Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

QR Code-Based Smart Document Implementation Using Distributed Database And Digital Signature

Muhammad Waqas Ayub, Idris Winarno, Amang Sudarsono
147645796552@pasca.student.pens.ac.id, idris@pens.ac.id, amang@pens.ac.id
Teknik Elektro, Politeknik Elektronika Negeri Surabaya

Article Information

Submitted: 11 Jan 2024
Reviewed : 29 Jan 2024
Accepted: 10 Feb 2024

Keywords

QR Code, Distributed Database, Smart Document, Digital Signature, Hashing

Abstract

In digitized world, digital documents are essential for information sharing. However, some organizations continue to place their reliance in traditional hard-copy formats concerning about the legitimacy of documents. This study presents an innovative approach to document verification with digital signatures, distributed databases, and QR codes. Using a two-step process for data integrity and document authentication, the study approach entails developing a Smart Document with a QR code and digital signature. For increased security and scalability, the system design distributes hash fragments among several databases using the hash split approach. The system's excellent performance, resistance to sluggish HTTP-based attacks, and effectiveness in document verification are highlighted by the results and debates. The report ends with recommendations for future improvements to strengthen the system's resilience, like implementing more secure database engines and enhancing fault tolerance. In conclusion, this method offers a viable way to verify documents in hardcopy and electronic formats in a secure and scalable manner.

A. Introduction

A contract, which defines rights and obligations, has moved to electronic formats, with digital signatures authenticating identification across many digital resources. Electronic signatures improve speed, security, and validity, and play an important role in ensuring the integrity of digital exchanges [1]-[3]. This study proposes a Smart Document architecture to address the growing concern of document counterfeiting. It verifies documents using digital signatures encoded in QR codes and uses blockchain for safe, transparent, and immutable record-keeping, ensuring integrity and authenticity. The framework improves document verification in electronic and paper formats, ensuring data availability and dependability [4]. The ongoing problem of document fraud in South Africa, offering a system that incorporates 2D barcodes, digital signatures, OCR, cryptographic hashing, and blockchain for improved document verification. Despite current demonstration trials, the goal is to build a viable system to check document integrity and identify tampering in the face of widespread hardcopy document use [5]. To fight the issue of false academic diplomas, this article suggests a blockchain-based solution based on Hyperledger Fabric. By integrating blockchain technology, the solution ensures tamper-proof records, transparency, and trust, allowing organisations to securely check certificate authenticity through end-to-end encryption [6]. Combining barcodes with public key encryption adds a strong cryptographic layer and offers concrete and effective document security with unique identifiers, all of which work together to guarantee authenticity and integrity across a range of media [7]. A decentralized, utility- and locality-aware method for maximizing replica placement in peer-to-peer cloud storage that takes latency and bandwidth into account is the Pyramid replication algorithm. The outcomes of the simulation show that Pyramid outperforms current algorithms in terms of both utility and locality awareness [8]. For high-throughput block storage, IPFS is a peer-to-peer file system that connects devices using a content-addressed approach. It looks like a BitTorrent swarm. It creates a decentralized Merkle DAG that eliminates single points of failure and supports blockchains, versioned file systems, and the Permanent Web [9]. The benchmarking findings for Hyperledger, a distributed ledger based on blockchain technology, are presented in the study. In a heavy workload scenario, Hyperledger outperformed typical database systems with a runtime of 74.30s, latency of 73.40ms, and 257 transactions per second. The investigation demonstrates how efficient Hyperledger is—it can handle higher data volumes and perform a single transaction 80–200 times faster than MySQL [10]. To solve problems with conventional centralized storage systems, this paper presents a decentralized transaction technique for cloud storage that makes use of a reverse VCG-based auction mechanism and smart contracts. The suggested concept, which is put into practice on an Ethereum private chain, shows how storage resources can be traded competitively to ensure both economic efficiency and safety [11]. Blockchain-based middleware ChainSQL allows distributed databases and blockchain to work together smoothly for auditable and tamper-proof data management. In terms of performance, it is on par with other blockchain systems and databases; it takes care of data integrity issues and guarantees quick query processing [12]. This paper introduces a Key-Value database for ship virtual test platforms, designed for distributed systems,

supporting heterogeneous data types and large files. The proposed system demonstrates efficiency, scalability, and reliability, offering a solution for managing complex data in ship simulation and testing [13]-[14]. This provides a unifying model that breaks down the essential functions of various data-intensive systems in the age of massive and quickly expanding data. This makes it easier for different communities to understand, compare, and identify research trends and future directions. It tackles issues with accessibility, analysis, and storage while providing insights into how contemporary software systems are changing [15].

B. Contribution

One big issue that has been faced in the past is the slow performance of early blockchain technologies. The first blockchain networks were limited to a few transactions per second and could take up to an hour to guarantee the authenticity of transactions [16]. It is hoped that the results of this study will help to develop a system of the kind that features the tamper-resistance of the blockchain and the fast query processing of the distributed databases. This work showcases MySQL, which has the distributed and audibility features of the blockchain quick query processing, and the well-designed data structure of the distributed databases.

C. Research Method

In this paper, a technique for building a Smart Document with a QR code and digital signature is proposed as shown in Figure 1. In addition to offering a shared, immutable, and transparent history for preserving these documents without depending on a third party, a smart document seeks to confirm the legitimacy and integrity of a document.

The owner's digital signature must be printed on the document to facilitate a quick verification process. A hard copy of a document cannot be used to verify a digital signature like an electronic document can as shown in Figure 2. The usage of the QR code on the Smart Document is suggested as a solution to this issue. A QR code scanner may read the embedded QR code on the Smart Document to extract information from the distributed database for verification. The benefit of the suggested method is that the document verifier can rely on manual verification procedures, which are laborious, tiresome, and difficult.

Flowchart

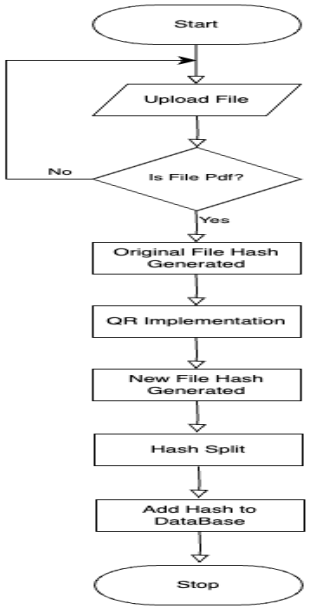

Figure 1. A Flowchart of the Hash Split Technique, dividing a hash into five parts and distributed among five databases.

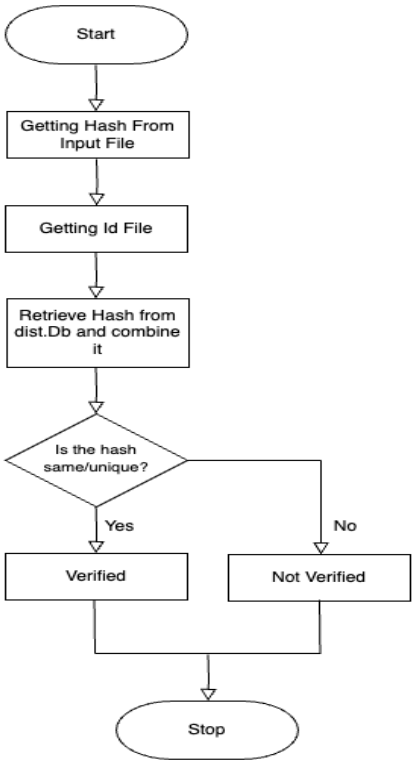

Figure 2. The Verification Process for Hash Combine, The Hash Retrieved Dist. DB and Compared to A New Hash to Validate Document Authenticity.

D. System Design

When a user uploads a PDF file, it is normal practice to check the integrity and authenticity of the supplied content. A two-step procedure can be used to accomplish this.

First, the system can generate a cryptographic hash of the uploaded PDF file. A hash is a unique fingerprint of the file's contents. This hash is then saved in a secure database for future reference. In the future, this saved hash can be compared to see if the file has been altered or tampered with. If the hash of the uploaded file matches the hash of the saved file, it shows that the file remains untouched and is the same as the original as shown in Figure 3.

The second stage is to improve the security and usability of the uploaded PDF. This is accomplished by including a QR code within the PDF page. The QR code may include important metadata or a link to the database containing the hash. This connection allows for a quick and easy approach to validate the document's legitimacy. When the QR code is scanned, the system can get the previously saved hash and compare it to the current document's hash. If they match, consumers can be sure that the PDF file is unchanged and trustworthy.

This two-step approach protects data integrity and enables easy authentication of the PDF file. It is a helpful addition to applications that require document security and data integrity, such as legal papers, certificates, contracts, or any other context where the content's dependability is required. Users can have confidence in the integrity of their uploaded PDF files by combining hash creation and QR code implementation.

Splitting Hash

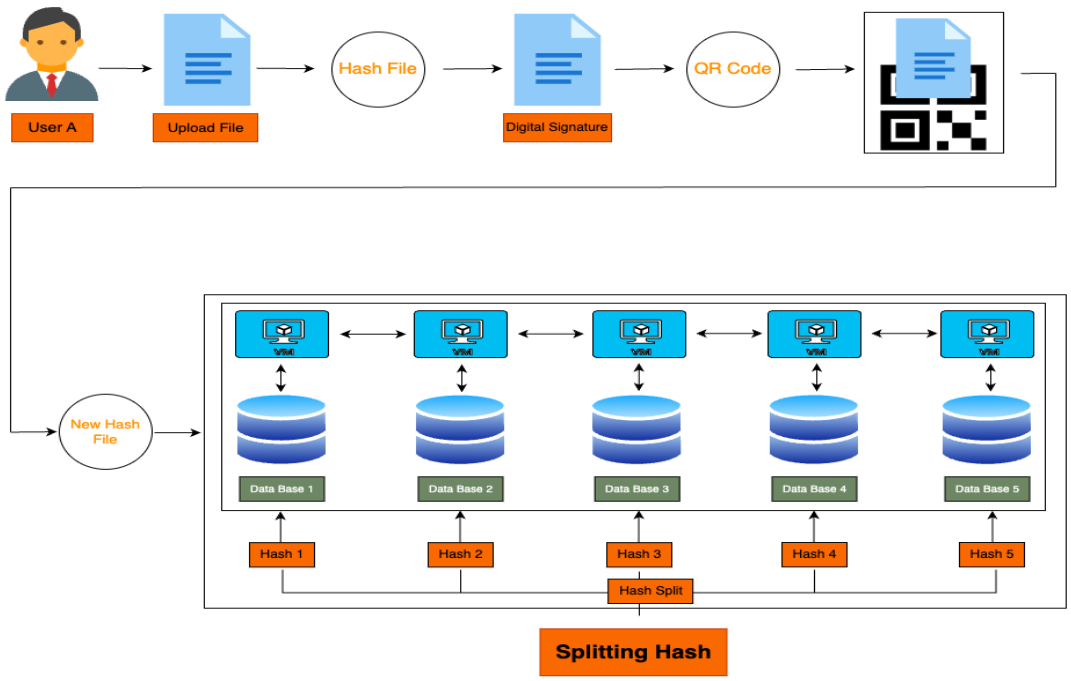

Figure 3. The Hash Split Block Diagram, Dividing Hash into Five Parts, and Distributing It Over Five Databases.

The architecture is created for increased data security, fault tolerance, and distribution in a system where the generated hash, after applying a QR code on the uploaded file, is divided into five parts and saved in five distinct databases. Here's a more detailed explanation:

Data Protection

The splitting of the hash into five parts and storing them in separate databases improves data security dramatically. Even if one database is hacked, it will only have a portion of the original hash. This adds an additional layer of security, making it extremely difficult for unauthorized individuals to reassemble the complete hash.

Load Dispersion

By using five virtual computers for database storage, load dispersion is made more efficient. Each virtual machine can manage a percentage of the total workload, enhancing system performance and reaction times, particularly when dealing with many hash verifications and retrievals.

Scalability

The utilization of virtual machines allows the system to scale. Additional virtual machines can be readily added to the system to accommodate the increased load as the volume of uploaded data and hash verifications grows. This ensures the system's ability to adapt to changing needs throughout time.

Geographic Redundancy

These virtual machines can be deployed across several geographic areas to improve data resilience even further. This reduces the risk of data loss due to localized calamities or outages, assuring data availability even when regional disruptions occur.

Verification

When a person or system must validate a document, they begin by creating a fresh hash of the document in question. Typically, the new hash is computed using the same cryptographic methodology and methods as the original hash.

The original hash must now be recovered from the five distinct virtual computers where it was partitioned and stored. Each virtual machine contains a portion of the original hash, and the system must connect with all five of them to rebuild the entire hash. This distributed strategy assures that no single point of failure can jeopardize the data's integrity.

After obtaining all five fragments of the original hash, they are concatenated to generate the whole hash. This rebuilt hash is then compared to the newly computed hash for verifying the document as shown in Figure 4.

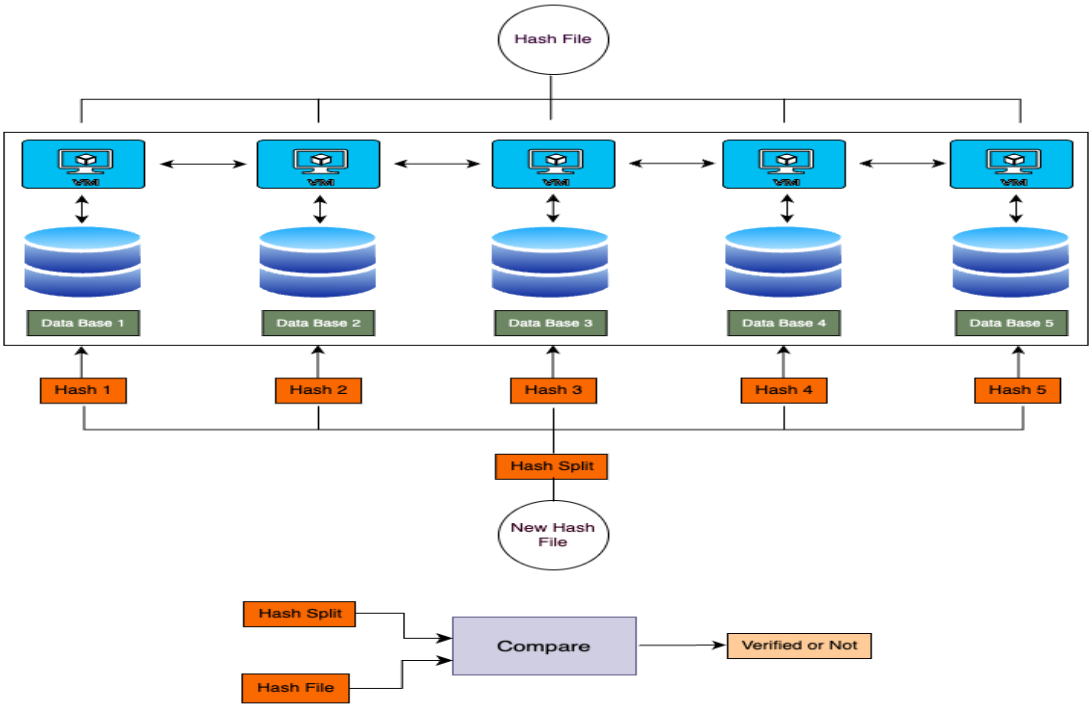

Figure 4. The Hash Verification Block Diagram, Hashes Stored in Databases Collected and Compared to Newly Produced Hash.

If the two hashes match, it means the document has been verified and hasn't been changed since the first hash was generated and saved. This match demonstrates the document's legitimacy and integrity. If there is any difference between the two hashes, it means the document has been tampered with, corrupted, or altered in any way, and it cannot be validated.

This verification procedure is critical to data security and integrity in a variety of applications, including legal, financial, healthcare, and sensitive document management systems. The
solution assures the integrity of the original data can be rigorously checked even in the face of hardware failures, cyberattacks, or other potential threats by partitioning the hash and distributing it over numerous virtual machines. This strategy lays a solid foundation for trust and reliability in digital transactions and record-keeping, giving users and organizations peace of mind about the legitimacy of their digital assets.

E. Result and Discussion

As shown in Figure 4, our experimental setup includes five virtual machines (VMs), each with the hardware characteristics listed below. The suggested framework was tested on a system equipped with an Intel(R) Xeon(R) CPU E5520 @ 2.27 GHz and 4 GB of RAM. The operating system used is Linux (Debian) software uses the OpenSSL library in Linux. To create a Smart Document a test case has been prepared in which a file is uploaded to a database, after implementing a QR code on the document it is distributed among all nodes and verify the document by comparing the hash of the uploaded file.

The evaluation of the system incorporates Apache Benchmark as shown in Figure 5 and Slowhttptest because of their real-world applicability, versatility, and ability to consistently duplicate results. These tools are well-known for their versatility, allowing us to thoroughly assess both the performance and security elements of our system as mentioned in Figures [6-18] and Tables [1-3]. The intentional selection of these tools not only assures a practical evaluation of our system's capabilities but also contributes to the reproducibility of our research, which aligns with the study's goals and objectives.

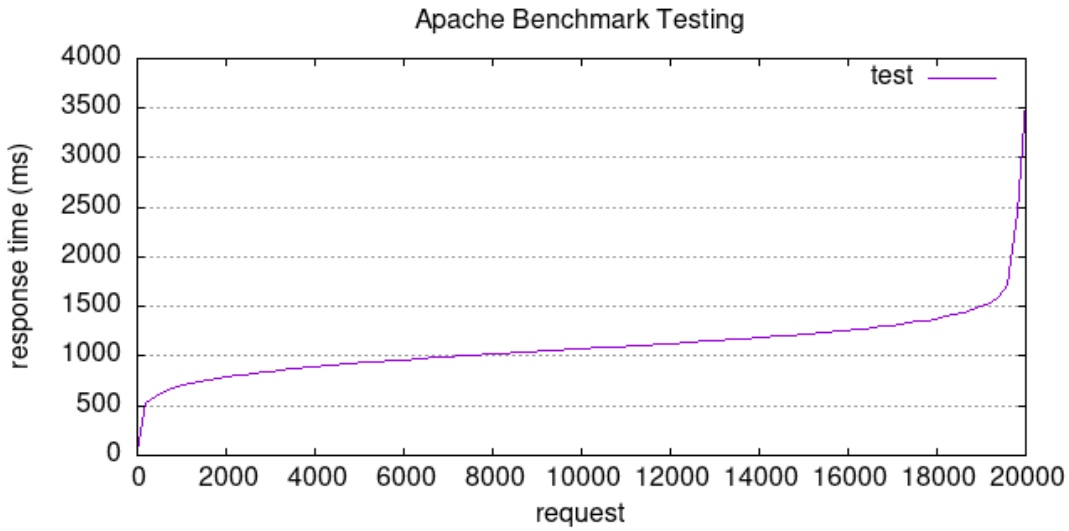

Figure 5. The Use of Apache Benchmark for System Performance testing Using httperf.

Table 1. The Slowhttptest Test Parameters (SLOW READ), Evaluating System's Resilience Against Slow HTTP-Based Assaults.

No.	***Parameters***	***Values***
1.	Test Type	SLOW READ
2.	Number of Connections	20000
3.	Cookie	---
4.	Receive Window Range	512 – 1024
5.	Pipeline Factor	3
5.	Read Rate from Receive Buffer	32 bytes / 5 sec
6.	Connections Per Seconds	200
7.	Timeout for Probe Connections	3
8.	Target Test Duration	240 seconds
9.	Using Proxy	No Proxy

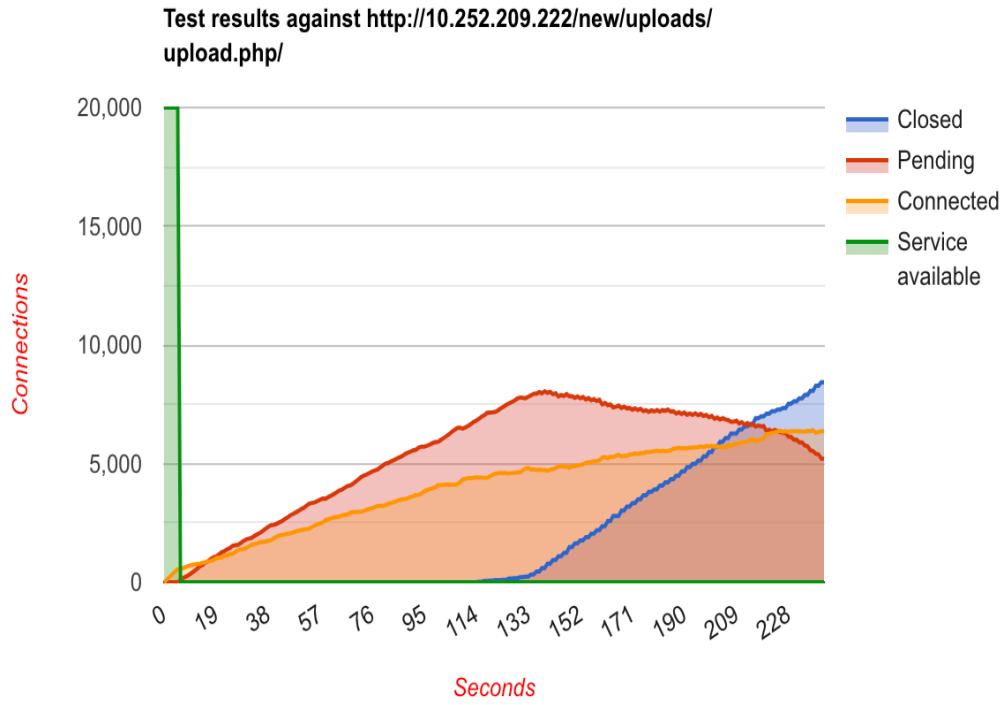

Figure 6. The Graphical Evaluation Findings by Slowhttptest (SLOW READ), Provides A Visual Depiction of The System's Performance in Slow HTTP-Based Attack Scenarios.

Table 2. The Slowhttptest Test Parameters (SLOW HEADERS), Evaluating The System's Resilience Against Slow HTTP-Based Assaults.

No.	***Parameters***	***Values***
1.	Test Type	SLOW HEADERS
2.	Number of Connections	1000
3.	Cookie	---
4.	Content-Length Header Value	4096
5.	Extra Data Max Length	52
5.	Interval Between Follow Up Data	10 seconds
6.	Connections Per Seconds	200
7.	Timeout for Probe Connections	3
8.	Target Test Duration	240 seconds
9.	Using Proxy	No Proxy
10.	Verb	GET

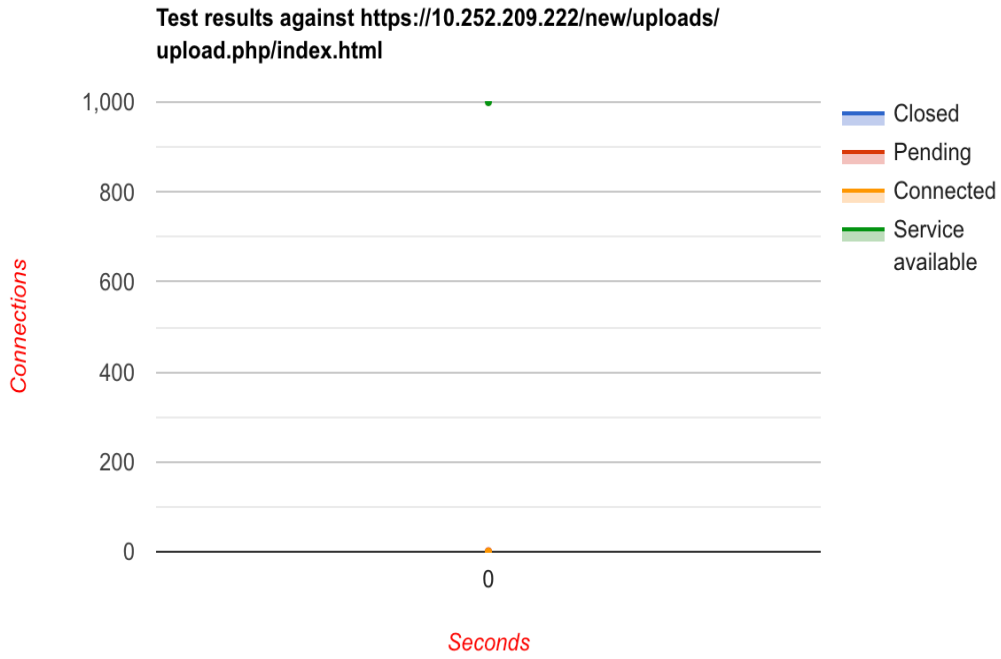

Figure 7. The Graphical Evaluation Findings by Slowhttptest (SLOW HEADERS), Provides A Visual Depiction of The System's Performance in Slow HTTP-Based Attack Scenarios.

Table 3. The Slowhttptest Test Parameters (SLOW BODY), Evaluating The System's Resilience Against Slow HTTP-Based Assaults.

No.	***Parameters***	***Values***
1.	Test Type	SLOW BODY
2.	Number of Connections	3000
3.	Cookie	---
4.	Content-Length Header Value	8192
5.	Extra Data Max Length	22
5.	Interval Between Follow Up Data	110 seconds
6.	Connections Per Seconds	200
7.	Timeout for Probe Connections	3
8.	Target Test Duration	240 seconds
9.	Using Proxy	No Proxy
10.	Verb	FAKEVERB

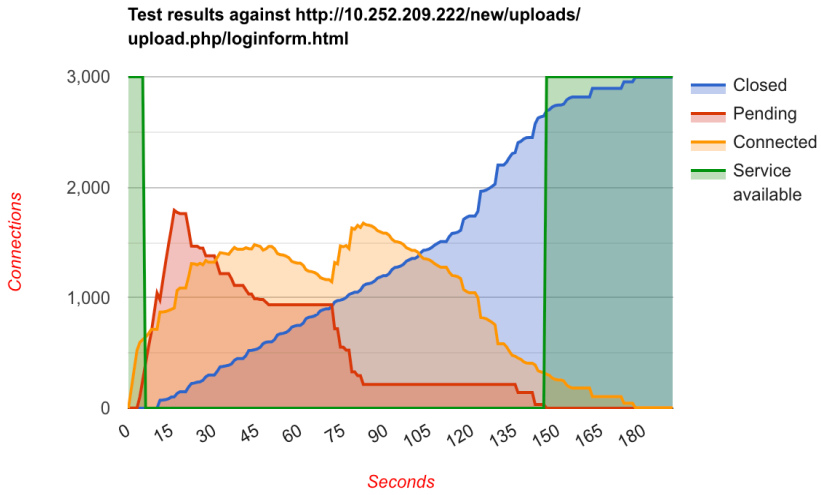

Figure 8. The Graphical Evaluation Findings by Slowhttptest (SLOW BODY), Provides A Visual Depiction of The System's Performance in Slow HTTP-Based Attack Scenarios.

A comprehensive comparison of distributed database and blockchain technologies is provided in Table 4, which is helpful when deciding on system architecture. Blockchain uses consensus methods like Proof of Work and cryptographic concepts to achieve distributed control, immutability, and transparency in a decentralised manner. It performs well in models with strong security and trustlessness but could have scaling issues. Conversely, distributed databases use different consensus techniques like Paxos or Raft and support both centralised and decentralised control. They also permit changes and deletions of data. They are appropriate for traditional applications and centralised systems because they provide faster transactions and scalability. This analysis helps to comprehend the subtle differences between each technology, directing decision-makers according to certain needs, such as effective decentralised applications with high security or centralized system.

Table 4. Comparative Analysis of Blockchain and Distributed Database Features for Decision-Making in System Architecture.

Feature	***Blockchain***	***Distributed Database***
Architecture	Decentralized	Centralized or decentralized
Control and Ownership	Distributed control and ownership	Centralized control and ownership
Consensus Mechanism	Requires consensus algorithms (e.g., Proof of Work, Proof of Stake)	Consensus may vary (e.g., Paxos, Raft)
Immutability	Immutable ledger	Data can be modified or deleted

Feature	***Blockchain***	***Distributed Database***
Transparency	Transparent ledger	Access control and visibility settings
Trust Model	Trust less (trust built on consensus and cryptographic principles)	Trust in the central authority or distributed trust model
Security	Highly secure due to cryptography and consensus mechanisms	Security depends on access controls and encryption
Permissioning	Permissionless or permissioned depending on the blockchain type	Permissioned or permissionless (public or private databases)
Transaction Speed	Varies (can be slower due to consensus mechanisms)	Generally faster due to centralized control
Scalability	Scalability challenges, especially in public blockchains	More scalable depending on the database architecture
Smart Contracts	Supports smart contracts (self-executing contracts with coded terms)	support smart contracts
Data Integrity	High data integrity through cryptographic hashing	Relies on database design and integrity constraints
Fault Tolerance	High fault tolerance due to decentralization	Fault tolerance depends on database replication and distribution strategy
Use Cases	Cryptocurrencies, supply chain, decentralized applications (dApps)	Traditional applications, enterprise solutions, centralized systems
Consistency	Eventual consistency (may take time for all nodes to reach a consistent state)	Immediate consistency in centralized systems
Regulatory Considerations	May face regulatory challenges due to decentralized nature	Easier regulatory compliance in centralized systems
Costs	Can be costly due to consensus mechanisms and decentralization	Potentially lower costs, especially in centralized databases

F. Conclusion

In conclusion, our approach based on QR codes and distributed database shows great promise for document verification. Considerations for improving future system robustness include switching to a more secure database engine, such as PostgreSQL, and addressing fault tolerance issues. The slowhttp test and Apache Benchmark provides useful insights into performance and security. We also urge for a strategy shift to boost resilience, particularly in the interconnected structure of distributed databases. Future advances should concentrate on strengthening security measures and optimising fault tolerance to ensures long-term reliability in document verification procedures and focuses on establish a hybrid system integrating Blockchain with distributed database.

G. Acknowledgement

I would like to express my heartfelt gratitude to Kemitraan Negara Berkembang (KNB) for their important financial and academic assistance, which has played a critical part in aiding my scholastic endeavours. KNB's aid has not only relieved financial problems but has also contributed considerably to the advancement of my academic path. Furthermore, I'd want to convey my heartfelt appreciation to my supervisor, whose unflinching support and experience have been critical in moulding the course of my research. Their insightful critiques and dedicated supervision have broadened my intellectual endeavours and improved the quality of my research output. I recognise the significance of joint efforts in developing academic advancement and research excellence by acknowledging KNB's generosity and my supervisor's impactful mentoring.

H. References

[1] A. Khrykova, M. Bolsunovskaya, S. Shirokova, and A. Novopashenny, 'Implementation of digital signature technology to improve the interaction in company', *E3S Web Conf.*, vol. 244, p. 12023, 2021, doi: 10.1051/e3sconf/202124412023.

[2] W. Zou *et al.*, 'Smart Contract Development: Challenges and Opportunities', *IIEEE Trans. Software Eng.*, vol. 47, no. 10, pp. 2084–2106, Oct. 2021, doi: 10.1109/TSE.2019.2942301.

[3] N. Szabo, 'Formalizing and Securing Relationships on Public Networks', *First Monday*, vol. 2, no. 9, Sep. 1997, doi: 10.5210/fm.v2i9.548.

[4] K. Pal and C. R. S. Kumar, 'QR Code Based Smart Document Implementation Using Blockchain and Digital Signature', in *Data Management, Analytics and Innovation*, vol. 1174, N. Sharma, A. Chakrabarti, V. E. Balas, and J. Martinovic, Eds., in Advances in Intelligent Systems and Computing, vol. 1174, Singapore: Springer Singapore, 2021, pp. 449–465. doi: 10.1007/978-981-15-5616-6_32.

[5] S. Mthethwa, N. Dlamini, and G. Barbour, 'Proposing a Blockchain-based Solution to Verify the Integrity of Hardcopy Documents', in 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), Plaine Magnien: IEEE, Dec. 2018, pp. 1–5. doi: 10.1109/ICONIC.2018.8601200.

[6] G. Sethia, S. Namratha, S. H, and S. C. S, 'Academic Certificate Validation Using Blockchain Technology', in 2022 International Conference on Trends in

Quantum Computing and Emerging Business Technologies (TQCEBT), Pune, India: IEEE, Oct. 2022, pp. 1–5. doi: 10.1109/TQCEBT54229.2022.10041550.

[7] A. Husain, M. Bakhtiari, and A. Zainal, 'Printed Document Integrity Verification Using Barcode', *Jurnal Teknologi*, vol. 70, no. 1, Aug. 2014, doi: 10.11113/jt.v70.2857.

[8] Y. Hassanzadeh-Nazarabadi, A. Kupcu, and O. Ozkasap, 'Decentralized Utility- and Locality-Aware Replication for Heterogeneous DHT-Based P2P Cloud Storage Systems', *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 5, pp. 1183–1193, May 2020, doi: 10.1109/TPDS.2019.2960018.

[9] J. Benet, 'IPFS - Content Addressed, Versioned, P2P File System', 2014, doi: 10.48550/ARXIV.1407.3561.

[10] O. W. Purbo, S. Sriyanto, S. Suhendro, R. Abd. Aziz, and R. Herwanto, 'Benchmark and comparison between hyperledger and MySQL', *TELKOMNIKA*, vol. 18, no. 2, p. 705, Apr. 2020, doi: 10.12928/telkomnika.v18i2.13743.

[11] Y. Gu, D. Hou, X. Wu, J. Tao, and Y. Zhang, 'Decentralized Transaction Mechanism Based on Smart Contract in Distributed Data Storage', *Information*, vol. 9, no. 11, p. 286, Nov. 2018, doi: 10.3390/info9110286.

[12] M. Muzammal, Q. Qu, and B. Nasrulin, 'Renovating blockchain with distributed databases: An open source system', *Future Generation Computer Systems*, vol. 90, pp. 105–117, Jan. 2019, doi: 10.1016/j.future.2018.07.042.

[13] Q. Meng, K. Zhang, H. Pan, M. Yuan, and B. Ma, 'Design and Implementation of Key-Value Database for Ship Virtual Test Platform Based on Distributed System', in *Data Science*, vol. 1879, Z. Yu, Q. Han, H. Wang, B. Guo, X. Zhou, X. Song, and Z. Lu, Eds., in Communications in Computer and Information Science, vol. 1879. , Singapore: Springer Nature Singapore, 2023, pp. 109–123. doi: 10.1007/978-981-99-5968-6_9.

[14] Z. Yang *et al.*, 'OceanBase Paetica: A Hybrid Shared-Nothing/Shared-Everything Database for Supporting Single Machine and Distributed Cluster', *Proc. VLDB Endow.*, vol. 16, no. 12, pp. 3728–3740, Aug. 2023, doi: 10.14778/3611540.3611560.

[15] A. Margara, G. Cugola, N. Felicioni, and S. Cilloni, 'A Model and Survey of Distributed Data-Intensive Systems', *ACM Comput. Surv.*, vol. 56, no. 1, pp. 1–69, Jan. 2024, doi: 10.1145/3604801.

[16] D. Essex, "Blockchain for businesses: The ultimate enterprise guide," June 2021, https://www.techtarget.com/searchcio/Blockchain-for-businesses-The-ultimate-enterprise-guide.