Z
    VoteEndTime  string `json:"voteEndTime"` // in the format of 2021-04-23T18:25:43Z
}

//PollingDivision representing a polling division in Sri Lanka. Need to be added prior to election starts
type PollingDivision struct {
    ObjectType string `json:"docType"` //docType is used to distinguish the various types of objects in state database
    Code       string `json:"code"`
    Name       string `json:"name"`
}

//PollingSection representing a polling sections in Sri Lanka. Need to be added prior to election starts
type PollingSection struct {
    ObjectType string `json:"docType"` //docType is used to distinguish the various types of objects in state database
    Code       string `json:"code"`
    Name       string `json:"name"`
    PollingDivisionCode string `json:"pollingDivisionCode"`
}

//VotingCenter representing a Leaf Level voting center in Sri Lanka. Need to be added prior to election starts
type VotingCenter struct {
    ObjectType string `json:"docType"` //docType is used to
```

```

distinguish the various types of objects in state database
Code          string `json:"code"`
Name          string `json:"name"`
PollingSectionCode string `json:"pollingSectionCode"`
PollingDivisionCode string `json:"pollingDivisionCode"`
}

//PoliticalParty represeting a political party in the country. Need to
be added prior to election starts
type PoliticalParty struct {
    ObjectType string `json:"docType"` //docType is used to distinguish
the various types of objects in state database
    Code       string `json:"code"`
    Name       string `json:"name"`
}

//PoliticalPartyElective represeting a political party for a electrol
section. Need to be added prior to election starts
//Elective refers to unique segment where candiates will campaign for
type PoliticalPartyElective struct {
    ObjectType      string `json:"docType"` //docType is
used to distinguish the various types of objects in state database
    PartyCode      string `json:"partyCode"` //relates to
code of PoliticalParty
    ElectiveCode   string `json:"electiveCode"` //relates to
code of Elective segment
    ElectionType   string `json:"electionType"` //relates to
election type code
    PollingDivisionCode string `json:"pollingDivisionCode"` //this is
used for Parliamentary and Provincial Council Elections.
    PollingSectionCode string `json:"pollingSectionCode"` //this is
used for local authorities
}

//Candidate representing a election candidate. Need to be added prior
to election starts
type Candidate struct {
    ObjectType string `json:"docType"` //docType is used to
distinguish the various types of objects in state database
    ElectiveCode string `json:"electiveCode"` //relates to code of
Elective segment
    Number       string `json:"number"`
    Name         string `json:"name"`
}

//RegisteredVoter representing a registered voter. Need to be added
prior to election starts
type RegisteredVoter struct {
    ObjectType string `json:"docType"` //docType is used to distinguish
the various types of objects in state database
    VoterId    string `json:"voterId"`
    FullName   string `json:"fullName"`
}

//VotingMetaRecord representing a voting status of a voter.

```

```

type VoterMetaRecord struct {
    ObjectType          string `json:"docType"` //docType is used to
    distinguish the various types of objects in state database
    VoterId             string `json:"voterId"`
    ElectionCode        string `json:"electionCode"`
    Token               string `json:"token"`
    VotingCenterCode    string `json:"votingCenterCode"`
    PollingDivisionCode string `json:"pollingDivisionCode"`
    PollingSectionCode string `json:"pollingSectionCode"`
    VotedTimestamp      string `json:"votedTimestamp"` // in the format
    of 2021-04-23T18:25:43Z
    TempToken           string `json:"tempToken"` //temporary token
    used at voter validity stage
    VotingStatus        string
    `json:"votingStatus"` //DISABLED, ENABLED, VOTE_PENDING, VOTED
}

//CandidateTallyVC holding vote counts against candidate number at the
voting center level
type CandidateTallyVC struct {
    ObjectType          string `json:"docType"` //docType is used to
    distinguish the various types of objects in state database
    ElectionCode        string `json:"electionCode"`
    PartyCode           string `json:"partyCode"`
    CandidateNo         string `json:"candidateNo"`
    VotingCenterCode    string `json:"votingCenterCode"`
    VotesCount          int    `json:"votesCount"`
}

//CandidateTallyVC holding vote counts against candidate number at the
Polling section level
type CandidateTallyPollSec struct {
    ObjectType          string `json:"docType"` //docType is used to
    distinguish the various types of objects in state database
    ElectionCode        string `json:"electionCode"`
    PartyCode           string `json:"partyCode"`
    CandidateNo         string `json:"candidateNo"`
    PollingSectionCode string `json:"pollingSectionCode"`
    PollingDivisionCode string `json:"pollingDivisionCode"`
    VotesCount          int    `json:"votesCount"`
}

//CandidateTallyVC holding vote counts against candidate number at
Polling Division level
type CandidateTallyPollDiv struct {
    ObjectType          string `json:"docType"` //docType is used to
    distinguish the various types of objects in state database
    ElectionCode        string `json:"electionCode"`
    PartyCode           string `json:"partyCode"`
    CandidateNo         string `json:"candidateNo"`
    PollingDivisionCode string `json:"pollingDivisionCode"`
    VotesCount          int    `json:"votesCount"`
}

//PoliticalPartyTallyVC holding vote counts against the political party

```

```

at the voting center level
type PoliticalPartyTallyVC struct {
    ObjectType          string `json:"docType"` //docType is used to
distinguish the various types of objects in state database
    ElectionCode        string `json:"electionCode"`
    PartyCode           string `json:"partyCode"`
    VotingCenterCode    string `json:"votingCenterCode"`
    VotesCount          int    `json:"votesCount"`
}

//PoliticalPartyTallyVC holding vote counts against the political party
at Polling Section level
type PoliticalPartyTallyPollSec struct {
    ObjectType          string `json:"docType"` //docType is used to
distinguish the various types of objects in state database
    ElectionCode        string `json:"electionCode"`
    PartyCode           string `json:"partyCode"`
    PollingSectionCode string `json:"pollingSectionCode"`
    PollingDivisionCode string `json:"pollingDivisionCode"`
    VotesCount          int    `json:"votesCount"`
}

//PoliticalPartyTallyPollDiv holding vote counts against the political
party at Polling Division level
type PoliticalPartyTallyPollDiv struct {
    ObjectType          string `json:"docType"` //docType is used to
distinguish the various types of objects in state database
    ElectionCode        string `json:"electionCode"`
    PartyCode           string `json:"partyCode"`
    PollingDivisionCode string `json:"pollingDivisionCode"`
    VotesCount          int    `json:"votesCount"`
}

//PollingSectionVoteTally holding vote counts at a polling section
type PollingSectionVoteTally struct {
    ObjectType          string `json:"docType"` //docType is used to
distinguish the various types of objects in state database
    ElectionCode        string `json:"electionCode"`
    PollingSectionCode string `json:"pollingSectionCode"`
    PollingDivisionCode string `json:"pollingDivisionCode"`
    VotesCount          int    `json:"votesCount"`
}

//PollingSectionVoteTally holding vote counts at a polling section
type PollingDivisionVoteTally struct {
    ObjectType          string `json:"docType"` //docType is used to
distinguish the various types of objects in state database
    ElectionCode        string `json:"electionCode"`
    PollingDivisionCode string `json:"pollingDivisionCode"`
    VotesCount          int    `json:"votesCount"`
}

```

## Appendix B: Section from Sample Test Data

Vote_Center	VID	Party_Code	Candidate_Code
01_A_0	198000505000	CCC	11
01_A_0	198000505001	GGG	10
01_A_0	198000505002	CCC	10
01_A_0	198000505003	AAA	4
01_A_0	198000505004	AAA	5
01_A_0	198000505005	AAA	20
01_A_0	198000505006	BBB	10
01_A_0	198000505007	AAA	11
01_A_0	198000505008	AAA	4
01_A_0	198000505009	GGG	25
01_A_1	198000505010	AAA	22
01_A_1	198000505011	III	15
01_A_1	198000505012	DDD	6
01_A_1	198000505013	CCC	22
01_A_1	198000505014	BBB	2
01_A_1	198000505015	JJJ	22
01_A_1	198000505016	EEE	21
01_A_1	198000505017	EEE	19
01_A_1	198000505018	JJJ	19
01_A_1	198000505019	BBB	21
01_A_2	198000505020	GGG	18
01_A_2	198000505021	CCC	23
01_A_2	198000505022	AAA	8
01_A_2	198000505023	AAA	16
01_A_2	198000505024	EEE	23
01_A_2	198000505025	GGG	7
01_A_2	198000505026	DDD	15
01_A_2	198000505027	AAA	12
01_A_2	198000505028	CCC	8
01_A_2	198000505029	JJJ	10
01_A_3	198000505030	III	24
01_A_3	198000505031	CCC	4
01_A_3	198000505032	EEE	12
01_A_3	198000505033	EEE	10
01_A_3	198000505034	BBB	3
01_A_3	198000505035	III	13
01_A_3	198000505036	CCC	20
01_A_3	198000505037	GGG	16
01_A_3	198000505038	CCC	3
01_A_3	198000505039	EEE	11
01_A_4	198000505040	CCC	10
01_A_4	198000505041	BBB	17
01_A_4	198000505042	BBB	19



**Appendix C: Candidate Level Vote Data vs E-Voting Recorded Data for  
Scenario Three**

<i>Election Party Code</i>	<i>Candidate Number</i>	<i>Votes in Data</i>	<i>Votes recorded by e-voting</i>
AAA	1	23	23
	2	18	18
	3	24	24
	4	12	12
	5	19	19
	6	24	24
	7	21	21
	8	11	11
	9	20	20
	10	28	28
	11	18	18
	12	14	14
	13	7	7
	14	17	17
	15	16	16
	16	29	29
	17	14	14
	18	18	18
	19	16	16
	20	15	15
	21	15	15
	22	18	18
	23	14	14
	24	17	17
	25	18	18
AAA Total		446	446
BBB	1	19	19
	2	24	24
	3	14	14
	4	19	19
	5	17	17
	6	26	26

	7	24	24
	8	17	17
	9	20	20
	10	18	18
	11	21	21
	12	19	19
	13	22	22
	14	26	26
	15	10	10
	16	13	13
	17	18	18
	18	13	13
	19	24	24
	20	19	19
	21	27	27
	22	17	17
	23	18	18
	24	19	19
	25	16	16
BBB Total		480	480
CCC	1	14	14
	2	24	24
	3	18	18
	4	17	17
	5	15	15
	6	19	19
	7	19	19
	8	21	21
	9	17	17
	10	24	24
	11	17	17
	12	21	21
	13	13	13
	14	18	18
	15	18	18
	16	23	23

	17	19	19
	18	15	15
	19	23	23
	20	13	13
	21	20	20
	22	18	18
	23	23	23
	24	24	24
	25	32	32
CCC Total		485	485
DDD	1	3	3
	2	9	9
	3	4	4
	4	3	3
	5	9	9
	6	7	7
	7	3	3
	8	4	4
	9	6	6
	10	9	9
	11	5	5
	12	6	6
	13	5	5
	14	7	7
	15	4	4
	16	9	9
	17	5	5
	18	7	7
	19	5	5
	20	7	7
	21	3	3
	22	3	3
	23	5	5
	24	9	9
	25	8	8
DDD Total		145	145

EEE	1	3	3
	2	3	3
	3	5	5
	4	4	4
	5	4	4
	6	4	4
	7	6	6
	8	6	6
	9	8	8
	10	7	7
	11	7	7
	12	4	4
	13	6	6
	14	4	4
	15	9	9
	16	8	8
	17	3	3
	18	5	5
	19	5	5
	20	5	5
	21	4	4
	22	6	6
	23	8	8
	24	8	8
	25	8	8
EEE Total		140	140
FFF	1	6	6
	2	6	6
	3	7	7
	4	7	7
	5	4	4
	6	6	6
	7	5	5
	8	6	6
	9	8	8
	10	6	6

	11	7	7
	12	3	3
	13	6	6
	14	3	3
	15	8	8
	16	3	3
	17	4	4
	18	3	3
	19	6	6
	20	4	4
	21	8	8
	22	1	1
	23	5	5
	24	8	8
	25	5	5
FFF Total		135	135
GGG	1	4	4
	2	11	11
	3	5	5
	4	10	10
	5	2	2
	6	6	6
	7	11	11
	8	3	3
	9	7	7
	10	8	8
	11	3	3
	12	2	2
	13	4	4
	14	4	4
	15	7	7
	16	9	9
	17	2	2
	18	6	6
	19	2	2
	20	4	4

	21	6	6
	22	5	5
	23	7	7
	24	5	5
	25	4	4
<b>GGG Total</b>		<b>137</b>	<b>137</b>
<b>HHH</b>	1	4	4
	2	7	7
	3	10	10
	4	7	7
	5	7	7
	6	7	7
	7	6	6
	8	5	5
	9	4	4
	10	9	9
	11	5	5
	12	8	8
	13	10	10
	14	7	7
	15	6	6
	16	9	9
	17	4	4
	18	4	4
	19	5	5
	20	9	9
	21	5	5
	22	5	5
	23	11	11
	24	8	8
	25	6	6
<b>HHH Total</b>		<b>168</b>	<b>168</b>
<b>III</b>	1	9	9
	2	4	4
	3	7	7
	4	8	8

	5	8	8
	6	6	6
	7	2	2
	8	5	5
	9	6	6
	10	8	8
	11	6	6
	12	3	3
	13	4	4
	14	2	2
	15	3	3
	16	8	8
	17	7	7
	18	2	2
	19	7	7
	20	5	5
	21	6	6
	22	6	6
	23	9	9
	24	5	5
	25	9	9
III Total		145	145
JJJ	1	3	3
	2	5	5
	3	6	6
	4	4	4
	5	6	6
	6	4	4
	7	5	5
	8	2	2
	9	7	7
	10	2	2
	11	12	12
	12	8	8
	13	5	5
	14	7	7

	15	2	2
	16	1	1
	17	2	2
	18	5	5
	19	7	7
	20	4	4
	21	3	3
	22	2	2
	23	7	7
	24	3	3
	25	7	7
JJJ Total		119	119
<b>Grand Total</b>		<b>2400</b>	<b>2400</b>



## Appendix D: Response Time Variation of All Requests for Scenario One

The following two figures illustrate the response charts generated by the JMeter at the 'Validation' and 'Vote' stages with three concurrent voters.

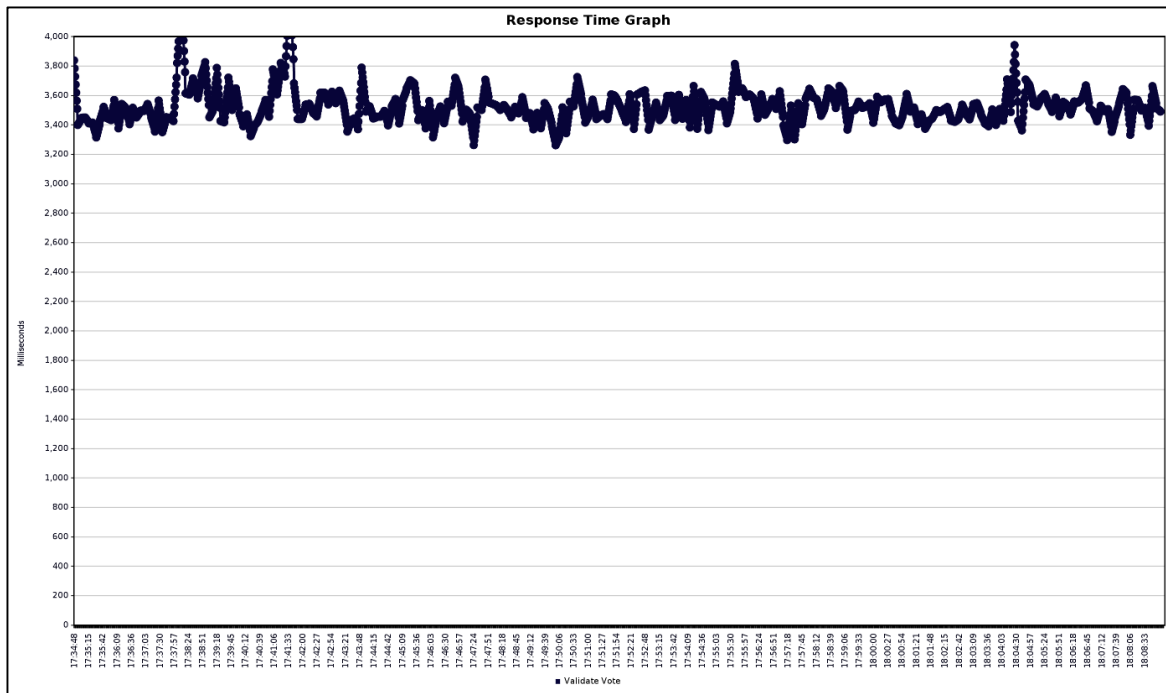


Figure D.1: Response Times for 'Validation' Step with Three Concurrent Voters

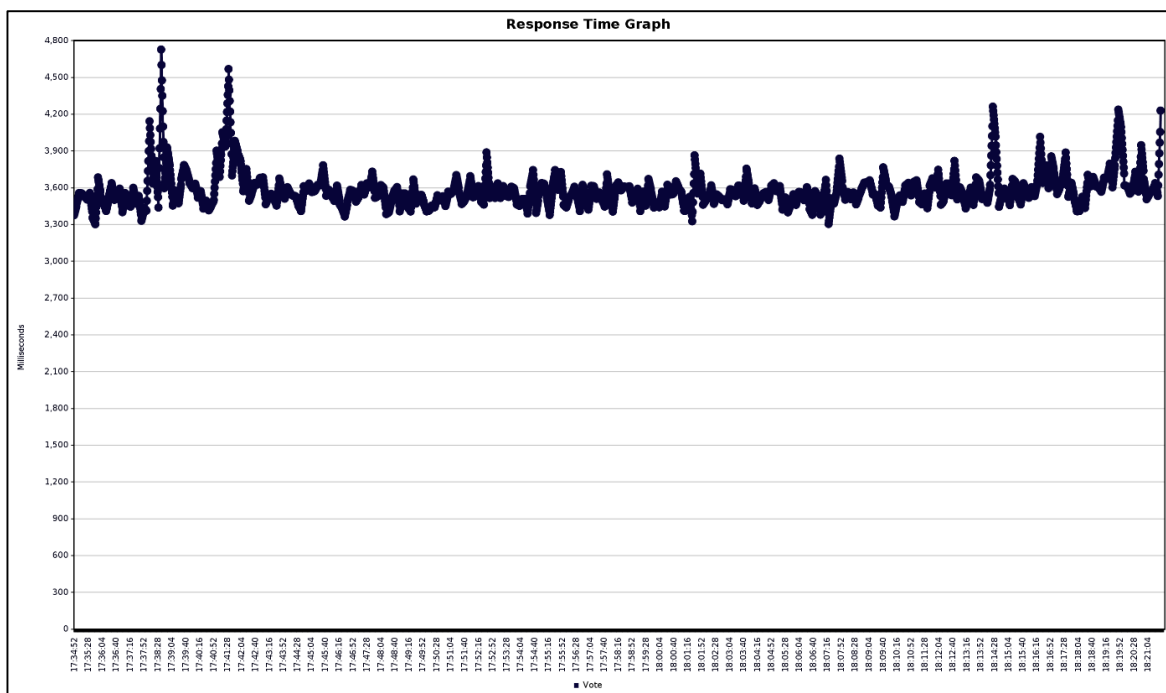


Figure D.2: Response Times for 'Voting' Step with Three Concurrent Voters

The following two figures illustrate the response charts generated by the JMeter at the ‘Validation’ and ‘Vote’ stages with three concurrent voters.

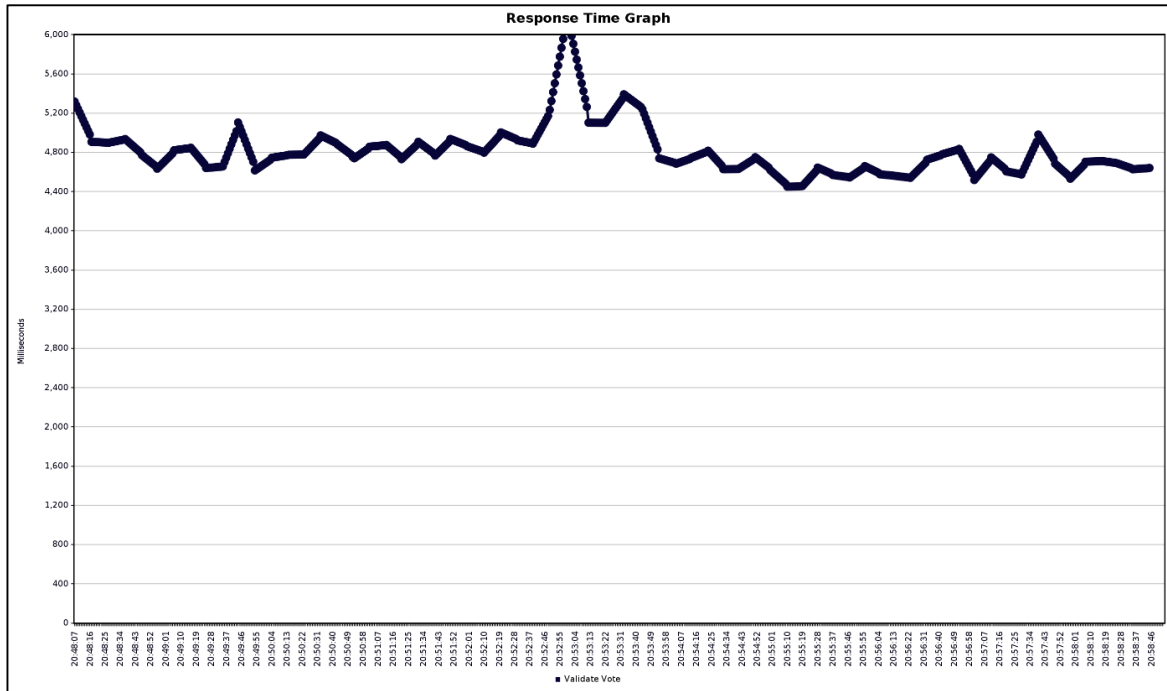


Figure D.3: Response Times for ‘Validation’ Step with Six Concurrent Voters

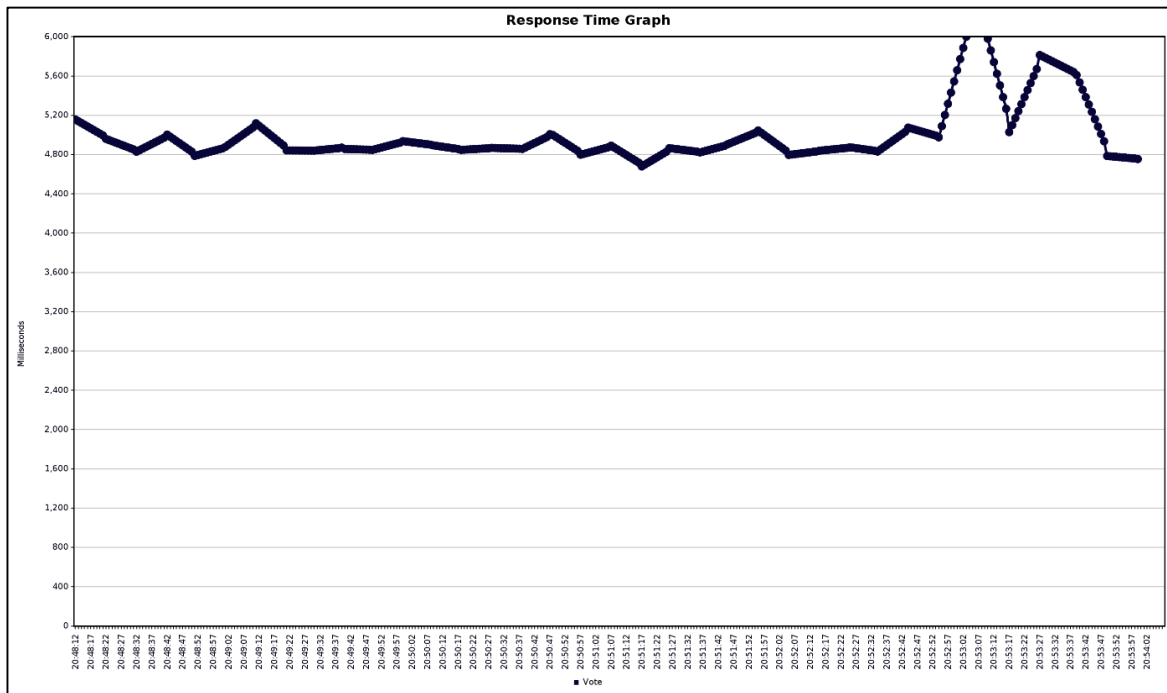


Figure D.4: Response Times for ‘Voting’ Step with Six Concurrent Voters

## REFERENCES

- Adiputra, C., Hjort, R., Sato, H., 2018. A Proposal of Blockchain-Based Electronic Voting System. <https://doi.org/10.1109/WorldS4.2018.8611593>
- Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S.W., Yellick, J., 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. Proc. Thirteen. EuroSys Conf. 1–15. <https://doi.org/10.1145/3190508.3190538>
- Bartolucci, S., Bernat, P., Joseph, D., 2018. SHARVOT: secret SHARe-based VOTing on the blockchain. Proc. 1st Int. Workshop Emerg. Trends Softw. Eng. Blockchain 30–34. <https://doi.org/10.1145/3194113.3194118>
- Ben Ayed, A., 2017. A CONCEPTUAL SECURE BLOCKCHAIN-BASED ELECTRONIC VOTING SYSTEM. <https://doi.org/10.5121/ijnsa.2017.9301>
- Bokslag, W., de Vries, M., 2016. Evaluating e-voting: theory and practice. ArXiv160202509 Cs.
- Bulut, R., Kantarcı, A., Keskin, S., Bahtiyar, Ş., 2019. Blockchain-Based Electronic Voting System for Elections in Turkey, in: 2019 4th International Conference on Computer Science and Engineering (UBMK). pp. 183–188. <https://doi.org/10.1109/UBMK.2019.8907102>
- Centre for Monitoring Election Violence, 2019. 2019 Sri Lankan Presidential Election-Observation Report.
- Chen, W., Xu, Z., Shi, S., Zhao, Y., Zhao, J., 2018. A Survey of Blockchain Applications in Different Domains, in: Proceedings of the 2018 International Conference on Blockchain Technology and Application - ICBTA 2018. Presented at the the 2018 International Conference, ACM Press, Xi'an, China, pp. 17–21. <https://doi.org/10.1145/3301403.3301407>
- Daramola, O., Thebus, D., 2020. Architecture-Centric Evaluation of Blockchain-Based Smart Contract E-Voting for National Elections. Informatics 7, 16. <https://doi.org/10.3390/informatics7020016>
- e-Estonia, n.d. i-Voting [WWW Document]. E-Est. URL <https://e-estonia.com/solutions/e-governance/i-voting/> (accessed 9.1.20).
- Election Commission of Sri Lanka, 2021. Comparative note on the number of registered voters registered according to the electoral district 2010 - 2019.
- Franke, D., Darmstadt, T., n.d. Security Analysis of the Geneva e-voting system 15.

- Hao, F., Ryan, P.Y.A., Zieliński, P., 2010. Anonymous voting by two-round public discussion. *IET Inf. Secur.* 4, 62. <https://doi.org/10.1049/iet-ifs.2008.0127>
- Hardwick, F.S., Gioulis, A., Akram, R.N., Markantonakis, K., 2018. E-Voting with Blockchain: An E-Voting Protocol with Decentralisation and Voter Privacy. *ArXiv180510258 Cs*.
- Hjálmarsson, F.P., Hreiðarsson, G.K., n.d. Blockchain-Based E-Voting System 10.
- Ibrahim, S., Kamat, M., Salleh, M., Aziz, S.R.A., 2003. Secure E-voting with blind signature, in: 4th National Conference of Telecommunication Technology, 2003. NCTT 2003 Proceedings. Presented at the 4th National Conference on Telecommunication Technology. Proceedings, IEEE, Shah Alam, Malaysia, pp. 193–197. <https://doi.org/10.1109/NCTT.2003.1188334>
- Julien Polge, Jérémy Robert, Yves Le Traon, 2020. Permissioned blockchain frameworks in the industry: A comparison.
- Khan, K.M., Arshad, J., Khan, M.M., 2020. Investigating performance constraints for blockchain based secure e-voting system. *Future Gener. Comput. Syst.* 105, 13–26. <https://doi.org/10.1016/j.future.2019.11.005>
- Khoury, D., Kfoury, E.F., Kassem, A., Harb, H., 2018. Decentralized Voting Platform Based on Ethereum Blockchain, in: 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET). Presented at the 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), IEEE, Beirut, pp. 1–6. <https://doi.org/10.1109/IMCET.2018.8603050>
- Kirillov, D., Korkhov, V., Petrunin, V., Makarov, M., Khamitov, I.M., Dostov, V., 2019. Implementation of an E-Voting Scheme Using Hyperledger Fabric Permissioned Blockchain, in: Misra, S., Gervasi, O., Murgante, B., Stankova, E., Korkhov, V., Torre, C., Rocha, A.M.A.C., Taniar, D., Apduhan, B.O., Tarantino, E. (Eds.), *Computational Science and Its Applications – ICCSA 2019, Lecture Notes in Computer Science*. Springer International Publishing, Cham, pp. 509–521. [https://doi.org/10.1007/978-3-030-24296-1\\_40](https://doi.org/10.1007/978-3-030-24296-1_40)
- Krimmer, R., n.d. E-Vote-ID 2018, *TUT Press Proceedings* 373.
- Kshetri, N., Voas, J., 2018. Blockchain-Enabled E-Voting. *IEEE Softw.* 35, 95–99. <https://doi.org/10.1109/MS.2018.2801546>
- Li, J., Liu, Z., Chen, L., Chen, P., Jigang, W., 2017. Blockchain-Based Security Architecture for Distributed Cloud Storage. 2017 IEEE Int. Symp. Parallel Distrib. Process. Appl. 2017 IEEE Int. Conf. Ubiquitous Comput. Commun. ISPAIUCC 408–411.
- Li, X.S., Lee, H. ran, Lee, M., Choi, J., 2015. A Study of Vulnerabilities in E-Voting System. Presented at the CIA 2015, pp. 136–139. <https://doi.org/10.14257/astl.2015.95.25>

Lukka, K., 2003. The Constructive Research Approach, in: Case Study Research in Logistics. pp. 83–101.

Luo, Y., Chen, Y., Chen, Q., Liang, Q., 2018. A New Election Algorithm for DPos Consensus Mechanism in Blockchain. <https://doi.org/10.1109/ICDH.2018.00029>

Mukherjee, P.P., Boshra, A.A., Ashraf, M.M., Biswas, M., 2020. A Hyper-ledger Fabric Framework as a Service for Improved Quality E-voting System, in: 2020 IEEE Region 10 Symposium (TENSYMP). Presented at the 2020 IEEE Region 10 Symposium (TENSYMP), IEEE, Dhaka, Bangladesh, pp. 394–397.  
<https://doi.org/10.1109/TENSYMP50017.2020.9230820>

Novotný, M., 2009. Design and Analysis of a Practical E-Voting Protocol, in: Matyáš, V., Fischer-Hübner, S., Cvrček, D., Švenda, P. (Eds.), The Future of Identity in the Information Society, IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 170–183. [https://doi.org/10.1007/978-3-642-03315-5\\_13](https://doi.org/10.1007/978-3-642-03315-5_13)

Okediran O. O, Omidiora E. O., Olabiyisi S. O., Ganiyu R. A., 2011. A Survey of Remote Internet Voting Vulnerabilities 5.

Ongaro, D., Ousterhout, J., n.d. In Search of an Understandable Consensus Algorithm 18.

Puthal, D., Malik, N., Mohanty, S.P., Kougianos, E., Das, G., 2018. Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems. IEEE Consum. Electron. Mag. 7, 6–14. <https://doi.org/10.1109/MCE.2018.2816299>

Sirimanna, I., Jinasena, K., 2019. Blockchain-based Secure, Reliable, and Distributed Voting System for Decision Making in Government Policies and Projects.

Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., Halderman, J.A., 2014. Security Analysis of the Estonian Internet Voting System, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14. Presented at the the 2014 ACM SIGSAC Conference, ACM Press, Scottsdale, Arizona, USA, pp. 703–715. <https://doi.org/10.1145/2660267.2660315>

Srivastava, G., Dhar Dwivedi, A., Singh, R., 2018. Crypto-democracy: A Decentralized Voting Scheme using Blockchain Technology:, in: Proceedings of the 15th International Joint Conference on E-Business and Telecommunications. Presented at the International Conference on Security and Cryptography, SCITEPRESS - Science and Technology Publications, Porto, Portugal, pp. 674–679. <https://doi.org/10.5220/0006881906740679>

Stenbro, M., 2010. A Survey of Modern Electronic Voting Technologies 162.

Taş, R., Tanrıöver, Ö.Ö., 2020. A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting. Symmetry 12, 1328. <https://doi.org/10.3390/sym12081328>

The Idea of Smart Contracts | Satoshi Nakamoto Institute [WWW Document], n.d. URL <https://nakamotoinstitute.org/the-idea-of-smart-contracts/> (accessed 9.1.20).

The Swiss authorities online, n.d. E-Voting - www.ch.ch [WWW Document]. URL <https://www.ch.ch/en/demokratie/voting-online/> (accessed 9.1.20).

Weber, S., 2008. Coercion-Resistant Cryptographic Voting: Implementing Free and Secret Electronic Elections.

Weiler, T., 2016. Constitutional Parameters for E-Voting in Germany.pdf.

Working group: Risks of online voting outweigh its benefits - Ministry of Justice [WWW Document], n.d. . Oikeusministeriö. URL <https://oikeusministerio.fi/en/-/tyoryhmanettiaanestyksen-riskit-suuremmat-kuin-hyodyt> (accessed 3.7.21).

Yao, Y., Houston, A., 2002. Electronic Voting System Characteristics and Voter Participation Intention 7.

Yavuz, E., Koc, A.K., Cabuk, U.C., Dalkilic, G., 2018. Towards secure e-voting using ethereum blockchain, in: 2018 6th International Symposium on Digital Forensic and Security (ISDFS). Presented at the 2018 6th International Symposium on Digital Forensic and Security (ISDFS), IEEE, Antalya, pp. 1–7. <https://doi.org/10.1109/ISDFS.2018.8355340>

Zhang, W., Yuan, Y., Hu, Y., Huang, Shaohua, Cao, S., Chopra, A., Huang, Sheng, 2018. A Privacy-Preserving Voting Protocol on Blockchain, in: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). Presented at the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), IEEE, San Francisco, CA, pp. 401–408. <https://doi.org/10.1109/CLOUD.2018.00057>

Zhao, Z., Chan, T.-H., 2016. How to Vote Privately Using Bitcoin. [https://doi.org/10.1007/978-3-319-29814-6\\_8](https://doi.org/10.1007/978-3-319-29814-6_8)

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., Wang, H., 2017. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. <https://doi.org/10.1109/BigDataCongress.2017.85>