UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Design and Scalability Performance Evaluation of Permissioned Blockchain Architecture for Online Voting System

Wong Chee Oon[1]* & Siti Hajar Othman[2]
Faculty of Computing,
Universiti Teknologi Malaysia
81310, UTM Johor Bahru, Johor, Malaysia
Email: wongcheeoon@graduate.utm.my[1]; hajar@utm.my[2]

Submitted: 22/2/2025. Revised edition: 20/4/2025. Accepted: 4/5/2025. Published online: 27/5/2025
DOI: https://doi.org/10.11113/ijic.v15n1.527

***Abstract*—This research addresses critical challenges in online voting systems by integrating blockchain technology, specifically leveraging Hyperledger Fabric. The objectives include designing a permissioned blockchain architecture tailored for online voting systems, developing and implementing smart contracts to manage the entire voting lifecycle, including voter registration, vote casting, and result viewing within the permissioned blockchain environment and assessing the scalability and performance of the proposed voting system architecture using Hyperledger Caliper. The study begins with a comprehensive review of literature and case studies to identify security gaps within existing online voting systems, focusing on both on-chain and off-chain aspects. Subsequently, an online voting system architecture with a single channel blockchain network is designed, using Hyperledger Fabric to enhance the scalability of online voting processes. Then, smart contracts are developed to implement the logic of the voting application. To assess the effectiveness of the designed architecture, extensive testing and evaluation are conducted. Key scalability performance metrics are measured using Hyperledger Caliper on critical operational functions. The Hyperledger Fabric Network was incrementally scaled from a single-peer network to a five-peer network configuration to assess scalability. The performance assessment suggests that a five-peer network is optimal for the proposed online voting system. The findings pave the way for future advancements in blockchain-based solutions, offering a secure, scalable and transparent framework for online voting systems.**

***Keywords*—Hyperledger Fabric, Hyperledger Caliper, permissioned blockchain, online voting system, scalability**

I. INTRODUCTION

Voting is an essential part of social decision-making in a democratic system. However, not much has been done to enhance our voting process, despite the importance and value of this activity. Although paper ballots are still the most popular technique, they are quite complicated, a source of many inconveniences, and stepping back to the advancements of the modern world.

Two types of automated voting systems have emerged: e-voting, which uses voting machines, and i-voting, which allows voting via internet browsers. Estonia was the first nation to set up a nationwide online voting system. Through the internet, they made it possible for people to cast their votes from anywhere in the world (Pranith, 2019). Shortly after, Switzerland and Norway introduced electronic voting for regional and local elections (Ben Ayed, 2017).

There are a few limitations to digital voting (R. Krishnamurthy *et al*., 2020). One notable concern is the secrecy surrounding key portions of the code, as witnessed in Estonian and Norwegian electronic voting systems. Due to concerns about confidentiality, the ballot format is restricted in Estonia. The I-Voting system's centralized nature makes it vulnerable to DDOS attacks, which could render elections inaccessible to voters (Ben Ayed, 2017). Voters may have questions about the process's fairness and anonymity (Zhang *et al*., 2018). Furthermore, police and security services can access and study network traffic, creating the potential of data manipulation. Despite improved security, system attacks are still possible (Ben Ayed, 2017). As a result, additional modifications are required to ensure reliability and address these concerns (R. Krishnamurthy *et al*., 2020).

There are several challenges for electronic voting systems, including authentication, privacy, data integrity, transparency, and verifiability. Blockchain technology, which was developed more than a decade ago, provides a unique answer to a variety of problems. Blockchain is a decentralized network in which node

members trade data but each user keeps an identical data replication (Zhang *et al*., 2018). The blockchain satisfies the security and transparency requirements that are currently absent from the current electronic voting methods due to its technological qualities (Fusco *et al*., 2018).

There is an increasing trend and interest in combining blockchain with voting systems. The unique qualities of blockchain technology, such as decentralization and immutability, are invaluable in ensuring that voting conducted on the system can follow the same rules governing more traditional forms of elections (Pawlak & Poniszewska-Marańda, 2021). Despite the benefits of blockchain technology, there are certain drawbacks, including scalability concerns, poor transaction rates, expense, and more. Different blockchain platforms have unique characteristics suitable for various applications, including voting. This proves that blockchain systems can be the missing puzzle to solve most of the cons of online voting systems while maintaining maximum security.

Hence, it is worthwhile considering the implementation of an online voting system since blockchain technology can make it feasible and realistic (Perry., n.d). With the existence of online voting systems, we can ensure enhanced security and transparency in the electoral process. Blockchain's decentralized nature reduces the risk of tampering and fraud, ensuring that each vote is securely recorded and accurately counted. Additionally, the immutability of blockchain records provides a reliable audit trail, increasing voter trust in the integrity of the election. The accessibility of online voting can also increase voter participation by making it more convenient for people to vote from anywhere, anytime. By leveraging blockchain technology, we can address many of the challenges associated with traditional voting systems and move towards a more efficient and trustworthy democratic process.

II. LITERATURE REVIEW

Several topics need to be covered to gain a clear understanding of how to execute the project and achieve its goals.

A. Blockchain Technology

Blockchain was popularized by the success of Bitcoin (Nakamoto, 2008) and it can be used to conduct trustworthy and secure transactions across an untrusted network without the need for a centralized third party. It is a decentralized data managing system where the data is sequentially stored in an encrypted chain of blocks and distributed via a peer-to-peer (P2P) network (Alam *et al*., 2018).

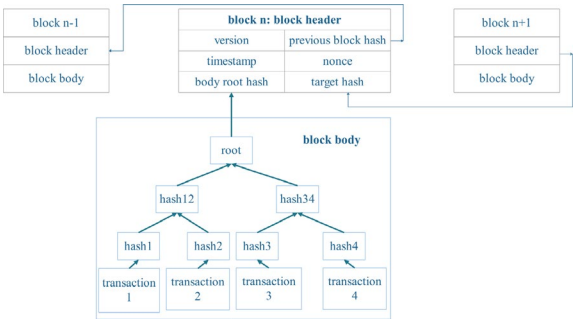

Fig. 1. Blockchain Block Structure (Shi *et al*., 2020)

Blockchain is a decentralized, trustworthy distributed ledger on a peer-to-peer network made up of chronologically ordered blocks. Every block contains a hash of the previous block, resulting in a chain. The genesis block is the first block on the blockchain, and the block that comes before it is referred to as its parent block (Shi *et al*., 2020). A block is made up of two parts: the block header and the block body, as seen in Fig. 1.

The block header includes the following information such as the block version, previous block hash, timestamp, nonce, body root hash and target hash. While the block body in a blockchain system consists of validated transactions within a specific time. The Merkle tree is used to store all valid transactions, with every leaf node representing a transaction and every non-leaf node representing the hash value of its two concatenated child nodes. This structure is efficient for verifying transaction existence and integrity, as node can confirm validation by the hash value of the branches. Any modification to a transaction generates a new hash value, resulting in a falsified root hash. The blocks are chained together using cryptographic hash functions, ensuring immutability and security.

B. Hyperledger Fabric

Hyperledger Fabric, designed for private blockchain networks, operates without miners and maintains distinct entities and responsibilities from Bitcoin and Ethereum. It categorizes entities into application clients, peers, and orderers. Clients independently generate transactions and communicate with other Fabric network members, while authenticated peers form the network's foundation, and orderers manage transaction ordering.

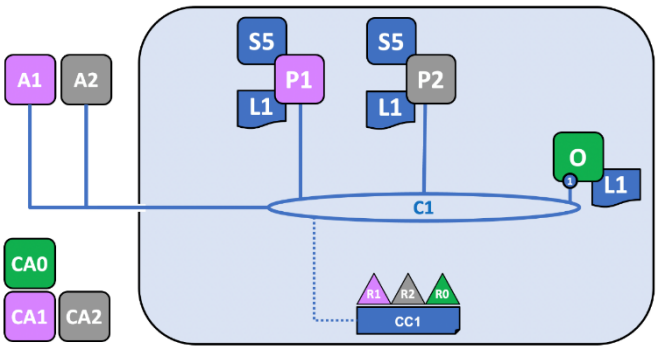

Fig. 2. Sample Hyperledger Fabric Network (*How Fabric networks are structured*¶, n.d)

On the sample Hyperledger Fabric network provided in Fig. 2 above, three organizations, R1, R2, and R0, have agreed to establish a network with a configuration called CC1, outlining their roles and policies. R1 and R2 will join P1 and P2 to the channel, while R0 owns O, the ordering service. Nodes will contain a copy of the channel's ledger, but the ordering service does not have a state database. R1 and R2 will interact with the channel through their own applications A1 and A2, and all three organizations have a Certificate Authority.

The transaction flow in Fabric begins with clients sending proposed transactions to endorsing peers, chosen based on an

endorsement policy specifying the minimum number required for validity. Endorsers, often from various consortium organizations, simulate transactions without altering the blockchain. They validate the read-write set and return it to the client. If approvals are insufficient, the transaction is halted; otherwise, the client submits the approved transaction and sets to the ordering service. Orderers receive endorsed transactions and arrange them into blocks using a consensus protocol. These blocks are distributed to all peers, which validate the transactions against the current blockchain state and update it accordingly.

Fabric lacks lightweight nodes and allows orderers to store blocks. In the transaction flow, peers endorse each other in the first phase, orderers arrange transactions in the second, and all peers validate and update the blockchain in the third. The need for a proposer is reduced due to Fabric's smaller scale, although scalability efforts continue. Application clients serve as query issuers and responders, while peers maintain the system state. In specific use cases, the number of Hyperledger Fabric channels can be increased to enhance network scalability and improve transaction throughput, thereby optimizing the overall performance and efficiency of the system. Overall, Fabric's structure and transaction flow facilitate efficient private blockchain operation tailored to specific organizational needs (Tabatabaei *et al*., 2023).

C. Online Voting System

In democratic countries, elections are an essential method of selecting representative leadership and offer a vital forum for the public to voice their opinions on issues of politics (Jennings & Wlezien, 2016). Traditional voting methods, especially those that use ballot paper, frequently struggle with trust concerns despite their crucial function (Simons & Jones, 2012).

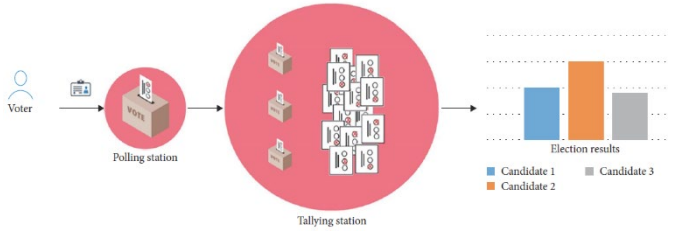

Fig. 3. Traditional Paper Ballot Voting System (El Kafhali & Sudhakar, 2024)

It costs a lot of money and effort to set up and maintain this traditional voting system. Election manipulation, fraud, and rigging are just a few of the factors that have highlighted the need for a stronger, more open, and safer system. Electronic voting, or "e-Voting," has become quite popular worldwide as a solution to these issues since it presents itself as a desirable substitute for conventional voting techniques (Kohno *et al*., 2004). The introduction of electronic voting (e-voting) aims to eliminate electoral fraud, streamline the voting process, save costs, and improve accessibility by mitigating the hazards associated with ballot paper voting (Kumar & Begum, 2012). However, despite its numerous advantages, online voting in its early stages was not immune to its own set of vulnerabilities.

There are generally two types of voting mechanisms: e-voting and i-voting, as shown in Fig. 4. below.

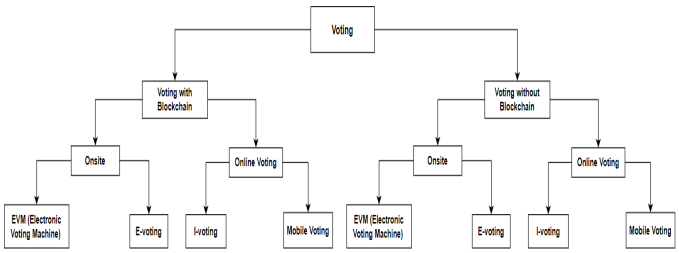

Fig. 4. Classification of Voting (Alvi *et al*., 2020)

E-voting involves an onsite machine that allows voters to cast their votes using the machine. I-voting, on the other hand, is software that enables voters to vote from anywhere using devices such as laptops, computers, or mobile phones with network access. The primary difference between e-voting and i-voting lies in their implementation and accessibility.

In summary, e-voting requires physical presence at a polling station with electronic voting machines, while i-voting allows remote voting via internet-connected devices, enhancing accessibility but posing different security challenges.

D. Hyperledger Caliper

Hyperledger Caliper is a benchmarking tool designed to measure the performance of various blockchain solutions. It supports several performance metrics that help evaluate the efficiency and effectiveness of blockchain networks. Some key performance metrics were:

1) Transaction Throughput (TPS)

$$TPS = \frac{Total\ Comitted\ Transactions}{Total\ Time\ in\ Seconds} \quad (1)$$

This is to measure the number of transactions processed per second by the blockchain network. Higher throughput indicates better performance and the ability to handle a larger volume of transactions.

2) Transaction Latency (TL)

$$TL = Confirmation\ Time - Submit\ Time \quad (2)$$

This is to measure the time taken for a transaction to be processed and included in a block. Lower latency indicates faster transaction processing and improved user experience.

III. IMPLEMENTATION DESIGN

The flow of the permissioned blockchain architecture for the online voting system was proposed as shown in Fig. 5.

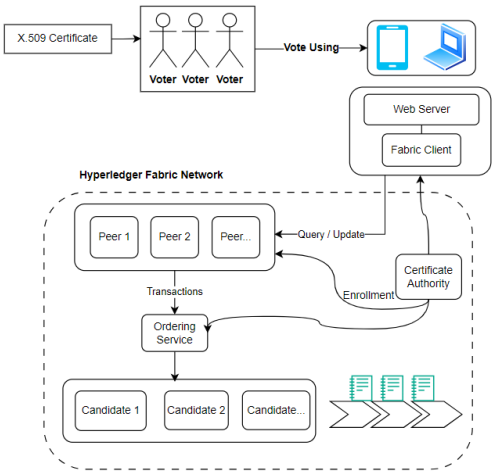

Fig. 5. Overview of Permissioned Blockchain Voting System Architecture

Voter authentication and vote casting in the system are managed efficiently. Voters are issued X.509 certificates by the Certificate Authority (CA), serving as their digital identities. These certificates authenticate voters, ensuring only authorized individuals can participate. Voters access the system through a user-friendly web interface available on devices like tablets and laptops, with the web server facilitating interactions between the users and the system.

The process of logging votes involves several components. The Fabric Client bridges external interactions and the blockchain by checking the Votes Database for new votes to process. It sends transaction proposals to endorsing peer nodes for validation. Peers play a critical role in verifying the authenticity and integrity of transactions before committing them to the ledger. These transactions represent votes for candidates, which are securely recorded on the blockchain.

A. Proposed Hyperledger Fabric Networks

Five Hyperledger Fabric networks, each with a different number of peers, will be set up for experimental evaluation to identify the optimal peer configuration for a single-channel network.

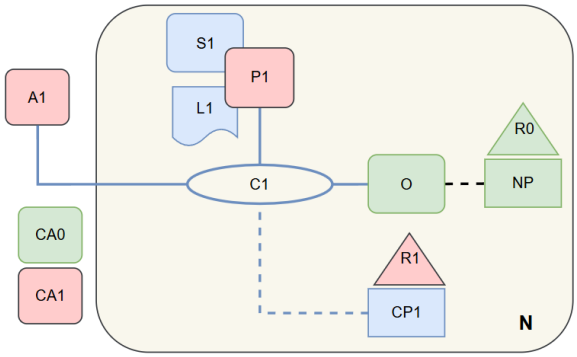

Fig. 6. Hyperledger Fabric Network with One Peer

The number of peers will be added the channel C1 incrementally and Hyperledger Fabric network will be evaluated for its scalability.

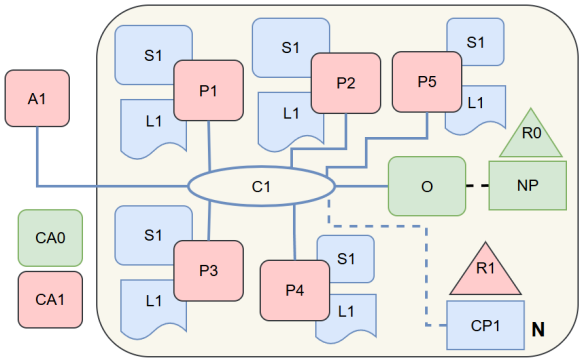

Fig. 7. Hyperledger Fabric Network with Five Peers

The Hyperledger Fabric network N operates under the governance of network policy NP and is supported by an ordering service O, which is owned by organization R0. Channel C1, established by consortium R1, functions as a communication and transaction layer for the network. Governed by channel policy CP1, channel C1 is managed by the ordering service O and maintained by five peers, P1, P2, P3, P4, and P5 depicted from the figure accordingly, which collaboratively uphold the ledger L1 associated with the channel. All five peers execute smart contracts through chaincode S1, ensuring maximum availability, redundancy, fault tolerance, and efficient transaction processing while facilitating the core business logic for the network.

Client applications A1, owned by organization R1, have been granted permission to transact on channel C1. These applications interface with the network to invoke transactions and query data as needed. To enable secure interactions, certificate authority CA1 serves organization R1, issuing digital identities and certificates for its members.

B. Experimental Setup

The experimental setup consists of both hardware and software components to develop the online voting system and evaluate its performance using Hyperledger Caliper.

For the hardware setup, the system is equipped with an AMD Ryzen 9950x processor, 64GB RAM, and 1TB disk storage. The operating system used is Ubuntu Linux, running within a Docker container to ensure a consistent and isolated testing environment.

On the software side, the setup follows the default configuration while installing the latest software dependencies to maintain a standardized implementation. Hyperledger Fabric serves as the blockchain framework, while Hyperledger Caliper is used for benchmarking and performance analysis. The entire system is deployed in a containerized environment using Docker and Docker Compose, enabling efficient resource management and scalability testing.

C. Implementation of Work

To begin setting up the Hyperledger Fabric network, the Hyperledger Fabric repository is cloned from GitHub, and the different number of peers network were constructed. Upon executing the command, the following components such as containers and volumes are created. The channel "mychannel" has been successfully created and the containers and volumes for each organization are up and running. However, it is important to note that after the script is successfully run, only one peer is set up in the network by default. This peer will be the primary node interacting with the network, and additional peers can be added in later steps to scale the network.

Then, chaincode for the voting system was developed and deployed using **Go**, which is one of the supported programming languages in Hyperledger Fabric. The chaincode encapsulates the business logic for the voting system, ensuring that all the operations are securely processed on the blockchain. Several key functions were developed as part of the chaincode to manage the election lifecycle, voter interactions, and calculation results. These functions include InitElection, RegisterVoter, LoginVoter, CastVote, GetResults, GetVoterStatus, GetElectionInfo, QueryVoters.

Furthermore, the backend for the voting system is developed using Go and serves as the intermediary between the client application and the Hyperledger Fabric blockchain network. It is responsible for managing interactions with the blockchain, handling user requests, and ensuring the system operates securely and efficiently.

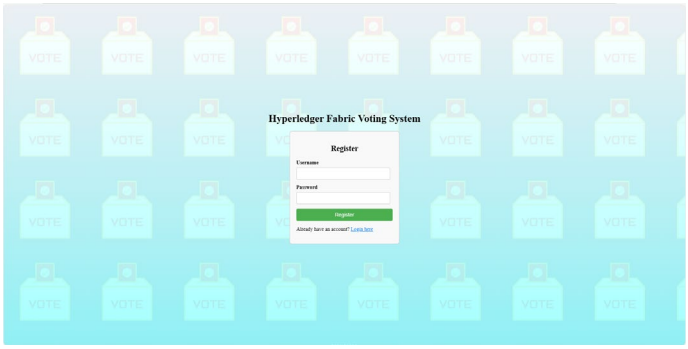

Fig. 8. Frontend Voting Application

A frontend interface is essential to enable user interaction with the voting system, serving as the bridge between users and the backend API. The frontend application for the voting system is implemented using Angular, a popular web development framework. The application is designed to provide a user-friendly interface for interacting with the blockchain network and executing voting operations.

D. Performance Evaluation

Performance evaluation is essential to understand how well the online voting system architecture performs under high-load conditions. This section outlines the testing environment, tools, metrics, scalability testing, and how data will be collected and analyzed, with Hyperledger Caliper as the primary tool for performance measurement.

For the performance evaluation of the online voting system's blockchain architecture, the following metrics will be used to assess the system's behavior under various loads:

a) Name

The name of the test or transaction being measured such as voter registration and vote casting.

b) Success

The number of successful transactions executed during the test. This metric indicates how many transactions were processed successfully without errors.

c) Failure

The number of failed transactions during the test. This metric is crucial for identifying potential issues in transaction processing or chaincode execution.

d) Send Rate (TPS)

The transactions per second (TPS) sent by the client or frontend. This shows the rate at which transactions are being requested by the system, reflecting the load being applied during the test.

e) Max Latency (s)

The maximum latency observed for any transaction. This represents the highest time taken for a single transaction to be processed and finalized by the blockchain network.

f) Min Latency (s)

The minimum latency observed for any transaction. This indicates the quickest transaction processing time during the test.

g) Avg Latency (s)

The average latency observed across all transactions in the test. This is an important metric for assessing the typical time required for transactions to be processed under normal load conditions.

h) Throughput (TPS)

The transactions per second (TPS) processed by the blockchain network during the test. This is a key metric for evaluating the system's capacity to handle a given load and its overall performance.

E. Scalability Testing

1) Benchmark configuration for adding more peers to the blockchain network using Hyperledger Caliper

```
name: basic-contract-benchmark
description: Benchmark for the voting contract
workers:
  number: 1
timeout: 180
rounds:
  - label: Register Voters
    description: Test scalability of voter registration.
    txNumber: 1000
    rateControl:
      type: fixed-rate
      opts:
        tps: 50
    workload:
      module: workload/registerVoter.js
      arguments:
        assets: 1000
        contractId: votingcc
  - label: Cast Votes
    description: Test scalability of vote casting.
    txNumber: 1000
    rateControl:
      type: fixed-rate
      opts:
        tps: 50
    workload:
      module: workload/castVote.js
      arguments:
        assets: 1000
        contractId: votingcc
```

Fig. 9. Benchmark Configuration Test with 50 Transactions Per Second

This .yaml file defines a benchmark configuration for evaluating the performance of a voting smart contract using Hyperledger Caliper. It specifies the test structure, including the number of workers, timeout settings, and individual test rounds.

The benchmark is named basic-contract-benchmark and is designed to measure the scalability of key operations in the voting system. It runs with one worker (workers: number: 1), meaning a single process will execute the transactions. The timeout for the benchmark is set to 180 seconds, ensuring the test does not exceed this duration.

The benchmarking process consists of two rounds:

(a) Register Voters

This round test the system's ability to handle voter registration on a scale. It submits 1,000 transactions (txNumber: 1000) at a fixed rate of 50 TPS (rateControl: fixed-rate). The corresponding workload script (workload/registerVoter.js) is executed with 1,000 voter assets (assets: 1000) under the contract ID "votingcc".

(b) Cast Votes

This round evaluates the vote casting process under similar conditions. It also submits 1,000 transactions (txNumber: 1000) at 50 TPS (rateControl: fixed-rate). The workload script (workload/castVote.js) executes the transactions, simulating 1,000 voting actions (assets: 1000) under the same contract ID "votingcc".

This benchmark configuration helps assess the scalability and performance of the voting contract by measuring transaction throughput and latency under controlled conditions.

2) Benchmark configuration for simulating an increasing number transactions per second (TPS) to 200, with the number of peers set to 5 using Hyperledger Caliper

Like the .yaml file defined as in Fig. 9, but the number of transactions per second (TPS) will be increased to 200 with the number of peers set to 5 only.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Scalability Test for Register Voters And Cast Votes Functions By Adding Number Of Peers To 5

The performance results of the Hyperledger Fabric blockchain with different numbers of peers with the help of Hyperledger Caliper were collected and analyzed. Each peer's results were obtained based on the average value of 5 test runs.

TABLE I. PERFORMANCE METRICS FOR 1 TO 5 PEERS

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
1 Peer							
Register Voters	958.4	41.6	50.1	2.040	0.030	0.136	45.66
Cast Votes	252.2	747.8	50.1	2.040	0.038	0.218	45.62
2 Peers							
Register Voters	960.6	39.6	50.1	2.058	0.030	0.136	45.54
Cast Votes	256.8	743.2	50.1	2.052	0.034	0.224	45.64
3 Peers							
Register Voters	959.2	40.8	50.1	2.056	0.032	0.144	45.64
Cast Votes	254.4	745.6	50.1	2.052	0.036	0.226	45.70
4 Peers							
Register Voters	959.0	41.0	50.1	2.042	0.038	0.148	45.82
Cast Votes	257.8	742.2	50.1	2.056	0.038	0.220	45.72
5 Peers							
Register Voters	964.6	34.2	50.1	1.740	0.040	0.140	46.74
Cast Votes	261.2	738.2	50.1	2.046	0.040	0.224	45.88

The data collected is subsequently transformed into two charts for analysis.

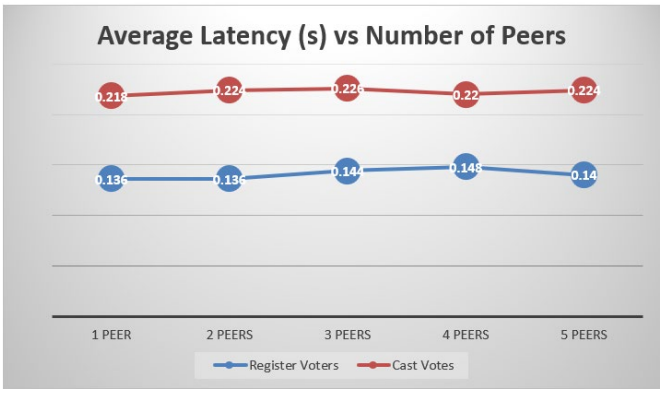

Fig. 10. Average Latency vs Number of Peers

This chart from Fig. 10 illustrates the average latency for the "Register Voters" and "Cast Votes" functions across different peer configurations (1 to 5 peers). The "Register Voters" function consistently showing lower latency compared to "Cast Votes," with values fluctuating slightly as the number of peers increases. For "Register Voters," the latency ranges from 0.136 seconds to 0.148 seconds, showing minimal variation. Meanwhile, the "Cast Votes" function exhibits a higher latency, fluctuating slightly around 0.218 seconds to 0.226 seconds, but overall remains stable. These results indicate that the "Register Voters" function is more efficient in an increased peer configuration, while the "Cast Votes" function maintains consistent performance despite its higher latency. This suggests that the "Cast Votes" function involves additional computational complexity.

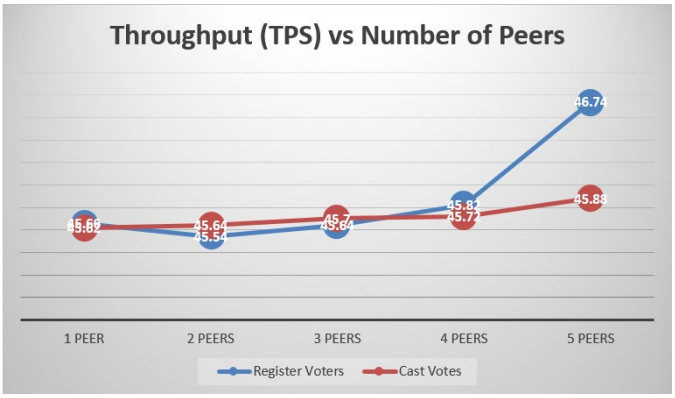

Fig. 11. Throughput (TPS) vs Number of Peers

This chart from Fig. 11 depicts the throughput, measured in transactions per second (TPS), for the "Register Voters" and "Cast Votes" functions across the network configurations with 1 to 5 peers. The "Register Voters" function shows a gradual increase in throughput as the number of peers increases, peaking at 46.74 TPS with 5 peers. In contrast, the "Cast Votes" function maintains a relatively stable throughput, ranging between 45.62 and 45.88 TPS across all peer configurations. These results suggest that while the system effectively scales to handle more transactions for the "Register Voters" function as the number of peers increases, the "Cast Votes" function remains consistent in its performance regardless of peer count, with its peak at 45.88 TPS with 5 peers.

The charts show that adding peers improves performance without bottleneck at orderer node. 5 peers achieving the best balance of throughput and stable latency. Adopting 5 peers offers optimal scalability and efficiency for the system.

B. Scalability Test for Register Voters And Cast Votes Functions By Adding Number of Peers to 5

TABLE II. PERFORMANCE METRICS FOR 5 PEERS WITH 200 TRANSACTIONS PER SECOND

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
5 Peers							
Register Voters	970	30	200.3	2.04	0.04	0.07	142.2
Cast Votes	243	757	200.3	2.04	0.04	0.09	142.4

As compared to Table I for 5 peers, the scalability test (with an increased send rate of 200.3 TPS) shows positive results in terms of increased throughput and decreased latency for both functions, particularly for Register Voters. The system demonstrates scalability, efficiently handling a larger transaction load, especially in Register Voters, where there is a significant improvement in both throughput and latency. Although Cast Votes experienced a higher failure rate, the improvements in latency and throughput are still notable. It is important to note that the higher failure rate for Cast Votes is expected due to the random selection of voters, which leads to variations in successful transactions. Thus, the system performs better in the second test with an increased send rate, confirming its ability to handle high volumes of voter interactions more effectively while maintaining reasonable response times.

V. CONCLUSION AND RECOMMENDATIONS

The study aimed to enhance the performance of online voting systems by leveraging blockchain network design, with a particular emphasis on addressing scalability challenges inherent in blockchain technology. The research successfully developed an architecture based on a single-channel network, capable of handling large volumes of votes while ensuring transparency and integrity throughout the voting process. The findings indicate that adding peers to the network led to performance improvements when without bottleneck at the orderer node. These observations reflect the results of the tests conducted and may vary depending on different experimental setup or real-world conditions. In conclusion, this research provides a robust foundation for the development of scalable and secure permissioned blockchain voting systems. The results highlight the system's potential to handle large volumes of voter interactions efficiently while maintaining high security and transparency.

ACKNOWLEDGMENT

I would like to extend my heartfelt gratitude to the library of Universiti Teknologi Malaysia (UTM) for providing me with unlimited resources to explore a wide range of scholarly articles, conference, and relevant literature from its database.

Finally, I want to mention my friends and family, especially my parents, for their understanding and encouragement throughout my studies, keeping me motivated all the time.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Pranith, M. V., Manoj Kumar, Sanjay H. A., Prashanth, B. S., Likewin Thomas, Srinivasa Murthy, Y. V. (2019). End-to-End Verifiable Electronic Voting System Using Delegated Proof of Stake On Blockchain. *Proceedings of the 5th International Conference on Cyber Security & Privacy in Communication Networks (ICCS) 2019.*

[2] Ben Ayed, A. (2017). A Conceptual Secure Blockchain-based Electronic Voting System. *International Journal of Network Security & Its Applications (IJNSA)*, *9*(3). https://doi.org/10.5121/ijnsa.2017.9301.

[3] R.Krishnamurthy, Rathee, G., & Jaglan, N. (2020). An Enhanced Security Mechanism through Blockchain for E-polling/counting Process using IoT Devices. *Wireless Networks, 26*. https://doi.org/10.1007/s11276-019-02112-5.

[4] Zhang, W., Yuan, Y., Hu, Y., Huang, S., Cao, S., Chopra, A., & Huang, S. (2018). A Privacy-Preserving VotingProtocol on Blockchain. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD),* San Francisco, CA, USA. 401–408. https://doi.org/10.1109/CLOUD.2018.00057.

[5] Fusco, F., Lunesu, M. I., Pani, F., & Pinna, A. (2018). Crypto-voting, a Blockchain based e-Voting System. *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* Seville, Spain. 0IC3K, 223-227. https://doi.org/10.5220/0006962102230227.

[6] Pawlak, M., & Poniszewska-Marańda, A. (2021). Trends in Blockchain-based Electronic Voting Systems. *Information Processing & Management. 58*(4). https://doi.org/10.1016/j.ipm.2021.102595.

[7] Perry., C. B. A. B. a. T. (n.d). Digital Voting with the Use of Blockchain Technology. https://www.economist.com/sites/default/files/plymouth.pdf.

[8] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

[9] Alam, A., Rashid, S. M. Z. U., Salam, M., & Islam, A. (2018). Towards Blockchain-based E-voting System. 2018 *International Conference on Innovations in Science, Engineering and Technology (ICISET),* Chittagong, Bangladesh. 351–354, https://doi.org/10.1109/ICISET.2018.8745613.

[10] Shi, S., He, D., Li, L., Kumar, N., Khan, M. K., & Choo, K.-K. R. (2020). Applications of Blockchain in Ensuring the Security and Privacy of Electronic Health Record Systems: A Survey. *Computers & Security, 97*, 101966. https://doi.org/https://doi.org/10.1016/j.cose.2020.101966.

[11] How Fabric networks are structured¶. (n.d). Hyperledger. Retrieved 10 June from https://hyperledger-fabric.readthedocs.io/en/release-2.5/network/network.html.

[12] Tabatabaei, M. H., Vitenberg, R., & Veeraragavan, N. R. (2023). Understanding Blockchain: Definitions, Architecture, Design, and System Comparison. *Computer Science Review,* 50. https://doi.org/10.1016/j.cosrev.2023.100575.

[13] Jennings, W., & Wlezien, C. (2016). The Timeline of Elections: A Comparative Perspective. *American Journal of Political Science, 60*(1), 219–233. http://www.jstor.org/stable/24583060.

[14] Simons, B., & Jones, D. W. (2012). Internet Voting in the U.S. Commun. *ACM*, *55*(10), 68–77. https://doi.org/10.1145/2347736.2347754.

[15] El Kafhali, S., & Sudhakar, K. (2024). Blockchain-based Electronic Voting System: Significance and Requirements. *Mathematical Problems in Engineering*, *2024*, 1–17. https://doi.org/10.1155/2024/5591147.

[16] Kohno, T., Stubblefield, A., Rubin, A. D., & Wallach, D. S. (2004). Analysis of an Electronic Voting System. *IEEE Symposium on Security and Privacy, 2004. Proceedings*.

[17] Kumar, D. A., & Begum, T. U. S. (2012). Electronic Voting Machine–A Review. *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012).*

[18] Alvi, S. T., Uddin, M. N., Islam, L., & Ahamed, S. (2020). From Conventional Voting to Blockchain Voting: Categorization of Different Voting Mechanisms 2020. *2nd International Conference on Sustainable Technologies for Industry 4.0 (STI).*

[19] Lake, J. (2022). What are the Risks of Electronic Voting and Internet Voting? https://www.comparitech.com/blog/information-security/electronic-voting-risks/