The Indonesian Journal of Computer Science
www.ijcs.net
Volume 13, Issue 5, October 2024
https://doi.org/10.33022/ijcs.v13i5.4378

Implementation and Performance Analysis of Non-Blockchain-based and Blockchain-based Business Licensing Systems Using Hyperledger Fabric

Moza Sajidah Putri Al Muzaffar[1], Farah Afianti[2], Edouardo Bintang Rokatenda
mozasajidah@student.telkomuniversity.ac.id[1], farahafi@telkomuniversity.ac.id[2], rokatendaedo@gmail.com[3]
[1, 2 ,3] School of Computing, Telkom University, Bandung, Indonesia

Article Information

Received : 1 Sep 2024
Revised : 6 Sep 2024
Accepted : 1 Oct 2024

Keywords

Implementation, performance analysis, business licensing system, blockchain, Hyperledger Fabric,

Abstract

The inefficiency of traditional administrative processes highlights the need for innovative digital solutions. This study aims to evaluate and compare the performance of blockchain-based and non-blockchain-based business licensing systems to determine which approach offers better efficiency and security. The blockchain system was developed using Hyperledger Fabric with IPFS for data storage, while the non-blockchain system utilized Node.js with the Express.JS framework and MinIO for storage. Testing involved document upload and retrieval operations based on User ID and Document ID. Results indicate that while the blockchain system offers enhanced data integrity and security, it suffers from significantly slower performance, especially in document upload operations. The non-blockchain system demonstrated faster and more consistent response times, suggesting that in contexts where speed is crucial, a non-blockchain approach may be more suitable, despite the security trade-offs.

A. Introduction

The bureaucratic process for document management in Indonesia remains inefficient and ineffective due to the conventional methods still being practiced, such as requiring citizens to visit government offices and use hardcopy documents. On the other hand, digitalization in Indonesia is rapidly advancing, particularly in the transportation sector, as seen with the emergence of online motorcycle taxis that utilize technology in daily life. The government sector is also beginning to adopt digital transformation, one example being OSS (Online Single Submission), a web-based application that simplifies business licensing for entrepreneurs and MSMEs by integrating various governmental sectors [1]. While OSS is expected to enhance public service efficiency and effectiveness, concerns arise regarding the security of the data processed within the system. Given the large volume of data and sensitive personal information, a technology capable of safeguarding data security and integrity is required. In this context, blockchain emerges as a potential solution. In this context, blockchain emerges as a potential solution. The decentralized blockchain technology, supported by peer-to-peer consensus mechanisms, can enhance data security and availability by preventing modification and forgery by third parties [2].

Blockchain offers various solutions to address challenges related to data management, security, and transparency in services such as e-passports, education certification, and multi-tenant services in local and central government systems [3][4][5]. For instance, blockchain's application in managing biometric data in electronic passports ensures that sensitive information, such as the passport holder's iris and fingerprints, is protected from unauthorized access and tampering. This system uses IPFS (InterPlanetary File System) to store user data permanently and securely, enhancing trust in a more efficient and decentralized e-passport management system [3]. Blockchain technology is also used to enhance the security of education certificates, which are vulnerable to forgery, through the implementation of a private blockchain framework, Hyperledger Fabric, in digital certification systems. This implementation helps maintain certificate data integrity by assigning a unique ID to each certificate and managing user access with different authority levels to prevent duplication and data manipulation by unauthorized parties [6][4]. Additionally, challenges in managing multi-tenant data in the same environment can be addressed through blockchain. This technology ensures that data from different tenants is isolated and accessible only to authorized parties. Moreover, if one node fails, other nodes can continue operating normally, maintaining system continuity and integrity [5]. Blockchain also plays a vital role in integrating various local government subunits, where previously isolated units can be connected through this technology to enhance transparency, efficiency, and inter-departmental trust. By implementing blockchain, the potential for corruption and fraud can be minimized as it prevents stored data from being modified and accessed by unauthorized parties while allowing faster and more secure access to information [7].

Although the benefits of blockchain in enhancing data security and e-government services are evident, further research is needed to identify the most sustainable and suitable blockchain solutions for government applications. This technology is expected to contribute significantly to creating more transparent,

efficient, and trustworthy government services [8][9]. Additionally, implementing blockchain in smart city systems can expedite civil administration processes by reducing direct interactions and minimizing human error. Consequently, this technology has the potential to bring about positive changes in how the government manages and provides public services [10].

The purpose of this research is to implement and analyze the performance comparison between a non-blockchain-based business licensing system and a blockchain-based system (Hyperledger Fabric) in terms of speed, as well as to explore the challenges and obstacles in the implementation process.

B. Research Method

a. Blockchain Service System Design

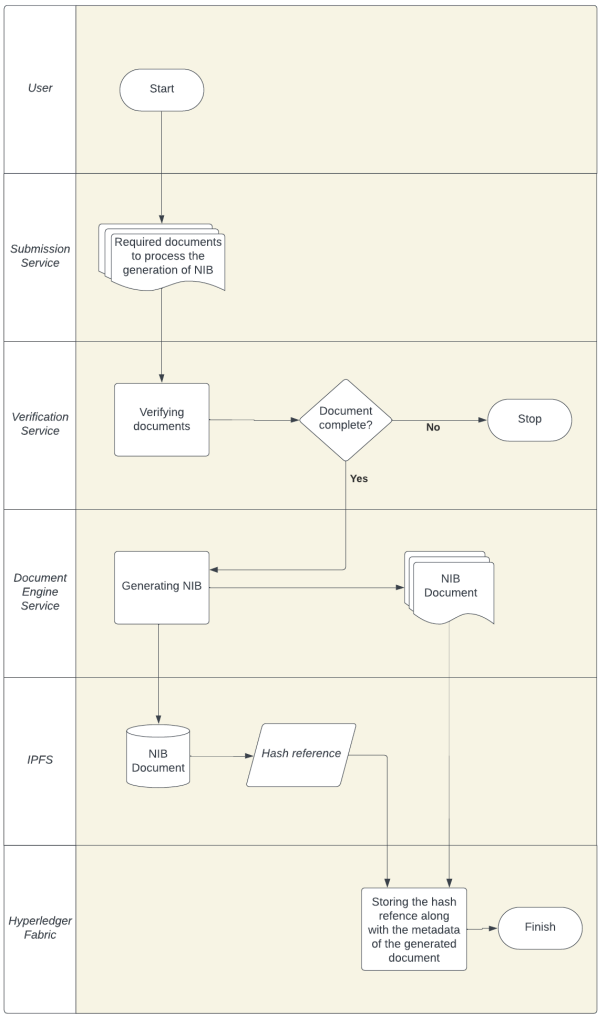

Figure 1. The flowchart of the document upload process for issuing the NIB (Business Identification Number) before it can be retrieved on the blockchain service (storing process).

The blockchain service system is designed with the support of other auxiliary services, such as Submission Service, Verification Service, and

Document Engine Service. These services are part of the various services provided by OSS, where their usage is interrelated and connected to the development of the Blockchain Service system being studied. However, these services will not be discussed further as they fall outside the scope of the system service being developed.

To illustrate the interrelationship of these services, here are the steps taken before the NIB (Business Identification Number) document is ready to be retrieved by the applicant from the system (figure 1). First, the documents required as prerequisites for issuing the NIB are uploaded by the applicant through the Submission Service. Then, these documents undergo a verification process conducted by the Verification Service. After the NIB issuance process is completed, the Document Engine Service forwards the document to the Blockchain Service to be uploaded into the IPFS (InterPlanetary File System) data storage. Once the document is successfully uploaded, IPFS provides an output in the form of a hash reference of the document, which is stored in the blockchain along with other data attributes within the body of a block that is part of a chain of blocks. Figure 2 illustrates the process when a user or applicant wishes to retrieve the NIB document from the system. The detailed process when a document retrieval request is received can be seen in figure 3.

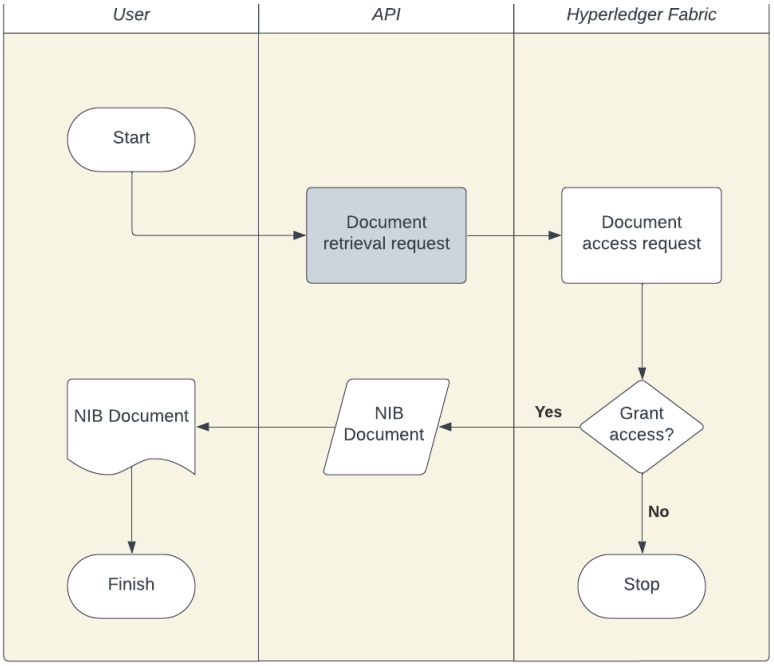

Figure 2. The flowchart of the NIB document retrieval process through the blockchain service (retrieval process).

b. Document (Engine) Service

This service is closely related to the operation of the Blockchain Service under study and serves as a simulation of the actual service. The Document Service is used to generate the NIB document, which will be stored in the Blockchain Service system (Figure 6). Although it is primarily intended to

support the Blockchain Service, the Document Service is also used in the non-blockchain system for experimental purposes (Figure 7).

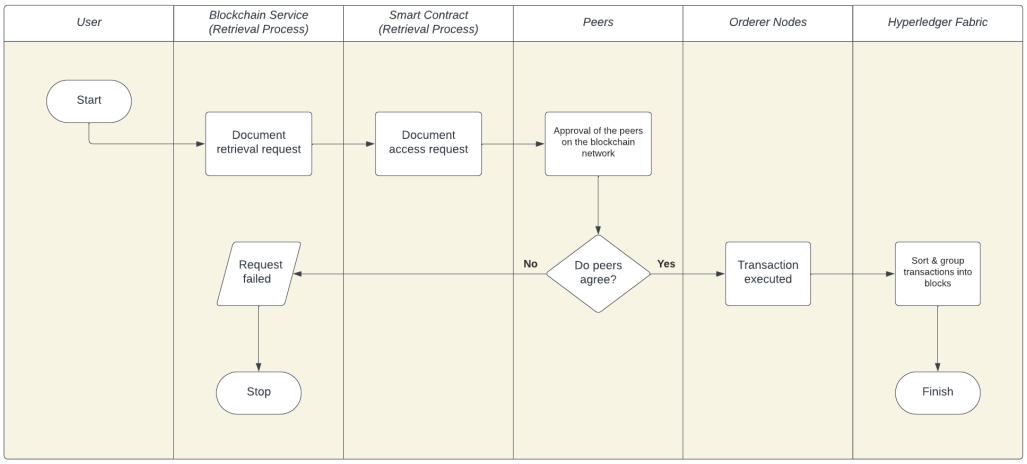

Figure 3. User interaction with the Hyperledger Fabric blockchain framework as a detailed part of the document retrieval request.

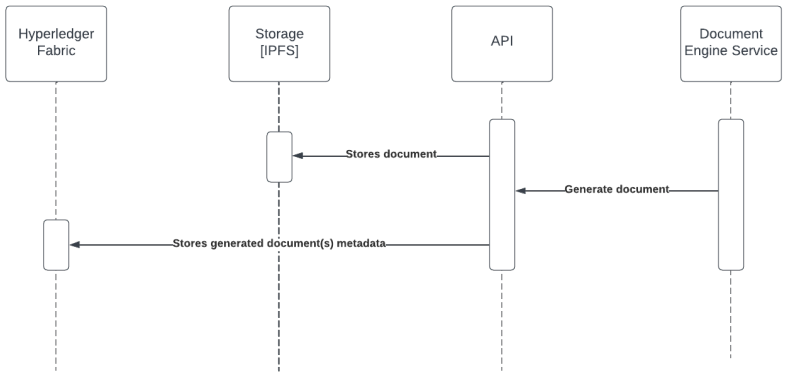

Figure 6. The Document Engine Service forwards the officially issued NIB document to the API for storage in the IPFS data storage, and the metadata of the document is stored on the Hyperledger Fabric blockchain.

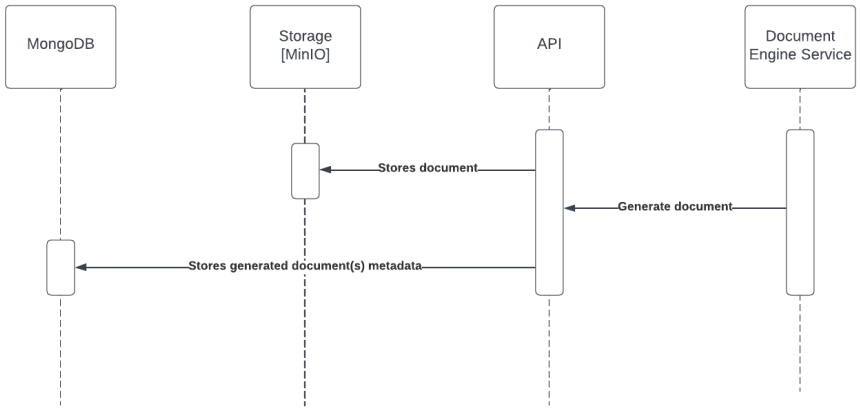

Figure 7. The Document Engine Service forwards the officially issued NIB document to the API for storage in the MinIO data storage, and the metadata of the document is stored in the MongoDB database.

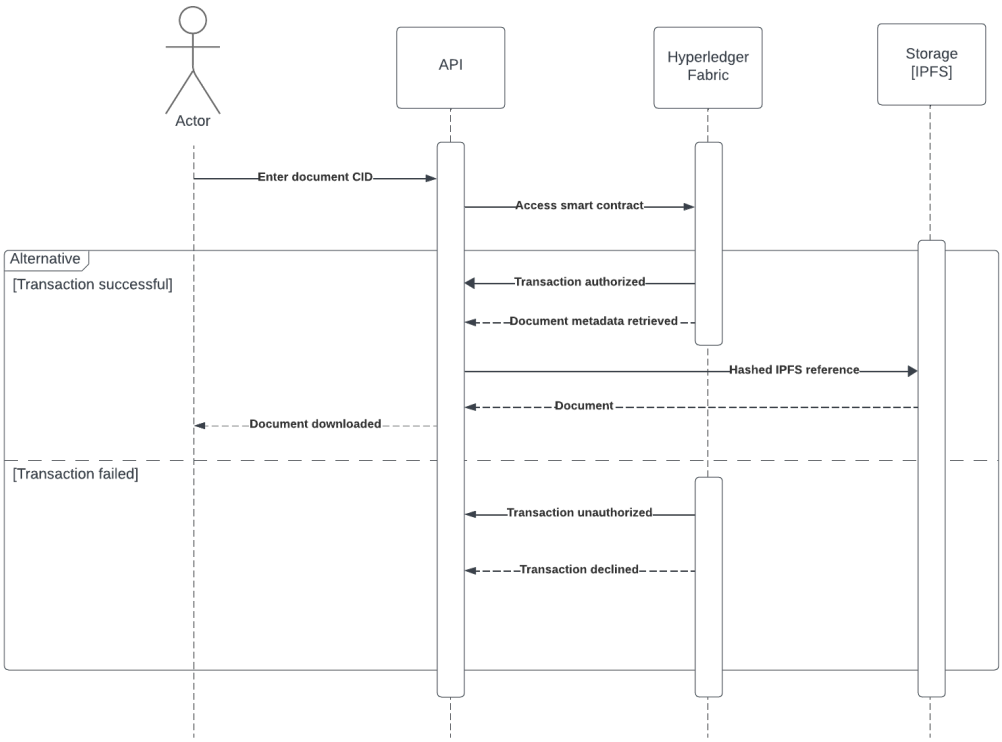

Figure 8. The process flow that occurs when a user attempts to retrieve a document stored in the Blockchain Service system.

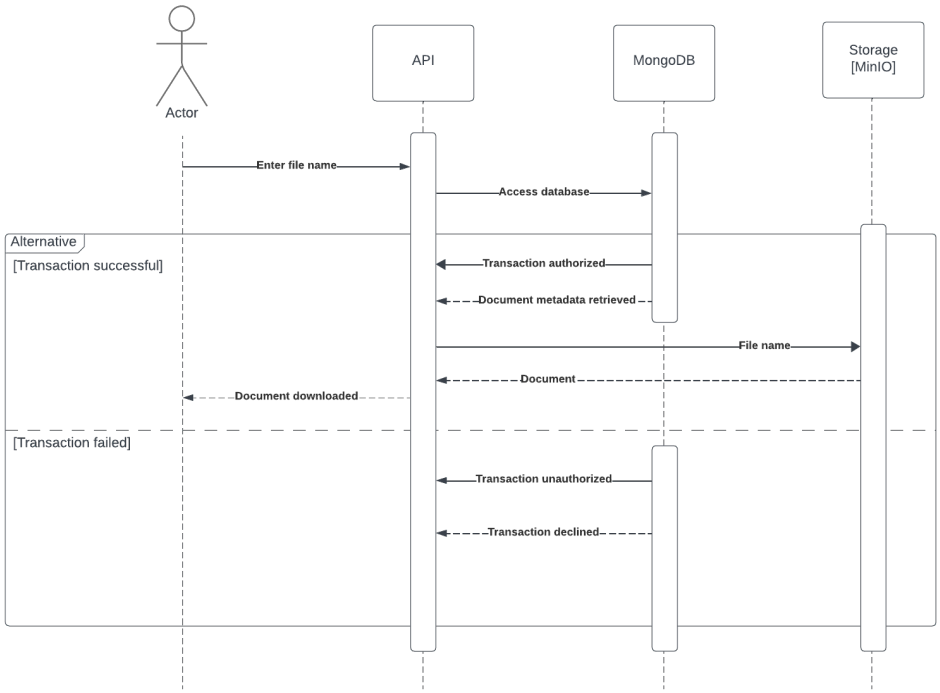

Figure 9. The process flow that occurs when a user attempts to retrieve a document stored in the non-blockchain system.

c. System Design for Testing

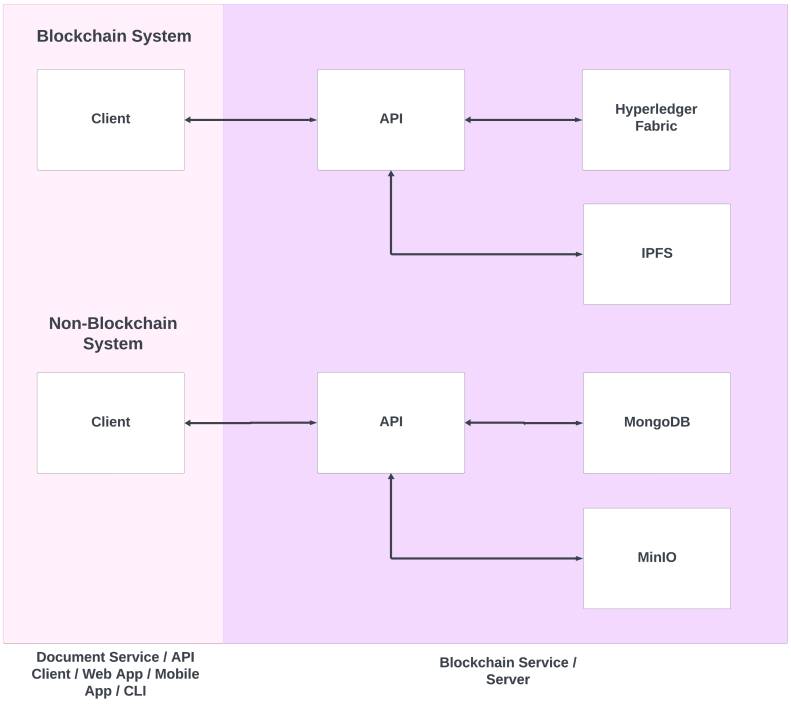

Figure 10. The architecture of the system that being developed.

In this experiment, two systems were created: a blockchain-based system and a non-blockchain system. Both have clients and APIs connected to networks and object data storage like IPFS and MinIO. In Figure 10, the pink section represents clients such as Document Service, API client, Web/Mobile App, or CLI. The Document Service sends the NIB document in PDF format to be stored in each system through an API, which acts as a communication bridge between applications [11].

On the other hand, the purple section represents the Blockchain Service for the blockchain-based system and the Server for the non-blockchain system. The Blockchain Service uses two main technologies: Hyperledger Fabric and IPFS. Hyperledger Fabric is an open-source, permissioned, enterprise-grade distributed ledger technology (DLT) platform that supports smart contracts in general programming languages without requiring cryptocurrency, thereby improving performance and maintaining transaction confidentiality [12]. IPFS is a peer-to-peer protocol for managing and transferring data through content-based addressing in the form of hashes. One of its primary applications is decentralized data publishing [13]. Both sides are connected on the same network using Tailscale [14].

d. System Functionality

Both systems will have the same functionality, allowing users to:

- Register a new account
- Log in

- Perform searches based on user ID, document ID, and document name
- Retrieve or download the NIB document

e. Sequence Diagram for Non-Blockchain and Blockchain Services System

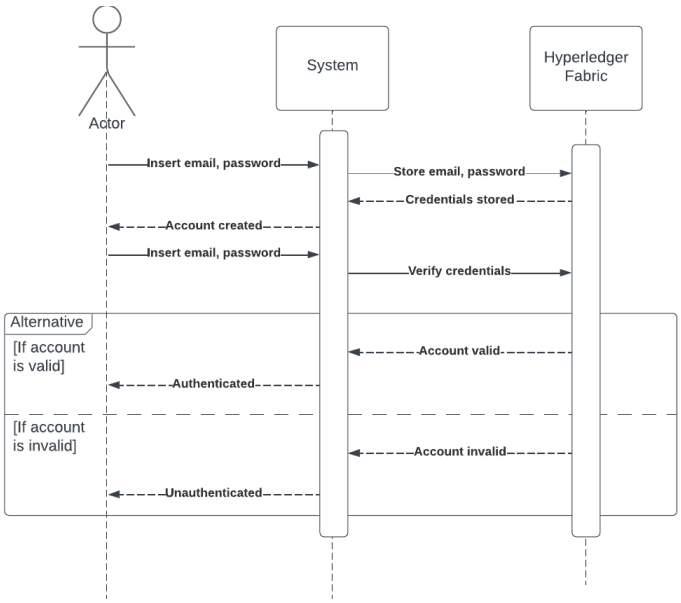

Figure 4. The user performs the process of creating a new account or logging in to the Blockchain Service system.

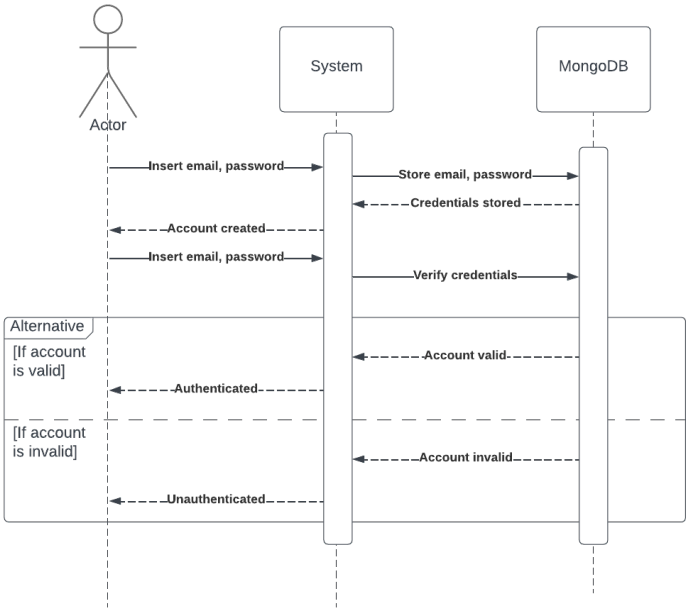

Figure 5. The user performs the process of creating a new account or logging in to the non-blockchain system.

This sequence diagram explains the process of using both systems. The account registration and login process for the blockchain service is shown in Figure 4, and for the non-blockchain system in Figure 5. The process of retrieving official business licensing documents stored in IPFS for the blockchain service is explained in Figure 8. When a user retrieves a document, the system verifies the transaction through a smart contract on HyperLedger Fabric. If the conditions are met, IPFS sends a hash reference to access the document according to the user ID. A similar process for the non-blockchain system using MinIO is shown in Figure 9.

f. Blockchain Structure

The data attributes stored in the body block component of a blockchain include User ID, Document ID, Document Name, Document Type, Timestamp, and IPFS hash reference. Meanwhile, the metadata stored in the header block component of a blockchain includes block hash, previous block hash, timestamp, and Merkle Root. This structure can be seen in figure 11.

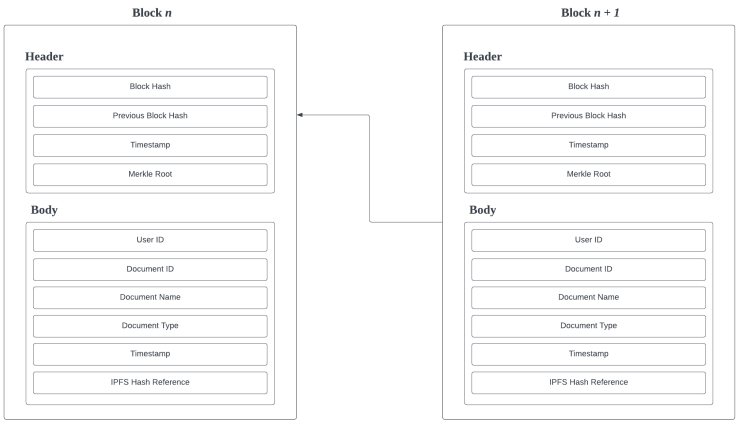

Figure 11. A blockchain is a series of blocks that are connected to each other. Each block has a header and a body component, with each component containing stored data.

g. Database Structure

The non-blockchain system uses MongoDB to store metadata related to issued NIB documents, such as User ID, Document ID, Document Name, Document Type, and Timestamp. Additionally, MongoDB is used to store data such as email and password for new account creation or user login purposes.

h. The Development of System being Tested

i. Blockchain Service System

The Blockchain Service is built using the Hyperledger Fabric test network to test the smart contracts and applications developed [15]. As shown in Figure 12, the test network is part of the on-chain entities that

directly interact with the blockchain and manage transaction requests from client applications in the off-chain entities. The client application connects to the test network through a REST API Server that uses the Fabric SDK with Node.js as the interface for communicating with Hyperledger Fabric. This test network has one channel that hosts two organizations: Org1 as OSS (blockchain service) and Org2 as the user, with each organization having a Root CA that manages the identities of network participants. Each organization has an endorsing peer that endorses transaction proposals and a committing peer that commits transactions to the ledger. The orderer, connected to the channel, is responsible for ordering transactions and distributing blocks to the organizations in the network. The ledger consists of a blockchain, which stores immutable transactions, and a world state, which speeds up transactions by storing key-value pairs of the latest data.

In this network, the installed chaincode, developed using the Go programming language, is tested. The REST API Server is developed using Node.js and the Express.js framework and connects to IPFS through the Helia library. The entire system is hosted on the same server and then tested.

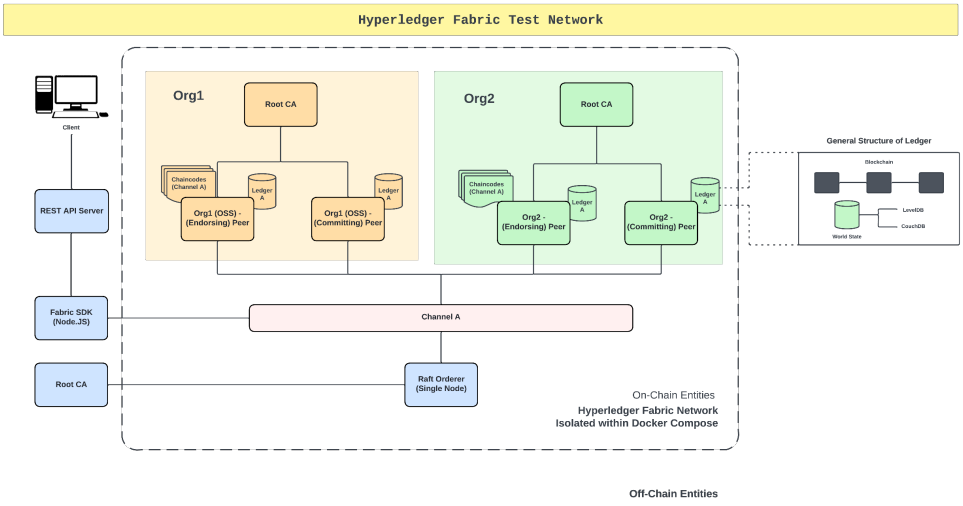

Figure 12. The test network architecture adapted to the network design intended for the Blockchain Service system.

ii. Non-Blockchain System

The Non-Blockchain system is built using Node.js and the Express.js framework, connected to a MongoDB database to store metadata related to issued NIB documents, as well as data such as email and password for new account creation or user login purposes. Additionally, the system is connected to MinIO, deployed using Railway App as an object data storage solution that stores NIB documents. The entire system is hosted on a server and then tested.

iii. Server

The experiments involving both systems are conducted on a server hosted using VM (Virtual Machine) instances on Google Compute Engine located in Singapore. Table 1 shows the virtual machine configuration used for hosting the server. Additionally, Tailscale configuration is applied to the server to enable connection with the client.

Table 1. Configuration of the virtual machine used for hosting the server on Google Compute Engine, utilized by both systems being tested.

Attributes	Configuration
Region	asia-southeast1 (Singapore)
Zone	asia-southeast1-c
VM Configuration	E2-standard-2, 1-2 vCPU (1 shared core), 8 GB RAM
Operating System	Ubuntu 24.04
Architecture	x86_64

iv. Client

In this experiment, the client is a computer set up locally and connected to Tailscale to communicate with the server, allowing for testing that involves interaction with the API.

v. Testing Tool

The experimental testing is conducted using a program written in JavaScript, run in a Node.js environment, which is customized to automate the testing of the two systems being evaluated.

i. Data Collection Techniques

The data used for testing is randomly generated using the Faker API. This is done to protect the original data, which is confidential and sensitive. The API has the ability to generate a number of realistic mock data based on real data for testing and development purposes [16]. In this experiment, the API is configured according to the testing needs, where 30 data entries are generated using a local Indonesian instance to ensure the data is as similar as possible to the population data of Indonesia. The generated data includes, among other things, full names, which are a combination of first and last names, and a randomly generated UUID string for the docId.

The collected data is then used to generate PDF documents that imitate NIB documents. This is achieved with the help of the JavaScript library jsPDF, which is used to generate PDFs through JavaScript [17]. Both the data and the generated documents are processed through the document (engine) service, which functions to generate NIB documents (Figures 6 & 7).

j. Analysis Techiques

The analysis will be conducted on the collected data related to the time required to store/create documents and to query files using User ID and Doc ID from both IPFS and MinIO. The time measurement does not include the document download process due to limitations in the application used for testing. This testing is applied to both systems by connecting the locally run

Document (Engine) Service with the Blockchain Service hosted on GCP using Axios, a JavaScript library used to make HTTP requests from programs written in Node.js [18]. After the connection is established, the program hosted on the Document (Engine) Service will be run 5 times, generating a total of 150 data entries stored in .txt files for each test case, distinguished by timestamp. Each program run will output the minimum, maximum, and average response times. The output will then be averaged, and the results will be analyzed by comparing the results obtained from both systems.

k. Experiment Procedure

The following outlines the steps taken in this experiment, separated based on the system being tested:

i. Blockchain Service System

1. Start the test network.
2. Create a test channel on the network that will host both organizations.
3. Run the installed chaincode.
4. Enroll the admin and register the client on the test network before running the client application.
5. Then, run the client application to test the Blockchain Service according to the testing scenarios. This application will also automatically record and store the time required for each testing scenario.

ii. Non-Blockchain System

1. Run the non-blockchain application.
2. Then, run the client application to test the Non-Blockchain Service according to the testing scenarios. This application will also automatically record and store the time required for each testing scenario.

C. Result and Discussion

Comparison of Average Document Upload Time

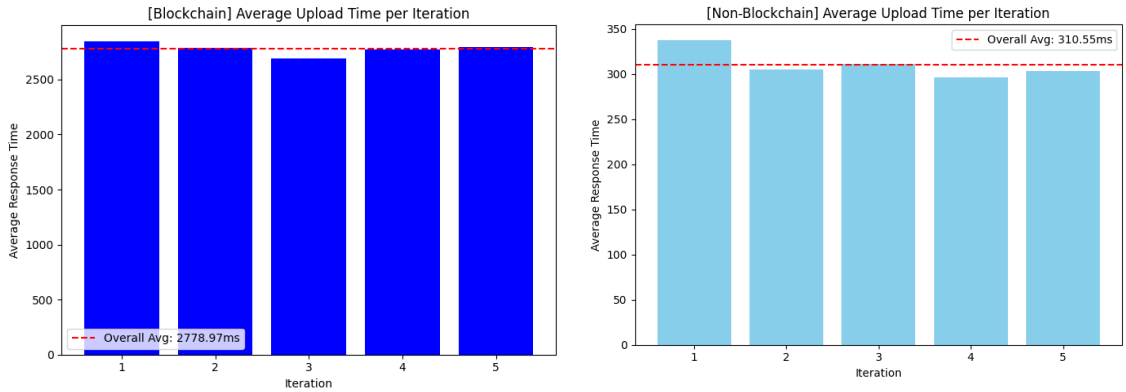

Figure 13. Bar chart illustrating the comparison of average document upload times per iteration in the Blockchain Service and non-blockchain systems.

Although Figure 13 shows that the Blockchain and Non-blockchain systems have similar data projections, particularly in the first iteration where the document upload time is longer compared to subsequent iterations, there is a significant difference in the time required. The Blockchain system, on average, takes 2850ms in the first iteration, which is 743% slower than the Non-blockchain system, which only takes 338ms. Overall, the Blockchain system takes an average of 2778.97ms to upload documents, much slower than the 310.55ms on the Non-blockchain system, indicating that the Non-blockchain system has superior performance.

Comparison of Average Query Time by User ID

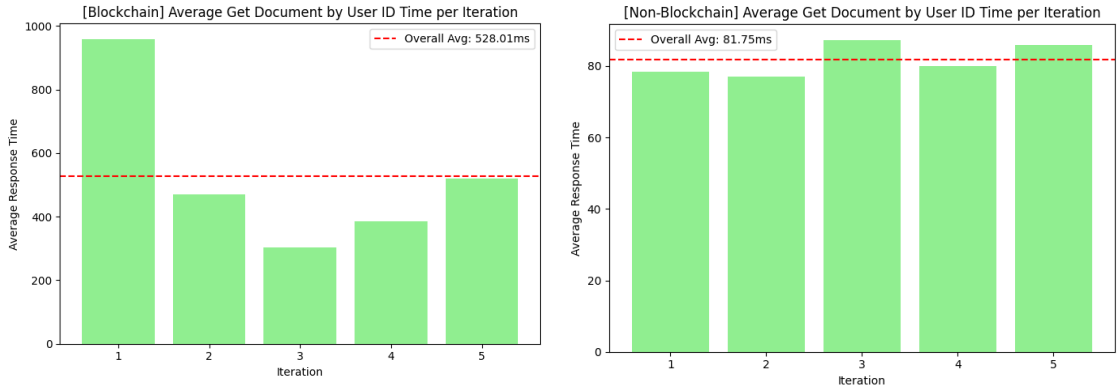

Figure 14. Bar chart illustrating the average document query time based on User ID per iteration in both system.

The first image in figure 14 shows the average response time for retrieving documents based on User ID in the blockchain-based system, while the second image shows the same for the non-blockchain system. The blockchain system shows a significantly higher response time compared to the non-blockchain system, indicating that the process of retrieving documents by User ID is slower in the blockchain-based system. The average response time for the blockchain system is more than six times higher than the non-blockchain system, demonstrating a significant difference in time efficiency. The blockchain system shows greater variability in response times between iterations, whereas the non-blockchain system maintains better consistency in similar cases.

Comparison of Average Query Time by Document ID

The first image in figure 15 shows the average response time for retrieving documents based on Document ID in the blockchain-based system, while the second image shows the same for the non-blockchain system. The blockchain system shows a significantly higher response time compared to the non-blockchain system, indicating that the process of retrieving documents by Document ID is slower in the blockchain system. The average response time for the blockchain system is about four times higher than the

non-blockchain system, indicating a significant difference in efficiency. The blockchain system shows greater fluctuation in response times between iterations, particularly in the third iteration, while the non-blockchain system exhibits better consistency.

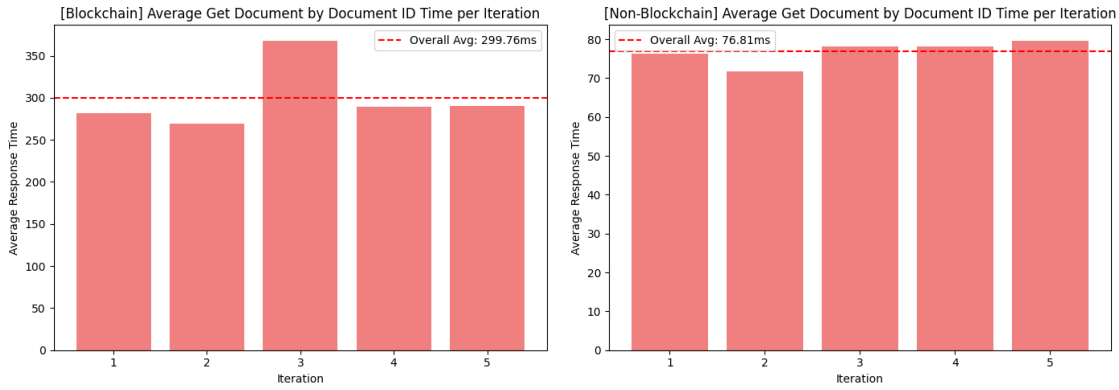

Figure 15. Bar chart illustrating the average document query time based on Document ID per iteration in both system.

Comparison of Average Time for All Operations on Both Systems

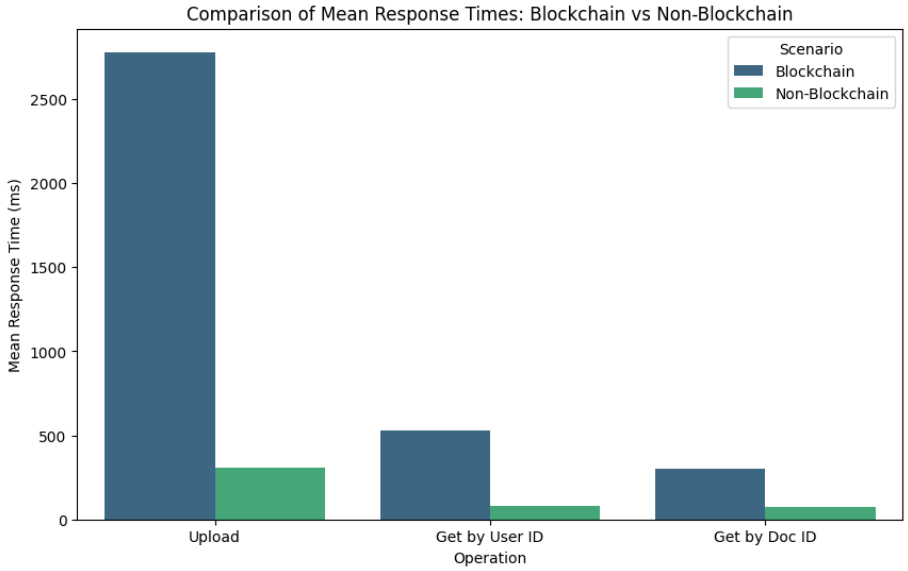

Figure 16. Bar chart illustrating the comparison of average upload times, and document query times based on User ID and Document ID per iteration in the Blockchain Service and Non-Blockchain systems.

This bar chart in figure 16 compares the average response times for three main operations: upload, query by User ID, and query by Document ID between blockchain-based and non-blockchain systems. The blockchain system shows much higher response times for all operations, indicating that blockchain adds significant overhead compared to the non-blockchain

system. The upload operation is the most affected by the use of blockchain, showing nearly ten times the response time compared to the non-blockchain system, followed by the query by User ID and query by Document ID operations. All operations on the blockchain system are consistently slower than on the non-blockchain system. This indicates that, while blockchain may offer advantages in terms of security and transparency, there are significant trade-offs in terms of speed and efficiency.

D. Conclusion

This study shows that the Hyperledger Fabric blockchain-based business licensing system with IPFS as data storage has significantly slower performance compared to the non-blockchain system developed using the Express.JS framework on Node.js with MinIO as data storage in terms of processing speed. Across all types of operations tested, such as document upload, document retrieval by User ID, and Document ID, the blockchain system consistently exhibited higher response times, with the most notable difference occurring in the document upload operation. While blockchain offers potential advantages in security, particularly in maintaining data integrity and preventing manipulation, these results indicate that a compromise must be made, which in this case is speed.

Additionally, the challenges and obstacles in implementing the blockchain system, such as the complexity of the architecture and the overhead associated with network management and consensus, also contribute to the decreased performance compared to the simpler and faster non-blockchain system. Therefore, although blockchain-based systems may be more reliable in terms of security, their implementation requires careful consideration, especially in contexts where speed is a critical factor in the operation of a system.

E. References

[1] "OSS - Sistem Perizinan Berusaha Terintegrasi Secara Elektronik." Accessed: Oct. 15, 2023. [Online]. Available: https://oss.go.id/informasi/artikel/presiden-jokowi-resmikan-peluncuran-oss-berbasis-risiko

[2] X. Deng, W. Huang, and X. Tang, "Basic Technology," in *Blockchain Application Guide*, X. Tang, X. Deng, and R. Bie, Eds., Singapore: Springer Nature Singapore, 2022, pp. 3–17. doi: 10.1007/978-981-19-5260-9_1.

[3] N. Jahan and S. Reno, "Utilizing Hyperledger-Based Private Blockchain to Secure E-Passport Management," in *Data Intelligence and Cognitive Informatics*, I. J. Jacob, S. Kolandapalayam Shanmugam, and I. Izonin, Eds., in Algorithms for Intelligent Systems. , Singapore: Springer Nature Singapore, 2023, pp. 579–593. doi: 10.1007/978-981-19-6004-8_46.

[4] Md. Jahid Alam, S. Hossain, A. Shekh, and S. Reno, "Utilizing Hyperledger Fabric Based Private Blockchain and IPFS to Secure Educational Certificate Management," in *2022 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, Naya Raipur, India: IEEE, Dec. 2022, pp. 5–11. doi: 10.1109/WIECON-ECE57977.2022.10151082.

[5] A. Sharma and P. Kaur, "Tamper-proof multitenant data storage using blockchain," *Peer--Peer Netw. Appl.*, vol. 16, no. 1, pp. 431–449, Jan. 2023, doi: 10.1007/s12083-022-01410-8.

[6] S. Reno, M. Ahmed, S. A. Jui, and S. Dilshad, "Securing Certificate Management System Using Hyperledger Based Private Blockchain," in *2022 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, Chittagong, Bangladesh: IEEE, Feb. 2022, pp. 46–51. doi: 10.1109/ICISET54810.2022.9775834.

[7] V. A. Bharadi, P. P. Ghag, S. R. Chavan, S. S. Gawas, and A. Kazi, "Integrating Blockchain with Local Public Service System," in *IC-BCT 2019*, D. Patel, S. Nandi, B. K. Mishra, D. Shah, C. N. Modi, K. Shah, and R. S. Bansode, Eds., in Blockchain Technologies. , Singapore: Springer Singapore, 2020, pp. 93–103. doi: 10.1007/978-981-15-4542-9_9.

[8] I. Lykidis, G. Drosatos, and K. Rantos, "The Use of Blockchain Technology in e-Government Services," *Computers*, vol. 10, no. 12, p. 168, Dec. 2021, doi: 10.3390/computers10120168.

[9] E. A. Franciscon, M. P. Nascimento, J. Granatyr, M. R. Weffort, O. R. Lessing, and E. E. Scalabrin, "A Systematic Literature Review of Blockchain Architectures Applied to Public Services," in *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Porto, Portugal: IEEE, May 2019, pp. 33–38. doi: 10.1109/CSCWD.2019.8791888.

[10] V. Shah, K. Padia, and V. B. Lobo, "Application of Blockchain Technology in Civil Registration Systems," in *IC-BCT 2019*, D. Patel, S. Nandi, B. K. Mishra, D. Shah, C. N. Modi, K. Shah, and R. S. Bansode, Eds., in Blockchain Technologies. , Singapore: Springer Singapore, 2020, pp. 191–204. doi: 10.1007/978-981-15-4542-9_16.

[11] "What Is an API (Application Programming Interface)? | IBM." Accessed: Sep. 01, 2024. [Online]. Available: https://www.ibm.com/topics/api

[12] E. Androulaki *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, Porto Portugal: ACM, Apr. 2018, pp. 1–15. doi: 10.1145/3190508.3190538.

[13] "What is IPFS? | IPFS Docs." Accessed: Sep. 01, 2024. [Online]. Available: https://docs.ipfs.tech/concepts/what-is-ipfs/#defining-ipfs

[14] "Tailscale · Best VPN Service for Secure Networks." Accessed: Sep. 01, 2024. [Online]. Available: https://tailscale.com/

[15] "Using the Fabric test network — Hyperledger Fabric Docs main documentation." Accessed: Sep. 01, 2024. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/test_network.html

[16] "@faker-js/faker," npm. Accessed: Sep. 01, 2024. [Online]. Available: https://www.npmjs.com/package/@faker-js/faker

[17] *parallax/jsPDF*. (Aug. 31, 2024). JavaScript. Parallax. Accessed: Sep. 01, 2024. [Online]. Available: https://github.com/parallax/jsPDF

[18] "Axios." Accessed: Sep. 01, 2024. [Online]. Available: https://axios-http.com/docs/intro