Kingdom of Saudi Arabia
Ministry of Education
Qassim University
College of Computer
Information Technology Department

PRESERVING THE SECURITY OF DIGITAL HADITHS USING A BLOCKCHAIN

A Thesis Submitted to the Department of Information Technology, Qassim University, in Practical Fulfillment of the Requirements for the Degree of Master of Cybersecurity

By

BASHAYER KALIFAH ALKALIFAH

431214115

Supervisor

Dr. Murad A. Rassam

(1446 H. / 2024 AD)


بسم الله الرحمن الرحيم

Kingdom of Saudi Arabia
Ministry of Education
Qassim University
College of Computer
Information Technology Department

PRESERVING THE SECURITY OF DIGITAL HADITHS USING A BLOCKCHAIN

By
Bashayer Kalifah Alkalifah

Recommendation of the Committee:

The Committee has approved this dissertation as a partial completion of the requirement for the master's degree in (Subject).

Examination and Decision-Making Committee

Committee Members	Name	Academic Degree	Specialists' action	Signature
Advisor				
Co-advisor				
External Examiner				
Internal Examiner				
Internal Examiner				

(1446 H. / 2024 AD)

Declaration

I hereby declare that this thesis, entitled: “Preserving the security of digital hadiths using a blockchain” is the result of my original research and that no part of it has been presented for another degree in this university or elsewhere.

Acknowledgments

Primarily, I am profoundly grateful to Allah the Almighty for the endurance, patience, and inspiration He grants us. I wish to convey my appreciation to my supervisor, Dr. Murad A. Rassam, for the invaluable guidance, oversight, recommendations, and support since the commencement of my studies. I extend my heartfelt gratitude to my parents for their financial support and unwavering assistance throughout my master's journey. I extend my profound gratitude to my siblings and other relatives for their support. I express my gratitude to the University of Qassim for affording me the opportunity to pursue a master's degree in a field of profound interest. Finally, I wish to express my gratitude to all my friends for their assistance and counsel throughout this journey.

ملخص الرسالة باللغة العربية

الكلية: كلية الحاسب.

القسم / البرنامج: قسم تقنية المعلومات.

التخصص/ المسار: علوم في الأمن السيبراني.

عنوان الدراسة: الحفاظ على أمن الأحاديث النبوية الرقمية باستخدام البلوكشين.

اسم الطالبة: بشاير خليفه الخليفه

اسم المشرف: د. مراد عبده رسام

الدرجة العلمية: ماجستير

تاريخ المناقشة أو المنح:

الكلمات الاستدلالية: تقنية البلوك تشين؛ الحديث الرقمي؛ نظام الملفات بين الكواكب؛ البلوك تشين المسموح به؛ أمن البيانات؛ هايبرلدجر فابريك؛ البيانات الرقمية.

الملخص

في العصر الحديث، أصبح الوصول إلى الأحاديث الرقمية عبر الإنترنت متاحًا على نطاق واسع للمسلمين في جميع أنحاء العالم. ومع ذلك، تكمن المشكلة الرئيسية في حماية هذه الأحاديث. إن انتشار الأحاديث الملفقة على المواقع الإلكترونية ووسائل التواصل الاجتماعي يجعل المسلم العادي في موقف صعب في التمييز بينها وبين الأحاديث الصحيحة. وعلى الرغم من أن بعض المواقع الإلكترونية تتيح للمستخدمين التحقق من صحة الأحاديث، إلا أنها تعتمد على أنظمة معلومات مركزية، مما يجعلها عرضة للتلاعب بالبيانات والتلف. يقترح هذا البحث نموذجًا يعتمد على البلوك تشين هايبرلدجر فابريك، وهي المنصة الرائدة لبلوك تشين الكونسورتيوم مع خوارزمية إجماع Raft، لتوفير وسيلة آمنة لتخزين الأحاديث الرقمية. علاوة على ذلك، يستخدم حلنا آليات التحكم في الوصول القائمة على الأدوار لمنع التعديلات غير المصرح بها. يعمل النموذج من خلال خطوات رئيسية: (1) تسجيل المستخدمين، مع قيام مؤسسات الحديث بتحديد من يتم قبولهم. (2) السماح لطلاب الحديث بتحميل أحاديث جديدة وطلب التحقق منها من قبل العلماء. (3) السماح لعلماء الحديث بمراجعة الحديث، ويتطلب كل حديث موافقة من عالمين من مؤسستين مختلفتين لمنع التحكم المركزي. (4) تخزين بيانات الحديث الأساسية فقط على البلوك تشين، مع تخزين السند الكامل وشرح الحديث على نظام الملفات الموزع لتحسين قابلية التوسع. بعبارة أخرى، يخزن هايبرلدجر فابريك فقط قيمة تجزئة السند وشرح الحديث. بالإضافة إلى ذلك، يمكن لأي عالم على الشبكة تقديم طلب لتحديث حديث إذا تم العثور على خطأ. يمكن للمستخدمين عرض كل من الإصدارات الأصلية والمصححة للحديث والعلماء الذين وافقوا على كل نسخة. توضح النتائج التجريبية باستخدام أداة Hyperledger Caliper قابلية توسع النموذج، حيث يحافظ على معدل إنتاجية مرتفع وزمن استجابة مقبول مع زيادة المؤسسات والعقد وتقديم الأحاديث. على سبيل المثال، مع زيادة معدل إضافة الأحاديث، ارتفع معدل الإنتاجية من ٤٨.٥ معاملة في الثانية إلى ٢٣٤.٥ معاملة في الثانية، مع زمن استجابة يتراوح بين ١.٢٣ و١.٢٤ ثانية. إن تنفيذ هذا النموذج له آثار مهمة على المجتمع الإسلامي، لأنه يعزز سلامة الأحاديث الرقمية وشفافيتها وإمكانية تتبعها وتوافرها مع تخفيف المخاطر المرتبطة بأنظمة التحقق المركزية والأحاديث المكذوبة.

English Abstract

College: Computer College.

Department/ program: IT Department.

Specialization/ track: Cyber Security.

Title of the Thesis: Preserving the Security of Digital Hadiths Using a Blockchain.

Student's Name: Bashayer Kalifah Alkalifah.

Supervisor's Name: Dr. Murad A. Rassam.

Degree: Master.

Discussion or date of granting:

Keywords: Blockchain Technology; Digital Hadith; IPFS; Permissioned Blockchain; Data Security; Hyperledger Fabric; Digital Data

Abstract

In modern times, the widespread availability of digital hadiths on the internet has made them accessible to Muslims worldwide. The biggest problem with readily available digital hadiths is their protection. The spread of fabricated hadiths on websites and social media leaves the typical Muslim at a disadvantage in differentiating between them and authentic ones. While some websites allow users to verify the authenticity of Hadiths, they rely on a centralized information system, making them vulnerable to data tampering and damage. This research proposes a model utilizing the Hyperledger Fabric blockchain, the predominant platform for consortium blockchains with the Raft consensus algorithm, to provide a secure way to store digital hadiths. Moreover, our solution employs Role-based access control mechanisms to prevent unauthorized modifications. The model operates through key steps: (1) registering users, with hadith institutions deciding who is accepted. (2) Allowing hadith students to upload new hadiths and requesting scholars verify them. (3) Allowing hadith scholars to review hadith, and each hadith requires approval from two scholars from different institutions to prevent centralized control. (4) Storing only the essential hadith data on the blockchain ledger, with the full Sanad and commentary stored on the interplanetary file system (IPFS) to improve scalability. In other words, Hyperledger Fabric only stores the Sanad and Hadith commentary's hash value. Additionally, any scholar on the network can submit a request to update a hadith if an error is found. Users can view both the original and corrected versions of a hadith and the scholars who have

approved each version. Experimental results using the Hyperledger Caliper tool demonstrate the model's scalability, as it maintains high throughput and acceptable latency as institutes, nodes, and hadith submissions increase. For example, as the rate of hadith additions increased, throughput rose from 48.5 TPS to 234.5 TPS, with a 1.23–1.24 second latency. Implementing this model has important implications for the Muslim community, as it enhances the integrity, transparency, traceability, and availability of digital hadiths while mitigating the risks associated with centralized verification systems and fabricated hadiths.

Table of Contents

List of Figures

List of Tables

1 Chapter One

1.1 Introduction

The growth of information and communication technology has changed every aspect of daily life. This includes scanning the Holy Quran and Hadith from physical copies to digital format and making them online[1]. Muslims regard the Quran as the divine words of God, which mandate adherence to the instructions of the Prophet Muhammad in their legislation, laws, and moral guidance. This explicit directive to imitate the Prophet and adhere to his rulings is essential, as not all regulations and Islamic laws are delineated in the Quran [2]. The prayer, including the Adhaan (calling to prayer), is derived from the documented actions of the Prophet. The Quran mandates prayers as an obligation, yet it lacks specific details regarding their execution, indicating that most Islamic practices are derived from the Prophet [3]. Consequently, documenting the Prophet's sayings and deeds is of significant importance. Hadith encompasses the sayings, practices, and teachings of Prophet Muhammad (PBUH) as well as his implicit endorsement or disapproval of the actions of others.

The authenticity of digital Hadith formats on the internet has raised concerns. The potential for tampering with Hadith's text has the potential to mislead Muslims and cause them to engage in actions that may be harmful or even dangerous. For instance, if someone were to fabricate a Hadith related to health, it could lead Muslims to consume a harmful herb and cause adverse effects on their health [1]. Preventing people from spreading non-authentic Hadiths on the internet is not feasible, making it a complex and challenging issue to solve [4].

Some websites provide tools for hadith verification. However, these methods are plagued by a single point of failure (SPOF). A centralized design that manages operations, controls, and storage creates a vulnerability that allows for an SPOF. This suggests the entire system could stop working if technological issues or disturbances threaten website availability [5]. Insider attacks can also delete or edit hadiths.

Blockchain offers a promising solution to the issue of a centralized Hadith verification system. By securely storing Hadiths in a decentralized way, blockchain technology functions as a distributed ledger, methodically storing digital records of transactions across multiple interconnected computers. Figure 1.1 illustrates the distinction between centralized types and blockchain architecture. Blockchain technology incorporates many

cutting-edge technologies, such as peer-to-peer networks, smart contracts, consensus algorithms, signatures, hash functions, and encryption, to offer several functionalities, including availability, integrity, transparency, and traceability [6]. Because of its unique characteristics, blockchain can substantially augment the security of Hadith storage systems.

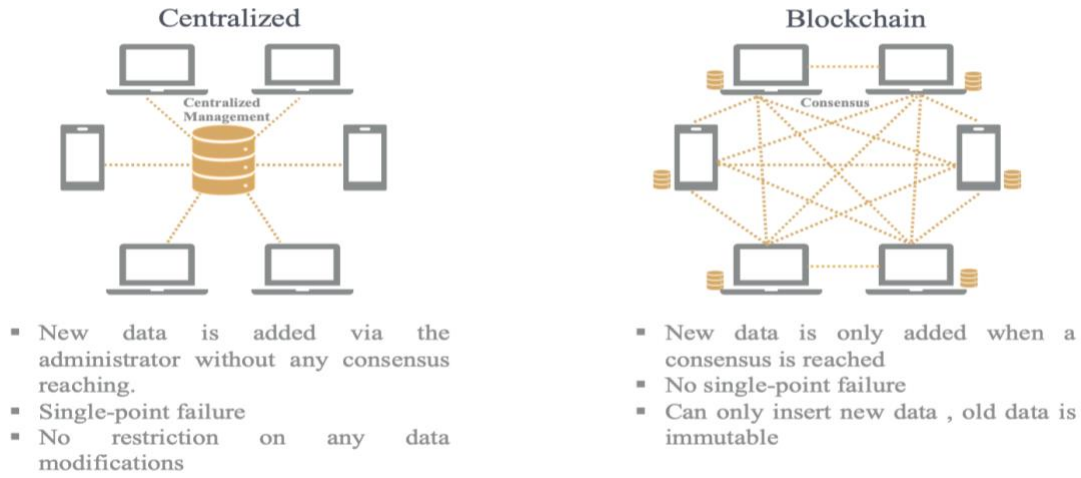

Figure 1.1 Comparison of Centralized and Blockchain Architecture

However, the existing scalability issues undeniably restrict the potential of blockchain technologies. A research study recently examined the use of blockchain and IPFS for many purposes, such as supply chain and healthcare [7],[8]. IPFS is a distributed and secure peer-to-peer network explicitly created to share files. Adding IPFS as a storage solution outside of the blockchain can improve scalability[9],[10],[11], making it a more suitable solution for storing hadiths.

1.2 Problem Background

Recently, blockchain has garnered attention as a potential solution to digital data management issues. Several studies have used blockchain technology to preserve health records, academic certificates, digital forensic evidence, the digital Quran, and the digital hadith.

In healthcare, blockchain is considered a secure way to store patient data and a tool for communication between healthcare providers and patients. For example, a study in [9]

established a framework utilizing Hyperledger Fabric blockchain and IPFS to guarantee the security and scalability of electronic health records. To increase security, even the information stored in IPFS was encrypted utilizing the public key encryption method. The results demonstrated that the encrypted hash prevented unauthorized parties from tracing the health data. Overall, this framework successfully implemented data security, privacy, scalability, and interoperability aspects.

Similarly, academia has discussed blockchain regarding the authenticity of certificates, their storage, and possible forgery. For example, researchers in [12] proposed a hyperledger fabric blockchain to store educational certificates in Pakistan. Their approach enabled universities and higher education commissions to easily and quickly obtain authenticated certificates. The results confirmed that the Hyperledger Fabric solution provided record security.

In digital forensics, blockchain technology has been explored to securely store digital evidence and improve its transparency. For example, the authors in [13] presented a blockchain solution called BCFL based on the Hyperledger Fabric. The presented approach aimed to improve the reliability and credibility of digital forensic evidence stored in the cloud environment. The authors wanted to address the problems of obtaining valid electronic evidence from the cloud, which could be altered during transmission or storage. The BCFL successfully preserved the evidence's content and avoided the risk of its deletion.

Blockchain was important for electronic voting because it improved the process's security and efficiency. For instance, a study [14] presented an electronic voting system that employed a blockchain solution using Hyperledger Fabric. The system aimed to avoid the vulnerabilities of centralized electronic voting systems, including altering the database and voting twice. It allowed the voter to remain anonymous throughout the process. The system also ensured the reliability of votes through smart contracts.

Blockchain can also be utilized to guarantee that the digital Quran is secured to avoid any form of tampering and guarantee the authenticity of the content. For instance, in [15], the authors developed an Ethereum blockchain-based system to protect the digital Quran and verify its authenticity. In their system, the Religious Affairs Authority used the SHA-256 method to create a hash for each authentic Quran document and stored it on the blockchain

to verify it in the future. Therefore, individuals could easily ensure the genuineness of a Quran digital document by comparing the hash value of the chosen Quran document with the hash value stored on the blockchain.

In [16], a study proposed a permissionless blockchain system to ensure the integrity of digital Quran texts without relying on third-party trust through Ethereum. The system comprises two active members: the Quran users, who utilize the application to confirm the authenticity of a Quranic verse through a consensus mechanism named Proof-of-Stake (PoS). Additionally, Quran miners publish the digital Quran in the application using a consensus algorithm named Proof-of-Work (PoW). Therefore, the framework facilitates the protection of the Quranic text and provides a way to confirm its authenticity.

Despite the extensive research on blockchain implementations to preserve various types of digital data, only one published study has proposed to protect the security of digital Hadith using blockchain technology [3]. The implementation used the Hyperledger Fabric platform with a Practical Byzantine Fault Tolerance (PBFT) consensus algorithm. The study leveraged the immutability and transparency of the blockchain to protect hadiths from unauthorized modification. This framework allowed only specialized institutions in Islamic countries to insert new Hadiths into the blockchain. However, this framework has several limitations. These include issues of scalability, performance, and access control. The system architecture must also consider the updating of hadiths and the scholars' agreement on the authenticity of the hadiths. The following section will elaborate on these issues.

1.3 Problem Statement

The internet has revolutionized how people access and share information, enabling easy access to massive Hadith collections. However, this also makes it easy to share non-authentic hadiths with a large audience. Additionally, online social networking sites have become breeding grounds for spreading non-authentic hadiths, which can spread rapidly to millions. The internet has also enabled individuals with limited knowledge or expertise to present themselves as specialists in Islamic matters, leading to the unintentional or intentional spread of non-authentic hadiths. Furthermore, the internet provides anonymity, allowing individuals to generate and disseminate content while keeping their identities

hidden, allowing anyone with bad intentions to fabricate hadiths without direct accountability. Therefore, the issue with digital hadiths is complex, with three leading causes. The first is the ease of fabricating hadiths and the inability to stop their spread. Second, because most Hadith verification systems are centralized, they are vulnerable to internal and external threats, including SPOF and data tampering. Thirdly, academic-based research on storing digital hadith using blockchain technology is scarce. To date, only one study has been identified in this regard. The study [3] needs to consider essential requirements like scalability and ignore access control. The absence of access control increases the risk of unauthorized modifications and reduces system security. Some flaws exist in the PBFT consensus, as it necessitates substantial traffic for practical implementation [17],[18]. One major limitation of their framework was the absence of scholar approval for a hadith, which posed a risk of adding non-authentic hadiths. A dual approval system, requiring validation by two independent scholars, prevents any single individual from gaining control. To speed up the process, the system should also allow hadith students to upload new hadiths and request confirmation from scholars. Moreover, the system must submit update requests to correct any errors.

1.4 Research Aim

This research aims to develop a secure model for storing digital Hadith by leveraging the Hyperledger Fabric blockchain integrated with IPFS, ensuring data integrity, transparency, traceability, scalability, and availability.

1.5 Research Questions

The following research questions should be answered upon the completion of this study:

1. How can the Hyperledger Fabric blockchain ensure digital hadiths' integrity, availability, transparency, and traceability?
2. How can the proposed model use the IPFS to improve scalability?
3. How can role-based access control (RBAC) using Hyperledger Fabric technology be implemented?

1.6 Research Objectives

To achieve the aim of this research, several objectives are set as follows:

1. To design and implement a secure hadith storage model using Hyperledger fabric blockchain with Raft consensus algorithm to ensure digital hadith data's integrity, availability, transparency, and traceability.
2. To explore how IPFS technology can be used as a decentralized storage solution for off-chain data storage and integrate it with the Hyperledger Fabric to reinforce the scalability of the system
3. To implement RBAC mechanisms to prevent unauthorized modifications.
4. To evaluate the proposed model by measuring its performance using metrics such as throughput and latency.

1.7 Research Scope

1. This research adopts the Hyperledger Fabric blockchain.
2. This research considers the IPFS technique as the off-chain data storage method.
3. The research is limited to the storage and retrieval of hadiths in Arabic.
4. This research focuses on hadith data integrity, transparency, traceability, scalability, and availability.

1.8 Research Significances

The proposed model overcomes centralized data management challenges by providing a secure system for managing and preserving hadiths. Consequently, the proposed model possesses the capacity to:

1. Make sure to maintain hadiths for future generations.
2. It preserves the integrity of the Hadith through immutable storage and decentralized management. This protects against fabricated hadiths.
3. It offers decentralized management, which lessens reliance on central authorities and enhances system availability.
4. Improves the transparency and traceability of hadiths, guaranteeing accountability

in the documentation process.

5. Makes sure only Hadith scholars can authenticate and approve Hadiths.
6. Protection against Bid'ah (Innovation). In Islam, Bid'ah refers to introducing practices not rooted in the Hadiths or the Quran.
7. Facilitate the uniformity of Hadith documentation procedures among different institutions.
8. This research demonstrates the applicability of blockchain technology in fields requiring information authentication, such as managing religious opinions (Fatwa).

1.9 Definitions of Terms

Table 1.1 defines the terms adopted for this research's objectives.

Table 1.1 Definitions of Terms.

Term	Definition
Security	The safeguarding of digital data from unauthorized access, alteration, and destruction through the careful implementation of security attributes including confidentiality, integrity, and availability.
Integrity	A security principle that ensures digital data has not been altered or modified without authorization.
Availability	A security principle that guarantees the accessibility and availability of digital data when required.
Scalability	The capacity of a system or framework to manage increasing network sizes or transaction volumes without a decline in performance.
Transparency	Transparency means that all data, including every transaction and modification, is visible to all participants in the system.
Traceability	The ability to track and verify the history, origin, and alterations of data or transactions within a system.
Access Control	Access control is a computer security mechanism that determines the eligibility of individuals to access system resources.

1.10 Thesis Organization

Chapter 1: Introduction

This chapter offers a comprehensive introduction to the research context, delineating the problem the study seeks to resolve. It clearly articulates the research aim and objectives, along with the research questions that will direct the inquiry. This chapter emphasizes the significance of the research.

Chapter 2: Literature Review

This chapter presents an overview of Hadith in Islam and comprehensively analyzes the relevant theories and concepts necessary for understanding blockchain technology. It compares different types of blockchains and explains the rationale for selecting a permissioned blockchain for the proposed model. Furthermore, it reviews the work of researchers across various blockchain fields, highlighting the shortcomings of current solutions.

Chapter 3: Methodology

This chapter delineates the proposed model in detail, offering a thorough overview of its five phases: registration, hadith verification request, verification by hadith scholars, storage, and performance evaluation. The model employs blockchain technology, particularly Hyperledger Fabric, in conjunction with decentralized storage via IPFS and Raft consensus.

Chapter 4: Model Design

This chapter examines the design of a structure for the proposed model. This chapter examines the Hyperledger Fabric technology in detail, focusing on its components, the design of the fabric network, and the transaction flow. It discusses IPFS technology in more detail. The detailed analysis of these technologies provides a solid foundation for understanding the implementation of the system.

Chapter 5: Implementation

This chapter covers the main components, starting with setting up the Hyperledger Fabric network and developing the chain code. It then shows how system users perform various operations. The results of the experiments showed that the network scales efficiently. The chapter also discusses how the model meets the requirements

of decentralization, access control, data integrity, availability, transparency, traceability, and scalability. Finally, the proposed model is compared with existing solutions to demonstrate its unique advantages.

Chapter 6: Conclusion

This chapter summarizes the research, emphasizing its contribution to the discipline and offering future directions.

2 Chapter Two

Literature Review

2.1 Introduction

This chapter encompasses the theoretical background and related works pertinent to this research. It begins by providing the background for the reader to understand Hadith in Islam. Additionally, it discusses the challenges in centralized Hadith systems. Then, it covers the fundamental concept of blockchain, including its history and components. Additionally, it will examine the various categories of blockchain and clarify the rationale for selecting a permissioned blockchain for the proposed model. It will also address Hyperledger Fabric, a form of permissioned blockchain. Moreover, it will provide an overview of IPFS. Finally, the chapter reviews the studies most relevant to the current research.

2.2 Hadith

Muslims regard the Quran as God's divine words, which mandate adherence to the instructions of the Prophet Muhammad in their legislation, laws, and moral guidance. This is apparent in the Quran, as demonstrated in the following verse.

> ﴿قُلْ أَطِيعُوا اللَّهَ وَأَطِيعُوا الرَّسُولَ فَإِن تَوَلَّوْا فَإِنَّمَا عَلَيْهِ مَا حُمِّلَ وَعَلَيْكُم مَّا حُمِّلْتُمْ وَإِن تُطِيعُوهُ تَهْتَدُوا وَمَا عَلَى الرَّسُولِ إِلَّا الْبَلَاغُ الْمُبِينُ﴾
>
> Say, "Obey Allah and obey the Messenger; but if you turn away – then upon him is only that [duty] with which he has been charged, and upon you is that with which you have been charged. And if you obey him, you will be [rightly] guided. And there is not upon the Messenger except the [responsibility for] clear notification."
>
> -The Quran [Verse 54 from Surah An-Nur]

This explicit directive to imitate the Prophet and adhere to his rulings is essential, as not all regulations and Islamic laws are delineated in the Quran. The prayer, including the Adhaan (calling to prayer), is derived from the documented actions of the Prophet. The Quran mandates prayers as an obligation, yet it lacks specific details regarding their execution, indicating that most Islamic practices are derived from the Prophet. Consequently, documenting the Prophet's sayings and deeds is of significant importance.

Hadith, an Arabic term meaning “report,” “speech,” or “narrative,” refers to the documentation of various aspects of the Prophet’s life. The types of Hadith vary. The content may consist of a brief sentence or an extensive paragraph detailing a specific incident involving the Prophet, a dialogue from the Prophet’s conversation with an individual, or a narrative recounted by the Prophet’s companions that elucidates the Prophet’s actions regarding a particular issue, such as prayers.

Unlike the Quran, Hadith was not written immediately after the Prophet’s death. Instead, scholars transmitted it orally through generations, each citing the individual from whom they received the Hadith. Nonetheless, specific unscrupulous individuals intentionally concocted content and attributed it to the Prophet. This resulted in the establishment of Hadith science, wherein scholars examine the chain of narrators to determine whether Hadith is accepted or rejected. This process established Hadith’s distinctive structure [19].

2.2.1 Hadith Structure

Every hadith consists of two components: Sanad and Matn. “Matn” refers to the actual content, body of hadith, or report after the Sanad. Sanad is the chain of narrators or persons who conveyed a hadith from the Prophet to us [20]. Sanad means support and refers to the process of determining the authenticity of Hadith through the narrator’s examination. Figure 2.1 shows an illustration of a hadith from the most reliable hadith book, “Sahih of Bukhari.” Narrator Umar ibn al-Khattab is a comrade of Prophet Muhammad who personally heard the Prophet make the statement. The first narrator was a more recent person who passed away in 219, Al-Humaidi Abdullah bin al-Zubayr. We included the death dates of every narrator in Hijri (the Islamic lunar calendar that starts with Prophet Muhammad’s journey to Madinah in 622 CE) to show how the narrators passed down the Hadith over time.

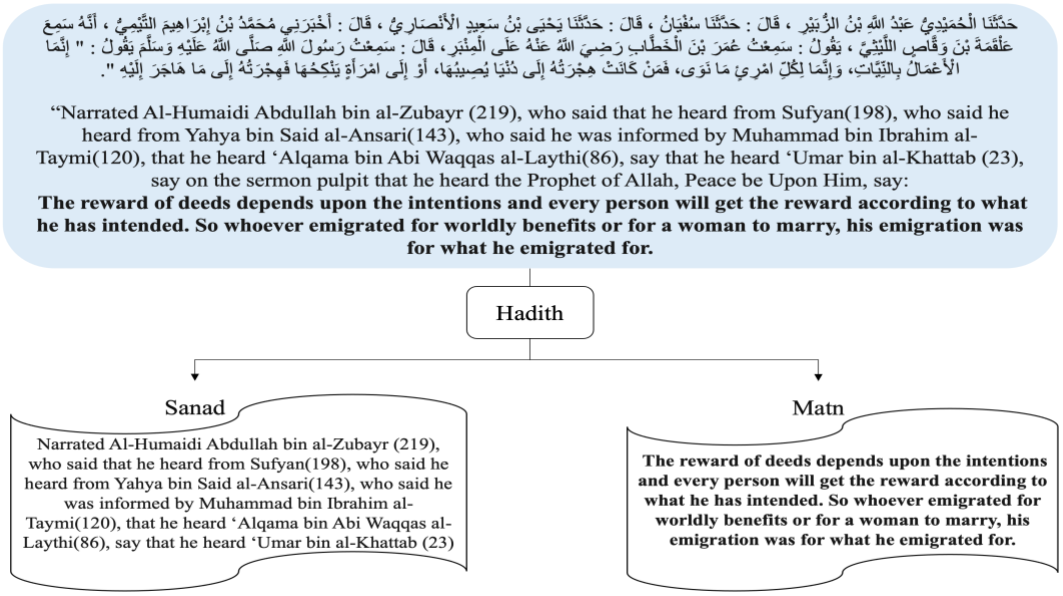

Figure 2.1 Example Hadith From "Sahih of Bukhari"

2.2.2 Hadith Authenticity

Throughout Islam's history, scholars have acknowledged the existence of non-authentic hadiths. Given the significance of hadiths in Islam, it was imperative to ensure their authenticity to shield people from being misled by claims incorrectly attributed to the Prophet Muhammad. Thus, hadith science came into being. The focus of hadith science is the assessment of hadith authenticity. Scholars carefully study the Sanad and Matn [21] to verify a particular hadith's authenticity.

In assessing the Sanad, scholars look closely at every narrator to ascertain their reliability, accuracy of memorization, and interconnectedness within the Sanad. A fundamental prerequisite for reliability is that the narrator be a devout Muslim who observes their religious duties and refrains from doing prohibited things. Narrators who have been stated to have lied are seen as unreliable. In addition, hadith narrators must have accurate and precise memories to prevent misrepresenting the sayings of the Prophet Muhammad (PBUH). Moreover, scholars emphasize how important it is for narrators to be interconnected with one another within the chain. This entails determining if each narrator

acquired the hadith directly from their prior narrator. This criterion must be met in every Sanad of hadith from the first to the final narrator [22].

Scholars of Hadith additionally investigate Hadiths' Matn to see whether or not they are consistent with the Arabic grammar, authentic Hadiths, or the content of the Quran due to the potential for the use of words or expressions that are inappropriate or inconsistent with the beliefs of Muslims, authentic Hadiths, or the teachings of the Prophet [4].

Based on the assessment of both the Matn and Sanad, a verdict is made on the authenticity of a hadith, leading to categorizations like Maudu (fabricated), Daif (weak), Hasan (good), and Sahih (authentic). The hadiths of Maudu are fabricated and purposefully misrepresented. Daif's hadiths are weak because of unreliable narrators, broken chains, or problems with the Matn. While Hasan hadiths are acceptable and good but not confirmed as Sahih, Sahih hadiths are regarded as highly reliable [21]. Hadiths categorized as Sahih and Hasan are considered authentic, while Daif and Maudu are considered non-authentic.

Among the most renowned books that examined hadith are Sahih Muslim, Sahih Al-Bukhari, Sunan Abu Daoud, Sunan Ibn Majah, Sunan Al Tirmidhi, and Sunan Al Nasai [23]. The following subsection provides an overview of the various types of Hadith books.

2.2.3 Hadith Books

Not all hadiths are deemed authentic. Consequently, early Islamic scholars recognized the necessity of compiling authentic hadiths for future generations. The Islamic scholar Muhammad Albukhari, who passed away in 870 C.E., conducted groundbreaking research. He compiled hadiths that adhered to the highest standards of authenticity into a volume widely recognized as Sahih Al-Bukhari. The Arabic term 'Sahih' signifies authenticity and correctness.

Following Albukhari's efforts, additional Islamic scholars produced further Sahih books, notably Muslim ibn al-Hajjaj, a disciple of Albukhari, who authored a Hadith compilation titled 'Sahih Muslim.' Six acknowledged Sahih books are collectively known as "Al-Sihah al-Sittah", meaning "The Authentic Six". These books are Sahih Muslim, Sahih Al-Bukhari, Sunan Abu Dawood, Sunan Ibn Majah, Sunan Al-Nasai, and Sunan Al-Tirmidhi.

They constitute the foundations for Islamic Hadith books in general. They are known as the authentic six because most of the hadiths in these books are authentic. The name "authentic six" originates from the predominance of authentic Hadiths within these books, even though not all incorporated Hadiths exhibit the same level of authenticity [19].

2.2.4 Hadith Commentaries

Hadith commentary books are where scholars with extensive expertise document their insights on Hadith for the general public. Commentators dedicate their lives to interpreting and elucidating Hadiths by utilizing their knowledge of Quranic commentary (Tafsir), the Quran, rhetoric, and Arabic grammar. The extensive number of commentaries stems from developing various explanatory types for the same collection of Hadiths. For example, multiple scholars have written fifty-six commentaries on Sahih Al-Bukhari. Fath Al-Bari by Ibn Hajar al-Asqalani is one of the most famous works [24].

2.3 Existing Problems for Centralized Hadith Systems

Centralized hadith verification systems are technological solutions for verifying the authenticity of hadiths. People can use applications and websites such as dorar.net and hdith.com to recognize the difference between authentic and non-authentic hadiths. This system requires a centralized server for data processing, network request services, and storage. Centralized architectures have several advantages, such as improved device connectivity, user authorization and identification, and efficient networks.

However, the centralized architecture also has some weaknesses. A central server that manages all operations, controls, and storage can lead to an SPOF. This means that if the central unit fails, the entire system becomes inaccessible [25]. To mitigate this issue, the most common method involves incorporating redundant switches, network connections, and servers to serve as backups that operate in succession when the primary centralized server malfunctions. However, this approach comes with numerous challenges, including the high cost of implementing alternative requirements and synchronization issues between the backup and original servers.

Furthermore, data in a centralized architecture is vulnerable to manipulation by insider attacks that compromise data integrity. Insider attacks include deleting, inserting, and substituting information. Deletion can remove lines or characters from a text, changing the sentence's overall meaning. By inserting content, the attacker can alter the essence of the original text. Substituting essential words with other terms alters the original text [26]. Centralized systems also lack transparency and traceability, which limits independent verification and accountability.

An alternative to centralized systems is blockchain, a distributed system in which multiple servers or nodes execute tasks and store data. In a decentralized system, the failure of one node does not hinder the system, as the remaining nodes assume the responsibilities of the failed node. In contrast to a centralized architecture, this decentralized structure eliminates SPOF, provides data integrity, transparency, and traceability, and ensures availability [27]. Subsequent sections will describe blockchain technology and explain its components, concepts, and historical background.

2.4 Blockchain

Blockchain is an immutable, distributed, decentralized ledger used in a peer-to-peer network to securely record transactions throughout several computers without the involvement of third parties[28]. While commonly connected with cryptocurrencies, blockchain technology has significantly influenced a wide range of other distributed application sectors. Blockchain eliminates the need for a central authority. It is a trustworthy technology because of features like the propagation of data storage across independent nodes and consensus algorithms that provide immutability (unchangeability) and transparency. Such technology protects blockchain-based systems from manipulating transaction data while offering many other features and addressing numerous system problems [29]. Any user that participates in a blockchain network may be considered a node. The subsection below explains the concept and history of blockchain.

2.4.1 Blockchain History and Concept

Blockchain is a promising and revolutionary technology since it helps to lower security concerns, increase transparency, and eliminate fraud to a level never previously observed [30]. As Chen et al. [31] stated, blockchain technology, sometimes called "the Internet of Value Exchange," is being considered the next industrial revolution after the development of electricity, steam engines, and information technology. According to Gartner, the corporate value addition of blockchain technology is expected to increase from a little over $360 billion by 2026 to over $3.1 trillion by 2030 [32]. While the initial concept for blockchain originated in 1991, it wasn't until 2008[33] that it began to gain popularity. At that time, a person or group with the pseudo name of Satoshi Nakamoto published a white paper on Bitcoin, a system for peer-to-peer electronic money that built upon the research of Stuart Haber and W. Stuart Haber on a chain of blocks protected by cryptography [34]. Haber's work was restricted to the generation of tamper-proof documents by connecting them in a chain, wherein altering a single document would change the chain as a whole. Satoshi Nakamoto expanded the system to preserve transaction histories and fully decentralized the system by including anonymous users and independent miners. Since the entire system relies on hash encoding with complex computational issues, the best possible level of security is guaranteed.

The blockchain is a primary yet effective method of receiving and sending data securely and automatically. As seen in Figure 2.2, a transaction begins when one of the participants creates a block. The confirmation of this block is coming from thousands of computers dispersed throughout the internet. The confirmed block is linked in a chain, which is, as a result, stored over the internet, creating a unique record with a remarkable history [35]. Once a block is recorded, it cannot be removed without additionally re-moving all blocks that come after it, making transactions on the blockchain irreversible [36].

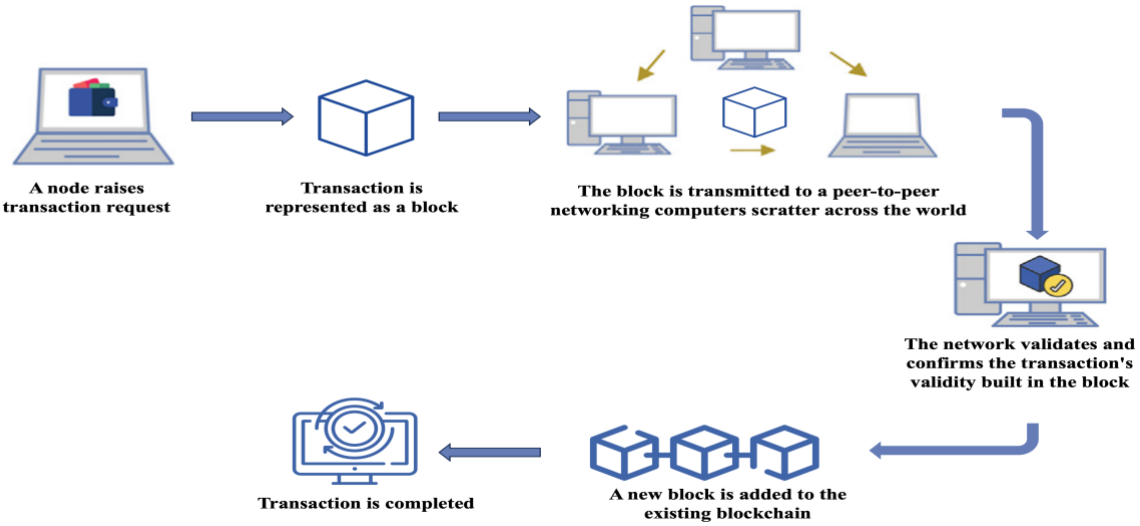

Figure 2.2 A Basic Blockchain Transaction

2.4.2 Blockchain Component

Several components form a blockchain platform, including blocks, distributed ledgers, peer-to-peer networks, smart contracts, and consensus mechanisms, as illustrated in Figure 2.3. Each element is elucidated as follows:

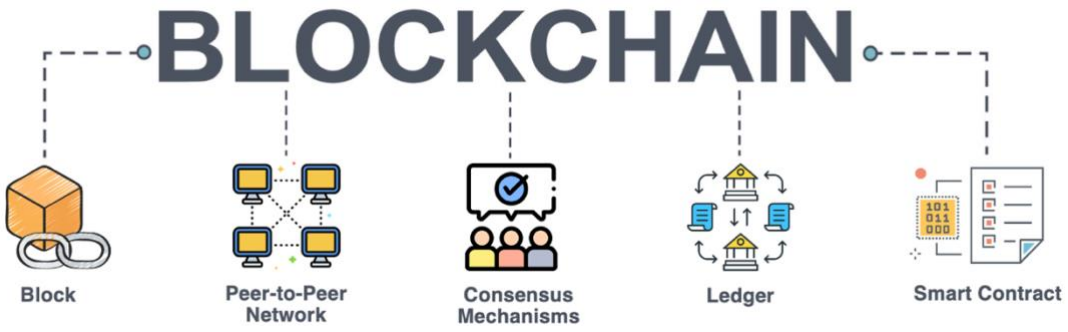

Figure 2.3 Blockchain Core Components

1. **Blocks**

Blockchain technology employs blocks to structure information or transactions, forming a chain of interlinked blocks. A hash function links the chains by producing distinct signatures for every block. Alterations to a block's data can modify the entire signature, enabling identification. The current blockchain appends new consensus blocks [37]. Each

blockchain platform can define its data fields for implementation. Blocks generally consist of a header and a body, as illustrated in Figure 2.4.

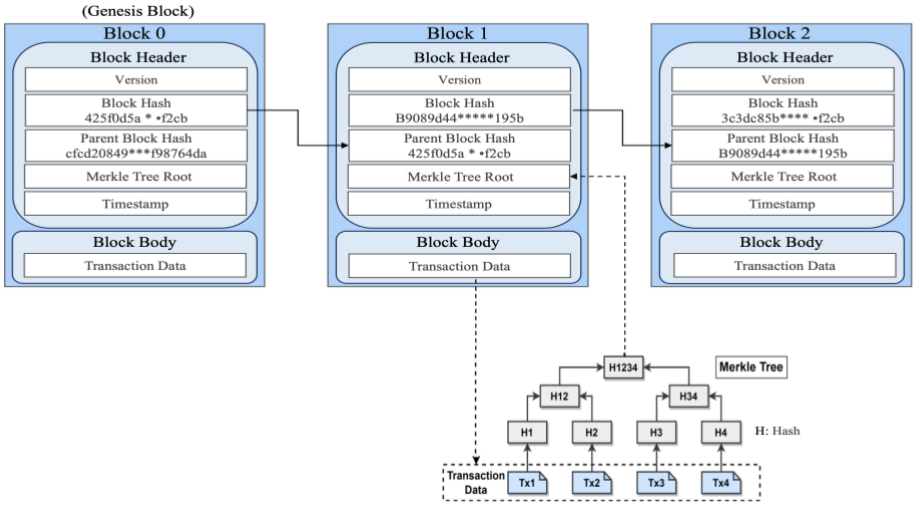

Figure 2.4 Structure of a Block in Blockchain

The block header generally consists of the following components [38]:

A. **Block number:** utilizes integers commencing from 0, designated as the genesis block, with increments of the number by 1 per subsequent block.

B. **Version:** utilized to monitor protocol/software upgrades.

C. **Block hash**: denotes a hash of the block itself. Hash functions encrypt any input text of any length into a fixed-length output.

D. **Parent block hash**: links the current block to the previous one to maintain blockchain integrity.

E. **Merkle tree root:** Transactions hash values are compressed into a Merkle tree, a type of hash binary tree, to verify block-wide transaction integrity.

F. **Timestamp:** denotes the publication time of the transaction, expressed in seconds of universal time.

G. **Nonce:** a 4-byte value that typically begins at 0 and increments with each hash iteration to generate distinct hash values. This value is essential for resolving the

consensus puzzle in a blockchain that utilizes a proof-of-work consensus; otherwise, it may be excluded if an alternative consensus method is employed.

Within a blockchain, the block body contains all transactions. A transaction may denote any exchange between participants. The transaction begins once the participant starts and digitally signs it with their private key [39]. A secure hash algorithm, such as SHA-256, processes the transaction information to produce a fixed-size hash result. The signature generation algorithm utilizes the hash value and the sender's private key. As illustrated in Figure 2.5, given that transaction x has a payload p, a signature can be generated by signing the hash of p with the transaction issuer's asymmetric private key, denoted as K.

A node authenticates and confirms the transaction request, utilizing the sender's public key to guarantee integrity. In networks comprising multiple nodes, a consensus algorithm ascertains transaction validity. Upon validation, the transaction becomes an unalterable component of the ledger.

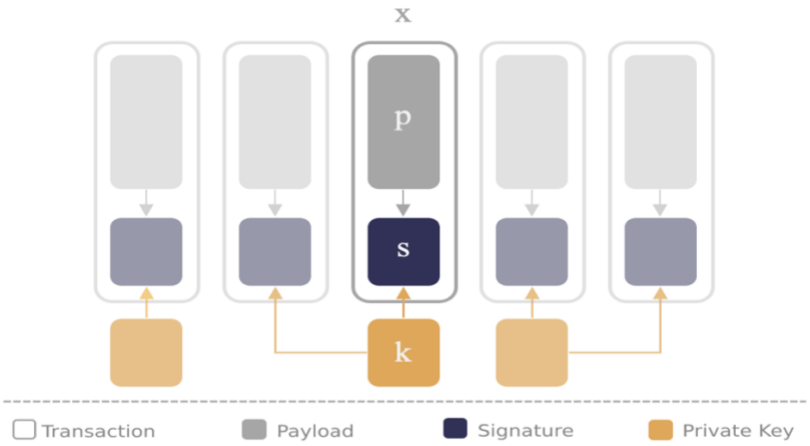

Figure 2.5 The Transaction Data in the Block Body

2. Peer-to-peer Network

A peer-to-peer network (P2P) is a distributed network structure. Users share their hardware resources, including processing power, network link capacity, printers, and storage capacity. Mediators are unnecessary for resource distribution, as these are accessible

directly through nodes. The members of these networks supply all necessary resources, whether services or content [40].

3. Consensus Mechanisms

Consensus processes are essential for establishing a reliable and trustworthy system in a decentralized environment lacking a central authority. Due to the absence of a centralized authority to authenticate or confirm a transaction, blockchain employs a consensus mechanism whereby network participants collectively agree on whether to approve or reject the transaction. This consensus process is the foundation of blockchain technology [41],[42]. They serve to verify the validity of a blockchain transaction. Different blockchain platforms employ various consensus algorithms, such as PoW, PBFT, Raft, and PoS, to maintain the consistency and integrity of data stored among geographically distributed nodes.

4. Ledger

A ledger consists of a series of blocks and can be called a compilation of transactions. Historically, traditional ledgers recorded service and product transactions with pen and paper. Ledgers were subsequently modernized through implementing electronic databases that depend on a centralized third party responsible for their management. This centralized architecture presents challenges, including SPOF attacks, as elaborated in Section 2.3. Recently, blockchain has facilitated the decentralized distribution of ledgers. A distributed ledger is a duplicated ledger across network members [43].

5. Smart Contract

Computer programs autonomously execute designated tasks upon the fulfillment of specific conditions. Every transaction is validated against the criteria before being recorded on the shared ledger. Smart contracts enable programs to be executed without third-party intervention [44]. Therefore, a smart contract is an agreement among multiple parties, dictating the conditions and terms of a collaboration. The contract concept can similarly elucidate how banks collaborate with their clients. For example, if a peer intends to transfer

funds via account X to account Y, a contract must exist to delineate and govern the transaction. This contract may delineate stipulations such as the ownership of account X, a requisite balance, and the existence of account Y, among others.

The conventional intermediary, represented by a bank, is supplanted via a network of voting devices that utilize executable code as contracts. Contracts define how nodes within the network interpret, validate, and execute transactions [45]. Specific platforms utilize various terminologies for smart contracts, such as chain code in Hyperledger Fabric.

2.4.3 Blockchain Characteristics

Bitcoin is widely recognized as the most successful and widespread application of blockchain technology, drawing attention to it. As time has passed, technology companies have realized blockchain's potential, which extends beyond bitcoin and financial industries. Thus, this subsection discusses the characteristics of blockchain that establish it as a revolutionary technology.

Decentralized: In a centralized system, a third party or single central authority will take on the responsibility of managing and maintaining information, transactions, or information verification. Being decentralized means no single individual or government controls the framework [46]. Instead, the decision-making process is transferred from a single individual to the participants in a blockchain network. Therefore, a blockchain network's many nodes will check and verify transactions, and a block will be added based on the approval of numerous nodes [47].

Secure: The blockchain's security aligns with the CIA Triad method for information security, which represents confidentiality, integrity, and availability [48].

1. Confidentiality means maintaining sensitive information securely and preventing unauthorized access to it. Permission-based blockchain technology has been developed to hide transaction data from participants within the network who should not view it [49].

2. Integrity means keeping information unaltered. Blockchain stores information within blocks connected like a chain using the block's hash value. The connected chain is then shared among members via a distributed network. As a result, all the members (nodes) could quickly detect any alteration or manipulation of any transaction [50]. Therefore, this unique characteristic of blockchain eliminates common misconceptions about retaining digital records, such as data substitution, deletion, or insertion.

   Consequently, even if a transaction is added incorrectly, it cannot be removed or replaced; instead, a new block must be created containing the correct transaction data. Furthermore, the participants could view both the newly added transaction and the first incorrect transaction in chronological order. This implies that information can be checked anytime, even after being issued for years. Based on this approach, the blockchain ensures a high level of integrity.

3. Availability refers to information being constantly accessible to those permitted. Blockchain comprises a distributed network that can function even in the event of a node assault. Consequently, availability can be guaranteed, and there is no SPOF [48].

Traceability: The ability for users to trace and track the transactions or movements of assets in the blockchain network. The distributed ledger technology supporting the blockchain allows for traceability by preserving a complete and transparent history of all network transactions and data [51]. Since every event can be traced, it is simple to determine who is in charge of each one, when it happened, and who approved the transaction's execution. Because of this feature, each participant must take accountability and behave honestly in every transaction [52].

Transparency: The data stored in every block, distributed across other linked nodes, is visible to all nodes. This means that every transaction and modification made to the blockchain ledger is visible to all nodes in the network [53]. Certain blockchain types can use private network channels to restrict nodes' access to certain information. Everyone joins the blockchain network, but access to a particular channel is restricted to approved users only. The channel becomes restricted for those who need to protect sensitive

information from others. In this situation, transparency is still ensured within the channel [54].

Persistence: Unlike centralized systems, the blockchain's availability does not rely on any entity, which might lead to far greater longevity and persistence [53].

2.4.4 Blockchain Categories

Blockchain can be permissionless or permissioned [55]. A permissionless blockchain lets any network members broadcast a block, while a permissioned blockchain limits publication to one member. On the other hand, in a permissioned blockchain, the network is controlled, while a permissionless blockchain does not restrict; instead, it allows any member to connect to the public internet, as seen in Figure 2.6. The next subsections will describe each blockchain type's distinct properties.

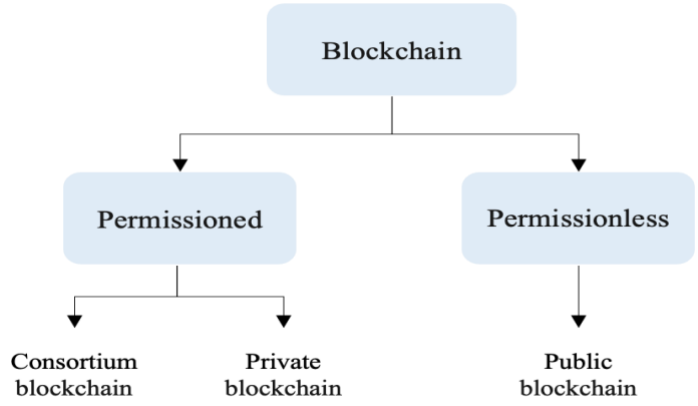

Figure 2.6 Categorizations of Blockchain Networks

1. **Permissionless blockchain**

A permissionless blockchain, also known as a public blockchain, is a decentralized ledger that allows any node to join the network and publish a block without requiring permission or authority. This type of network is typically open-source software that can be downloaded by any user without permission and run on millions of devices. Thus, any node can access a permissionless blockchain network without requiring user identification.

This type of blockchain network utilizes consensus algorithms to mitigate against any nodes connecting to the ledgers. One popular consensus algorithm is proof of work. Each node in the network employs a consensus process to verify transactions, which can result in resource consumption, such as power and storage costs. Most participants must validate each transaction that spreads across the network. This makes it almost impossible to tamper with the public blockchain. The latency level is always high in the public blockchain, coupled with limited transaction throughput. It takes a lot of time to propagate transactions and blocks across the network[56]. Examples of permissionless blockchain networks commonly applied in finance and cryptocurrency include Bitcoin [33] and Ethereum[57].

2. Permissioned blockchain

Users must first be authorized to publish blocks in a permissioned blockchain. There are limitations on who can use the network to access the blockchain and create transactions. Several restrictions can be applied to this type of network, such as allowing members to read the ledgers, limiting it to just authorized members, or allowing any members to create a transaction. A permissioned network may be closed or open source.

Permissioned blockchain networks employ a consensus algorithm, like PBFT, for publishing blocks. Nevertheless, they consume only a few resources. The specific reason is that a user's identity is necessary for participation in a permissioned blockchain network. Because of their authorization, the network establishes a certain trust status with its members. Furthermore, these attributes of networks contribute to accelerated consensus mechanisms and inferior operational costs.

Permissioned blockchain networks are suitable for organizations that require control over their blockchain or organizations that collaborate but are independent. These organizations may create consensus mechanisms according to the requisite trust level. In addition to trust, this network has the potential to achieve transparency, which could aid in making better business decisions.

In permissioned blockchains, two types of networks exist: private and consortium. The private blockchain is a centralized network controlled by one organization where only specific nodes can access and transact. Within this network, a single trusted node has

enough capability to establish roles for participants and alter rules that may hinder access to information. While it maintains a distributed feature, relying on a trusted peer poses a risk. If compromised, it can jeopardize the overall trust and security of the network. Private networks involve fewer participating nodes than public blockchains, resulting in more efficient resource utilization and enhanced transaction throughput[58]. Examples of private blockchain networks include Ripple[59].

Finally, a consortium blockchain network has the features of a private and public blockchain network. In a consortium blockchain model, nodes from various organizations form a decentralized network. The decentralized network formed with role-based structures gets distributed control access, resources, and security features. In contrast to private blockchains, which employ centralized administration, consortium blockchains share governance among the nodes. A consortium blockchain does not suffer from redundant computations since nodes work according to their roles by endorsing or committing transactions. This enhances the network's performance and throughput.

A consortium blockchain is mainly open source but has a modular architecture model that provides plug-and-play features and services. This feature makes it easy to adapt to many distributed solutions. It has been used in many industrial applications, such as banking sectors, healthcare, and the supply chain [58]. An instance of consortium blockchain is Hyperledger Fabric [60] and Corda [61]. A consortium blockchain will be utilized for the proposed model in this research. Table 2.1 shows a detailed description of the types of blockchain networks and their characteristics.

Table 2.1 Summarizes the different types of blockchain and their characteristics.

Criteria	Consortium	Private	Public
Network Structure	Decentralized	Centralized	Decentralized
Require Membership Services	Yes	Yes	No
Peers Joining the Network	Authenticate nodes	Authenticate nodes	Any node
Validate Transaction By	Selected nodes from multiple organizations	An organization	All nodes
Throughput Rate	High	High	Low
Energy Consumption	Low	Low	High
Efficiency	High efficiency	High efficiency	Low efficiency

2.4.5 Hyperledger fabric

It is an open-source blockchain platform. Its primary function is a permissioned ledger that prioritizes robust security and identity features[62]. With a unique architecture [63],[64], Hyperledger Fabric determines the processing of transactions by executing chain code written in popular programming languages such as Go or JavaScript. This process follows an execute-order-validate flow consisting of the following phases:

During the execution phase, a client application sends a transaction proposal to peers who want to endorse it as a policy of relative endorsement. This invokes a chain code function that interacts with the blockchain ledger. The endorsement is returned to the client once the endorsers successfully execute the chain code.

In the ordering phase, the client sends a combined transaction to the ordering service. The orderers combine multiple transactions into a single block and then broadcast it to all the selected peers within the network.

In the validation phase, each peer verifies the received transactions by checking the endorsement policy and updating the local ledger.

The Hyperledger Fabric platform's unique architecture and execute-order-validate flow provide a secure and efficient means of processing transactions, making it an excellent option for organizations seeking to leverage blockchain.

2.5 IPFS

IPFS[65] is a decentralized P2P distributed file system that incorporates a content-based hypermedia transfer protocol, reducing the risk of central server attacks. Simultaneously, during file uploads, IPFS computes the file's hash value, guaranteeing the integrity of the file post-upload and rendering it more appropriate for preserving critical data. IPFS relies on Distributed Hash Table (DHT), BitTorrent protocol, Merkle Directed Acyclic Graph (DAG), Git, and Self-Certifying File System (SFS). Merkle DAG is a tree-structured data format used to verify IPFS file integrity. IPFS divides files into data blocks, and the Merkle DAG leaf node stores each block's hash value. The parent node hash value is calculated from the leaf node hash value. Finally, the root node hash is the file IPFS hash. DHT is a decentralized key-value storage system that endows IPFS with robust distributed storage

and addressing functionalities, thereby alleviating the risk of an SPOF on the central server[66].

2.6 Applications of Hyperledger Fabric Blockchain

Recently, blockchain has rapidly gained interest as an innovative and promising solution to numerous problems related to digital data storage. Various researchers have utilized the Hyperledger Fabric blockchain in the literature to protect multiple forms of digital information, such as digital forensics data, healthcare data, academic records, and electronic voting.

In digital forensics, blockchain technology has been explored to securely store digital evidence and improve its transparency. For example, the authors in [13] presented a blockchain solution called BCFL based on the Hyperledger Fabric. The presented approach aimed to improve the reliability and credibility of digital forensic evidence stored in the cloud environment. The authors wanted to address the problems of obtaining valid electronic evidence from the cloud, which could be altered during transmission or storage. The BCFL successfully preserved the evidence's content and avoided the risk of its deletion.

Another study [67] focused on managing electronic medical records in digital forensics using Hyperledger Fabric. Medical images were securely transmitted, uploaded, and stored in encrypted form, and the chain of custody was transparent. The results confirmed that the Hyperledger Fabric solution protected the integrity and confidentiality of the forensic evidence.

In healthcare, blockchain is considered a secure way of storing patient data and a tool for communication between patients and healthcare providers. For example, the study in [6] presented a permissioned blockchain framework for ensuring the control of patient records. The study enabled patients to grant health personnel access to their health records. The framework integrated the Hyperledger Fabric platform and usage control (UCON), tracking all activities via the blockchain and UCON. In their study, the authors showed that Hyperledger Fabric was scalable, effective, and suitable for healthcare records.

Furthermore, a study in [9] established a framework utilizing Hyperledger Fabric and IPFS to guarantee the security and scalability of electronic health records. To increase security, even the information stored in IPFS was encrypted utilizing the public key encryption method. The results demonstrated that the encrypted hash prevented unauthorized parties from tracing the health data. Overall, this framework successfully implemented data security, privacy, scalability, and interoperability aspects.

In [68], they proposed a secure model for protecting healthcare information based on Hyperledger Fabric. This model enhanced the privacy of health records via an identity mixer, also known as Idemix. It helped preserve patient records while promoting unlinkability and anonymity.

The authors in [69] proposed VerifyMed 2.0, a permissioned blockchain framework for managing trust relationships in healthcare using Hyperledger Fabric. It solved the problem of verifying the credentials of healthcare professionals and complied with GDPR by storing only the hash values. The framework increased trust, security, and scalability by eliminating issues encountered with the previous Ethereum-based solution.

Another study in [70] presented a system for access and identity management using Hyperledger Fabric to address privacy and security concerns in healthcare. It adopted the RBAC to minimize the risk of unauthorized access, where permissions were granted according to the user's role. The proposed system incorporated a tamper-proof audit trail to detect unauthorized modification or access attempts. The system helped to securely and efficiently manage patient access.

Similarly, academia has discussed blockchain regarding the authenticity of certificates, their storage, and possible forgery. For example, researchers in [12] proposed a hyperledger fabric blockchain to store educational certificates in Pakistan. Their approach enabled universities and higher education commissions to easily and quickly obtain authenticated certificates. The results confirmed that the Hyperledger Fabric solution provided record security.

In [71], the authors proposed a blockchain solution called Educert-chain. This solution utilizes Hyperledger Fabric to securely validate educational certificates and the Raft consensus algorithm for transaction validation. The researchers also evaluated performance utilizing the Hyperledger Caliper tool, concentrating on throughput and latency. The authors confirmed that the EduCert-Chain model has better security features than previous works.

Blockchain was considered necessary for electronic voting to improve its security and efficiency. For instance, a study [14] presented an electronic voting system that employed a blockchain solution utilizing Hyperledger Fabric. The system aimed to avoid the vulnerabilities of centralized electronic voting systems, including altering the database and voting twice. It allowed the voter to remain anonymous throughout the process. The system also ensured the reliability of votes through smart contracts.

2.7 Applications of Blockchain in Islamic Literature

The hadiths, together with the Quran, are the primary sources of Islamic law, and it is vital for Muslims that they be kept safe. However, little literature deals with using blockchain to improve the security of digital hadiths. Therefore, this review adopts a broader viewpoint by examining previous studies that have explored the use of blockchain technology within the Islamic context.

Blockchain-based methods can be used in Islamic financial transactions to enhance global trust in Islamic finance [72]. Islam advocates honesty and transparency in all transactions, and blockchain effectively upholds these principles. In one paper[73], the solution known as ‘Saadiqin’ was provided, focusing on an Islamic banking structure corresponding to the elements and specifications of financial contracts. In their study, the authors investigated the feasibility of using Hyperledger Fabric to enhance the security and transparency of such Islamic financial transactions. The authors only outlined a general framework for integration and initial progress.

The study [74] proposed a Sukuk tokenization model employing the Ethereum blockchain. It aimed to explore the application of blockchain technology to reduce costs, increase transparency, and enhance the efficiency of the issuance process of the Sukuk, an Islamic-compliant financial instrument. To demonstrate the possibility of using blockchain in the Sukuk structuring process, the authors developed an intelligent al-murabaha Sukuk contract on Ethereum.

Blockchain can be utilized for zakat, one of Islam's fundamental principles. In [75], the researchers presented a blockchain-based Zakat collection and distribution system. The system enabled Zakat contributors to choose and distribute funds to verified beneficiaries. It utilized a blockchain to enhance accountability and transparency. The researchers aimed to reduce the associated distribution and collection expenses by eliminating intermediaries and enhancing the effective and secure administration of zakat funds.

The authors in [76] also presented a framework involving the Ethereum blockchain to improve trust in the Zakat collection. The distributed ledger autonomously documented all transactions, ensuring transparency and mitigating fraud, as records were immutable. The decentralized structure reduced costs by eliminating the necessity for third-party supervision, For example banks or government entities.

Blockchain can be utilized in sectors like halal food supply chain systems [77], encompassing procurement, material processing, handling, distribution, storage, and final delivery to the client. Regulators in predominantly Islamic nations can oversee a blockchain system, ensuring that food and beverages are halal. By employing a permissioned blockchain, the halal supply chain can maintain immutability, security, and confidentiality while providing the end-user with a fully transparent sight of the entire chain.

The study in [78] proposed a system based on the Ethereum blockchain to enhance transparency in Indonesia's halal supply chain. It involved slaughterhouses, halal bodies, retailers, abattoirs, consumers, and distributors. Each stakeholder could monitor the halal

status in real-time. This study enhanced trust by reducing information gaps and increasing data granularity.

Another study [79] presented a conceptual framework for utilizing blockchain and smart contracts to enhance transparency, integrity, and traceability within the halal supply chain. This was intended to address issues such as contamination and improve the functioning of the halal certification procedure by Islamic law. This theoretical framework could be used for future implementation and evaluation studies.

Furthermore, the study in [80] proposed a system for the halal industry based on Hyperledger Fabric, including numerous channels along with the Raft consensus algorithm. The authors emphasized how this structure addresses contamination and compliance issues and improves transparency. The network ensured that transactions were secure while maintaining high transaction throughput.

Blockchain can also be used to ensure that the digital Quran is secured to avoid any form of tampering and guarantee the authenticity of the content. For instance, in [15], the authors developed an Ethereum blockchain-based system to protect the digital Quran and verify its authenticity. In their system, the Religious Affairs Authority used the SHA-256 method to create a hash for each authentic Quran document and stored it on the blockchain to verify it in the future. Therefore, individuals could easily ensure the genuineness of a Quran digital document by comparing the hash value of the chosen Quran document with the hash value stored on the blockchain.

In [16], a study proposed a permissionless blockchain system to ensure the integrity of digital Quran texts without relying on third-party trust through Ethereum. The system comprises two active members: the Quran users, who utilize the application to confirm the authenticity of a Quranic verse through a consensus mechanism named PoS. Additionally, Quran miners publish the digital Quran in the application using a consensus algorithm named PoW. Therefore, the framework facilitates the protection of the quranic text and provides a way to confirm its authenticity.

Numerous survey papers on hadiths have been conducted [1],[20],[22], and [81]. One survey in [1] focused on Hadith authenticity in the digital age. It examined the challenges and technological advancements in verifying the authenticity of digital Hadith, which is susceptible to modifications and fabrications. Following their analysis, the researchers advised storing the Hadith in a blockchain to prevent attackers from tampering.

Therefore, the study [3] proposed a framework for the secure storage of digital hadiths based on a blockchain. The implementation utilized the Hyperledger Fabric platform and the PBFT [82] consensus algorithm. The study leveraged the immutability and transparency of the blockchain to protect hadiths from unauthorized modification. This framework only allowed specialized institutions in Islamic countries to add new hadiths to the blockchain. However, this framework has some limitations, which can be summarized as follows:

1) The proposed model's use of the PBFT consensus algorithm increases network traffic as communication adds additional workload to networks [17],[18]. Therefore, due to performance issues, Hyperledger Fabric no longer supports the PBFT algorithm [66].
2) This new research does not address the scalability problem when the volume of data exceeds the blockchain's capacity, which leads to increased transaction latency, nor does it include access control.
3) A significant limitation of their framework was the lack of scholar approval for Hadith authenticity, which risks adding non-authentic Hadiths. Additionally, approval should involve two scholars from different institutions to avoid centralized decision-making.
4) Furthermore, the system should facilitate the updating of hadiths to ensure that there is no incorrect information.
5) Additionally, the system architecture requires enhancement, such as a feature for Hadith students to upload new Hadiths so they can request validation from scholar Hadith. Subsequently, this leads to adding more Hadith to the system.

As a result, using blockchain technology for digital Hadith necessitates further investigation and improvement to strengthen security.

2.8 Summary

This chapter included an overview of hadith in Islam and discussed the challenges associated with centralized hadith systems. The chapter provided a comprehensive overview of blockchain and its constituent elements. All categorized blockchains have been presented. Blockchain has garnered considerable interest in different domains, including hadith and the Quran. Distributed ledgers, hashing algorithms, and cryptographic principles integrate to transmit data immutable and transparently. The chapter emphasized the technology's advantages in this domain and illustrated the restrictions of recent research. The following chapter will present a secure hadith storage model.

3 Chapter Three

Methodology

3.1 Introduction

This chapter delineates the methodology of the suggested model and explains how to fill the gaps identified in the preceding two chapters. It also outlines the phases of the study's general model and explains the rationale behind the choice of Hyperledger Fabric, Raft consensus, and IPFS for the proposed model.

3.2 Problem Visualization

Upon reviewing the previous chapters, several issues impacting digital hadith were identified. These issues include hadith fabrication, existing centralized techniques, and current hadith studies. Figure 3.1 visualizes these problems.

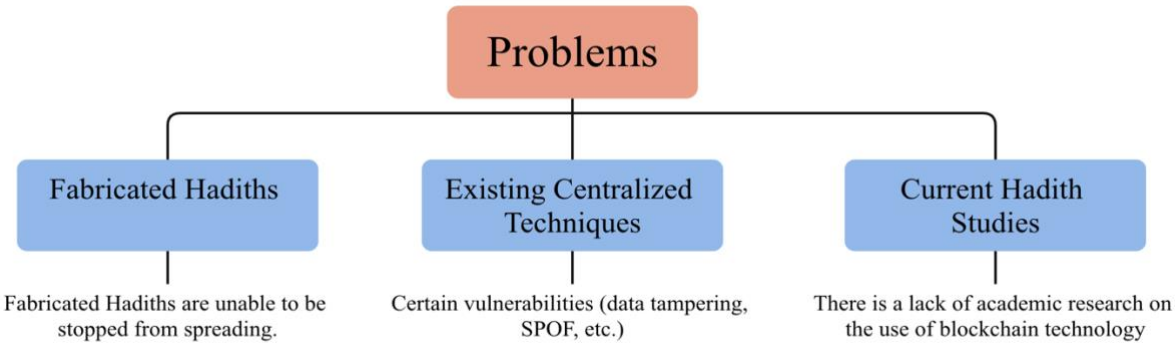

Figure 3.1 Problem Visualization

Firstly, the ease of fabricating hadiths and the inability to stop their spread on the internet. Secondly, centralized verification systems are vulnerable to internal and external threats, including SPOF and data tampering. Finally, academic research on using blockchain technology for digital hadith storage is scarce. Only one relevant study has been identified [3], which does not address scalability and access control.

In addition, the PBFT consensus mechanism requires significant network traffic for effective implementation. This framework may include non-authentic hadiths if a scholar does not confirm them. However, it would be better if hadith students could enter new hadiths, request confirmation from scholars, and facilitate update requests.

3.3 Concept of Solution

This research aims to solve the problems highlighted in the above section. The following steps encapsulate the concept of a solution:

1) Using Hyperledger Fabric blockchain to ensure the integrity, availability, transparency, and traceability of digital hadiths.
2) Integrate IPFS with the Hyperledger Fabric blockchain to reinforce the scalability of the proposed system.
3) Implement RBAC mechanisms to ensure that only hadith scholars can authenticate and approve hadiths.
4) Evaluate the proposed system by measuring its performance using metrics such as throughput and latency.

The proposed solution requires the approval of hadiths by two scholars from different institutions to avoid centralized decision-making. It also includes an update feature for hadiths to ensure the inclusion of correct information. In addition, hadith students can upload new hadiths for confirmation by a hadith scholar. This will lead to the addition of more hadiths to the system. Table 3.1 below presents a summary of the issue and the proposed solution.

Table 3.1 Summary of Solutions.

Issue	Solutions
Fabricated Hadiths	Using Hyperledger Fabric blockchain to provide a tamper-proof way to access authentic Hadiths that have been confirmed by Hadith scholars.
The existing centralized techniques	Using Hyperledger Fabric blockchain to provides a new structure for storing Hadith data, ensuring integrity, transparency, traceability, and availability. Additionally, it eliminates the risk of a SPOF or centralized control.
The limitation of current Hadith studies	Integrate the Hyperledger Fabric blockchain with Raft consensus and IPFS to ensure secure and scalable storage of digital hadiths. In addition, RBAC mechanisms are implemented to ensure that only hadith scholars can authenticate and approve hadiths.

3.4 Methodology

The research proposes a consortium blockchain model for digital Hadith security, utilizing Hyperledger Fabric as the blockchain platform and IPFS as an off-chain storage solution. Numerous consortium blockchain platforms exist, including Corda [83] and Quorum [84]. Nonetheless, Hyperledger Fabric's favorable reputation and active community render it the most appropriate blockchain platform for the proposed model. It is an open-source initiative by the Linux Foundation, thus undergoing rapid and regular enhancements by a substantial community of developers [6]. Furthermore, the framework's pluggable features and modular architecture provide a significant advantage. Businesses and enterprises extensively utilize the Hyperledger blockchain [63]. This platform executes smart contracts or chain codes, enabling participants to create scripts without intermediaries. The lack of economic incentives to reach consensus in Hyperledger Fabric networks reduces operational costs.

This research will employ Raft's [85] consensus. The Raft mechanism eliminates the communication overhead problem in PBFT-based consensus by allowing leader-backup node communication without backup node communication. Raft is a consensus protocol designed to withstand crash faults, categorized into three roles: leader, follower, and candidate. In the absence of a leader or their responsiveness, a follower transitions to the candidate state, wherein the candidate solicits votes from other nodes and ascends to leadership upon securing a majority of votes[86]. Nonetheless, it lacks resilience against assaults from malevolent nodes. Since all nodes are known and have network access permission, harmful tolerance is down, and the model's crash fault is considerably important. Additionally, Hyperledger Fabric nodes independently verify their accuracy before committing transactions to the ledger. This consensus exhibits high throughput and low latency, rendering it suitable for the system.
Figure 3.2 illustrates the techniques used in the proposed system.

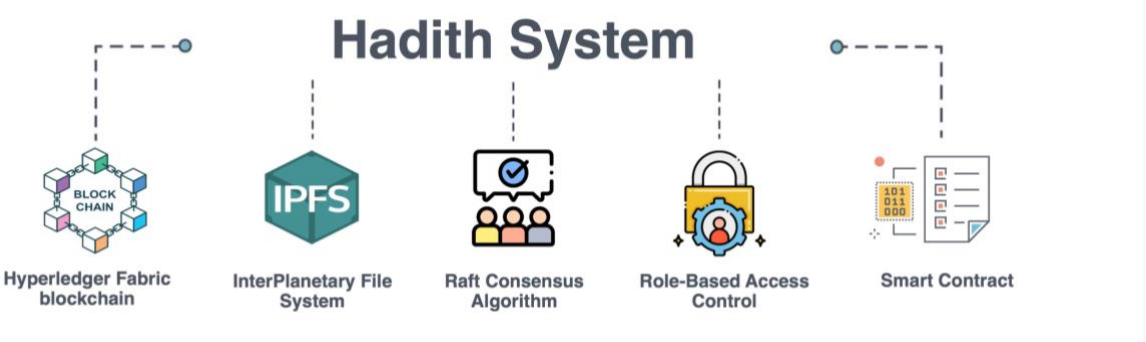

Figure 3.2 Key Components of the Proposed Hadith System

3.4.1 Requirements for the Hadith Systems

This subsection describes the requirements of the hadith system for secure and scalable digital hadith storage.

1. **Decentralized**: It ensures that no single individual or institute controls the model.
2. **Access Control**: It will implement RBAC for every system functionality, providing a solid barrier against unauthorized and unauthenticated users.
3. **Integrity**: The blockchain ledger must record hadith data in a write-only, non-deleting format. Therefore, it is impossible to overwrite or remove a transaction that contains mistakes. Instead, we must add a new transaction with accurate information in a separate block. Participants could view the original, mistake-filled transaction and the newly appended transaction in chronological sequence, complete with a timestamp.
4. **Availability**: System availability must not depend on a specific institution. If certain nodes fail, it will not impact the network's overall operation. It is also crucial to maintain service when most institutions include or exclude an institute.
5. **Transparency**: All network members must be able to see the data stored, meaning that each node can access all transactions and modifications to the blockchain ledger.
6. **Traceability**: All network members must be able to monitor and follow the flow of transactions within the blockchain network. It allows for simple identification of the individuals responsible for each event, the timing of these events, and the parties involved in executing the transaction. This characteristic ensures that all participants take ownership of their actions and conduct transactions honestly.

7. **Scalability**: The system should entail the creation of a network that can effectively accommodate potential growth in the number of institutes, nodes, and Hadith submissions

3.4.2 Proposed Model

The model is divided into five steps: registration, hadith verification request, hadith verification process, storage, and performance evaluation. Figure 3.3 illustrates these steps, and the following subsections explain each one.

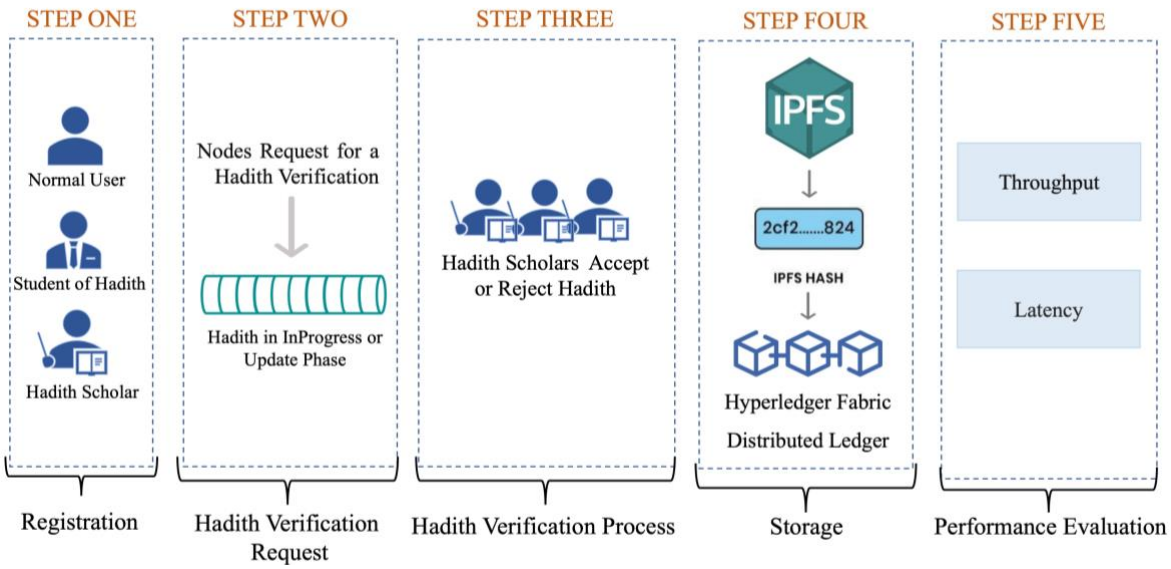

Figure 3.3 Proposed Model

Step 1: Registration

Hyperledger Fabric is a permissioned blockchain, meaning only authorized nodes can participate. Users submit their email, name, password, registration type (e.g., Hadith Student, Scholar, or Normal User), and institute ID to register. MongoDB stores this data, which forms the initial registration. The relevant Hadith institutes then evaluate the registration request, either approving or rejecting it. If approved, the institute submits a request to the Certificate Authority (CA), which issues the user's digital certificates and public and private keys, enabling secure participation in the network. Table 3.2 illustrates the participant nodes in the proposed model.

Table 3.2 Participant Nodes in the Proposed Model.

Participant	Description
Hadith Student	A student of hadith possesses information about hadith but lacks the extensive experience of a scholar. They are using the system to upload new hadiths and request verification from hadith scholars, enabling them to accept or reject it. They can upload and read hadiths.
Hadith Scholar	The hadith scholar plays a crucial role in determining the authenticity of Hadiths by endorsing the correct Hadith transactions. They can upload, accept/reject, update, and read hadiths.
Normal User	is a node that uses the system to verify the authenticity of Hadiths. They can only read hadiths.

Step 2: Hadith Verification Request

In the proposed model, any node can know the degree of authenticity of a Hadith and view its content after registering with the system. If a Hadith is not found in the blockchain, a student of Hadith can upload a new hadith and request its verification from hadith scholars. The Hadith verification request contains the Hadith, the first narrator, the hadith number, the source, and the degree of authenticity of the Hadith. At the same time, the complete Sanad with Hadith commentary is stored in IPFS, as shown in Figure 3.4. The system holds all Hadith verification requests for future validation by Hadith scholars in the InProgress phase. Keeping unapproved Hadiths in "InProgress" prevents unverified information from reaching users.

Figure 3.4 The Upload New Hadith

On the other hand, any hadith scholar on the network can send a hadith update request to correct a mistake in a Hadith. This privilege is exclusive to Hadith scholar nodes. The scholar is required to provide a note explaining the reason for the update, as depicted in Figure 3.5.

Figure 3.5 The Hadith Update Request

Step 3: Hadith Verification Process

When a Hadith scholar receives a request for Hadith verification, they begin a thorough verification process. There are various regulations to follow, including checking the Hadith format and all provided information. Hadith scholars approve after verification. When two Hadith scholars from different institutions agree on the authenticity of a Hadith, it becomes active. If one scholar rejects it, the Hadith is removed from the world state before activation.

Two approvals balance efficiency and accuracy. Too many approvals may slow the approval process, while too few may not provide enough verification. This method allows speedy action without giving any scholar too much authority over the decision. A single rejection can remove a hadith before activation, preventing users from accessing incorrect Hadith.

On the other hand, if two scholars from different institutions approve a Hadith update, it is updated. An update is not accepted if it receives two rejections from various institutions. Scholar identities and timestamps are recorded for all changes.

Step 4: Storage

This proposed model employs off-chain storage located external to the Hyperledger Fabric blockchain. The blockchain may encounter scalability challenges when handling substantial volumes of data, resulting in network performance difficulties. Hence, decentralized storage solutions like IPFS, Swarm, and Storj [87] have the potential to surpass these limitations. These solutions use a peer-to-peer distributed file system to shred

and distribute data to several network nodes for integrity and availability [88]. A primary challenge with these systems is a need for more traceability and transparency.

As shown in Figure 3.4, the blockchain holds hadiths and important Sanad information. At the same time, the complete Sanad with Hadith commentary is stored in IPFS. In other words, Hyperledger Fabric only stores the Sanad and Hadith commentary's hash value. This approach improves scalability.

IPFS is the most stable off-chain data storage. It has a strong community and well-written documentation[89], so it was chosen to integrate with the architectural design. IPFS has a commendable history of effectively implementing projects. In addition, significant organizations such as Cloudflare utilize IPFS to offer their cloud services. Over 230k peers utilize IPFS weekly, handling tens of millions of requests daily [90].

Step 5: Performance Evaluation

The primary aspect of the proposed model architecture evaluation focuses on performance. Performance evaluation is measured based on two crucial factors: throughput and latency. Throughput is defined as the number of successful transactions completed per second. In contrast, transaction latency refers to the time it takes for a transaction to complete between submission and response receipt. As stated in [91], throughput and latency are the primary quantifiers for comprehending blockchain performance and its limitations.

In addition, each requirement described in Section 3.4.1, including decentralization, access control, integrity, availability, transparency, traceability, and scalability, is reviewed to show how the proposed model meets them.

3.5 Summary

This chapter addressed the gaps highlighted in the preceding chapters and elaborated on the methodology of the suggested model. Hyperledger Fabric was chosen for its pluggable features, modular architecture, and community support. Raft consensus was chosen for its high throughput, low latency, and crash fault handling. To address scalability challenges, the model also integrates IPFS for off-chain storage. The next chapter will present the proposed model's design.

4 Chapter Four

Model Design

4.1 Introduction

This chapter explains the model's design and architecture. It examines Hyperledger Fabric technology in detail, focusing on its components, the design of the fabric network, and the transaction flow. The IPFS technology used is also discussed in more detail.

4.2 Hyperledger Fabric Structure

Public blockchains, like Ethereum, are developed as order-execute structures [92]. Transactions are initially ordered within the architecture and subsequently performed consecutively on all nodes in that exact order, as illustrated in Figure 4.1. This architecture constrains the blockchain's scalability and adversely impacts throughput. This architecture accommodates domain-specific languages like Solidity for smart contract design but does not support general-purpose languages like Java and Go.

Figure 4.1 Order-Execute Structure

Hyperledger Fabric blockchains addresses the issue through its innovative architecture, which employs an execute-order-validation structure[92]. Figure 4.2 illustrates the architecture's three phases: the execution stage, wherein a chain code in Hyperledger Fabric is developed and executed on several endorsers to perform transactions; as well as the validation stage, in which network nodes validate blocks obtained from the orderer service prior to updating their ledger. Subsequent subsections will provide a more detailed examination of Hyperledger Fabric's structure.

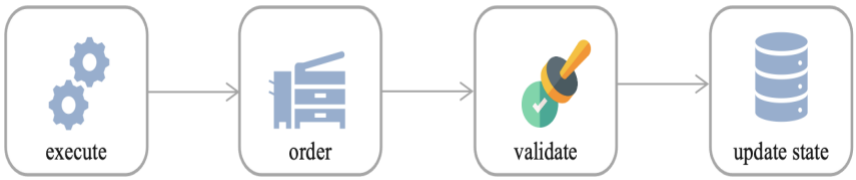

Figure 4.2 Execute-Order-Validate Structure

4.2.1 Structure Components

Nodes in Hyperledger Fabric are assigned distinct roles and responsibilities. They are classified as clients, orderers, endorsers, and committer peers.

The **client** is a peer responsible for generating the transaction. It may represent an application or a particular organization's portal. The client may engage with the network through the Hyperledger Fabric SDK or the REST web service. It is responsible for summoning the endorser node to submit transaction proposals.

Peer receives client invocation transactions and preserves the distributed ledger. The peer can be an orderer, endorser, or committer.

1. Orderer peers are service nodes that broadcast transaction messages as blocks to all peers in the network.
2. Endorser peers check the requester's roles and certificates. The endorser executes the chain code, simulates the transaction's result, and then relays the result to the client following affixing its digital signature. An endorser peer may also assume the role of a committer peer. Figure 4.3 delineates the distinction between a committer peer and an endorser peer.
3. Committer peers update the local ledger and commit transactions. By default, all peers function as committer peers.

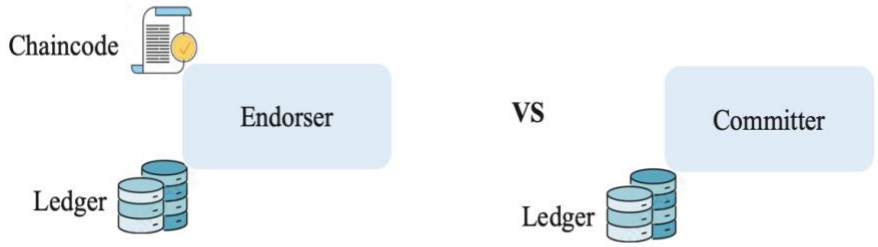

Figure 4.3 Endorser Peer Versus Committer Node

The **ledger** of each peer comprises two primary elements: the blockchain and the world state database, as illustrated in Figure 4.4. The world state database records the most recent state values of every transaction log within the blockchain. Hyperledger Fabric offers two distinct ledger state databases: LevelDB and CouchDB. LevelDB is embedded and included in nodes by default. In addition, it can store composite key queries, key queries, and key range queries. In contrast, CouchDB is an optional alternative that adopts a client-server model and enables rich queries beyond using just keys to perform operations based on IDs. CouchDB will be utilized in this study.

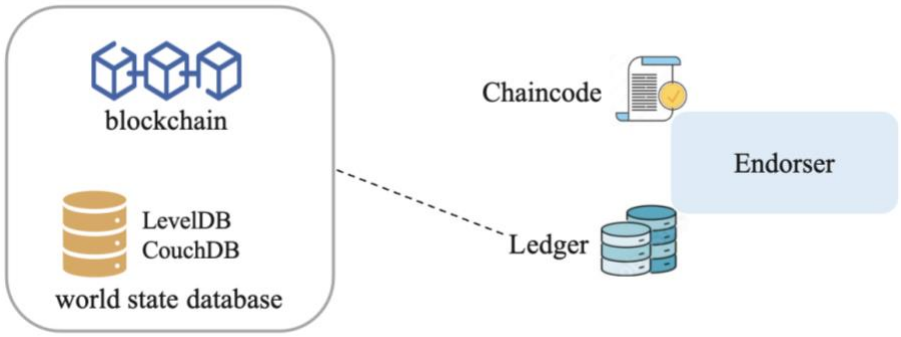

Figure 4.4 Internal Elements of a Node's Ledger

Chaincode, a crucial element of the architecture, embodies the same principle as a smart contract. A chain code encompasses business logic and governs the world state database; the chain code should be invoked initially to interact with the node's ledger via an external

application. For retrieving and updating the world state, the chain code performs the get() and put() functions, respectively.

The **CA** component is required to manage permissions by issuing identities to all network participants, including nodes and clients. Hyperledger Fabric's CA, known as Fabric-CA, is pivotal in facilitating authorization and handling identity. Every institution member receives an x.509 certificate from Fabric-CA. A single root certificate is produced for each institution.

4.2.2 Establishing the Fabric Network

The initial step in establishing a network fabric is to create three institutions: Institution1, Institution2, and Orderer Institutions. As depicted in Figure 4.5, a CA associated with each institution must establish the definitions of these institutions and the identities of their administrators. Each institution can select its CA provider or use the default Fabric-CA within Hyperledger Fabric.

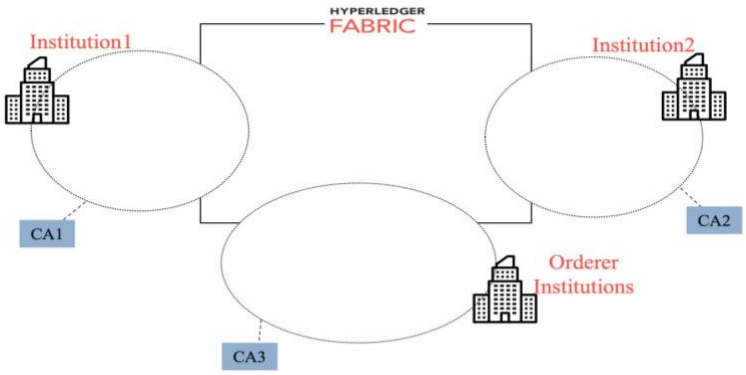

Figure 4.5 Establishing the Fabric Network

After this, Institution 1, 2, and Orderer Institutions must agree on the channel configuration in a "configuration block." This configuration block documents the institutions permitted to join components and engage on the channel, along with the policies delineating the framework for decision-making and attaining specific outcomes.

Channel setup is required for institutions to communicate securely. Various institutions may own multiple nodes upon joining nodes into the channel, as illustrated in Figure 4.6. Node1 is a peer that joins into channel H1. Node1, owned by Institution1, possesses a ledger L1, which is also retained by channel H1. The ledger is physically in Node1 but logically in H1. Furthermore, Node2, owned by Institution2, has been integrated into channel H1 and possesses an identical copy of the ledger. Within the network, there exist three orderers in the ordering service. The ordering service includes only the blockchain portion of a ledger, without the state database. Each institution node possesses x.509 certificates issued by the corresponding CA.

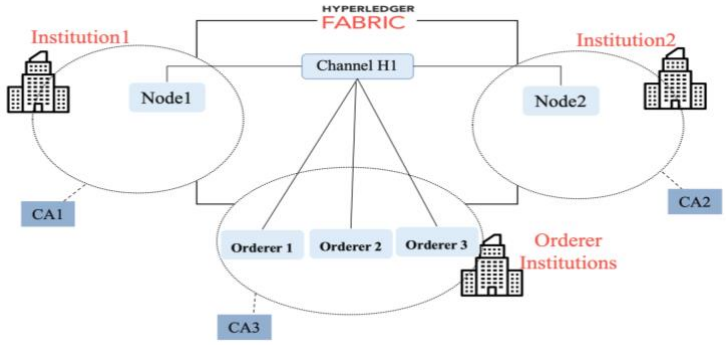

Figure 4.6 Defining Peers

The subsequent step involves installing chain code on Nodes 1 and 2, sanctioned by the pertinent node institutions and committed to the channel, as illustrated in Figure 4.7. It is important to note that the ordering service lacks chain code installation, as ordering nodes do not propose transactions.

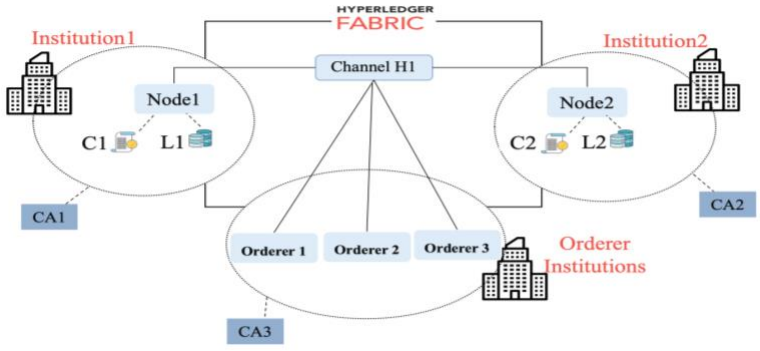

Figure 4.7 Completing the Fabric Network

Establishing a connection to a singular channel indicates the existence of just one logical ledger within the network. Consequently, Node1 and Node2 possess identical replicas of the ledger and the chain code. It can expand the networks by identifying additional institutions and nodes, in addition to establishing additional channels with specific configurations. Even within a single network, distinct channels may adhere to different regulations. The ordering nodes within the same network might manage various channels or establish isolated ordering services. Adding more channels to the network will facilitate the creation of additional chain code and logical ledgers.

4.2.3 Fabric Network Transaction Flow

An invoked transaction necessitates ledger updates, which differ from a query transaction. A single node cannot update the ledger; the network's consensus must validate the changes before implementing them in the nodes' ledgers. As a result, updating a transaction requires additional processing steps, known as consensus, that go beyond the two-phase steps needed for a query transaction. Figure 4.8 illustrates the Hyperledger Fabric transaction flow.

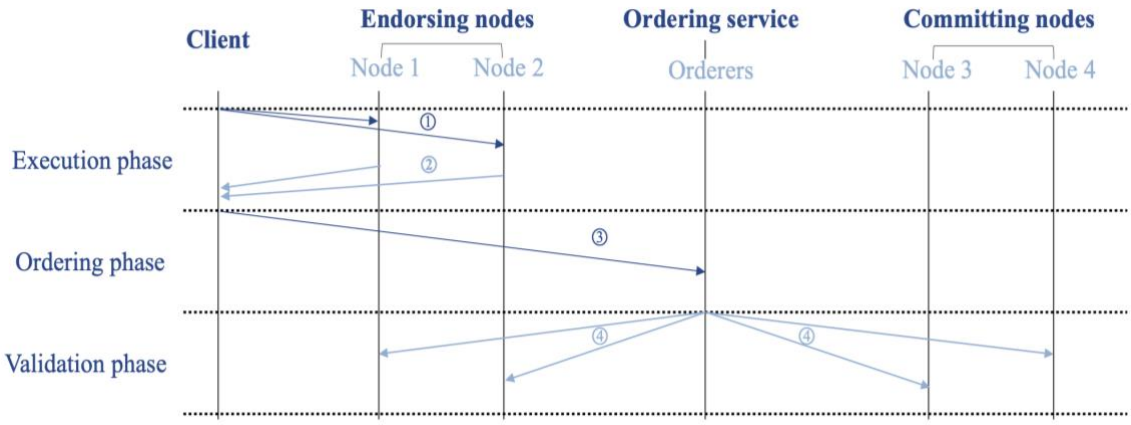

Figure 4.8 Hyperledger Fabric Transaction Flow

A. Execution

During the execution phase, a client application submits proposals to the endorsing nodes (Nodes 1 and Node 2① in Figure 4.8), following the specified endorsement policy. The endorsement policy delineates the criteria for a chain code that instructs a node on how to

ascertain the validity of a transaction. For instance, all network endorser nodes are required to validate a transaction. Subsequently, each endorsing node generates a transaction in two phases: first, the proposal is converted into an executable transaction through the execution of the chain code, and second, the transaction is endorsed with a digital signature utilizing the endorsing node's private key. The digital signature serves as evidence of the endorsing node's approval and ensures the integrity of the transaction. It guarantees that the endorsing node is accountable for endorsing the transaction as well as avoids tampering throughout the subsequent stages of the transaction lifecycle. Nodes that endorse the generated transaction return it to the client application. The client gets two transactions from Node1 and Node2 ② in Figure 4.8. This stage does not change the endorser's local ledger.

B. Ordering

After the client receives responses, the signature's authenticity is verified in every transaction, specifically its presence and origin. The client application determines the results if the proposed transaction is a query. At this stage, it terminates the query transaction process and does not forward to the orderer nodes. If the proposed transaction requires ledger updates, the client application transmits the transaction to the orderer nodes following the verification of the endorsement policy's completion (③).

The ordering node aggregates the transactions. Upon accumulating a specified quantity of transactions (batch size), the orderer node compiles these transactions into a "block." Upon the creation of the block, the orderer node disseminates the block through a consensus algorithm, ensuring uniformity among all orderer nodes. After all orderer nodes possess the identical block, they disseminate it to all channel nodes (④).

C. Validation

Upon receiving a block from an orderer node, all nodes execute transaction validation on the block. Each node verifies the endorsements inside the transactions to ensure they comply with the endorsement policy. Due to the orderer node sending each authorized transaction to nodes without judging the content, nodes must verify the transaction. Nodes will mark transactions invalid without changing their ledgers if they are incorrect. If the

transactions are valid, every channel node updates its state database and adds it to its blockchain [93].

The validation process at this stage differs from that in the preceding stage, wherein the client application gets transaction replies and then verifies the endorsement policy prior to submitting the transaction. Consequently, the node in the validation stage can reject the transaction if the client application submits an erroneous transaction without verifying the endorsement policy.

Furthermore, unlike in the initial phase, this stage does not require running chain code for validation. This is an essential characteristic of Hyperledger Fabric, as chain code is applied to the endorsers of a specific institution rather than the entire network.

Thus, the chain code can remain confidential to endorsers. Nonetheless, regardless of their roles as endorsers or committers, the channel's nodes receive the outcomes of the chain code constituting the transaction responses, thereby enhancing scalability and confidentiality within the network.

Finally, every node must notify the client application that the invoked transaction has been irreversibly added to the blockchain and indicate whether the block has been validated.

4.3 Raft consensus mechanism

The communication overhead issue presented in PBFT is a common issue in BFT-based consensus. By permitting communication between the leader and the backup nodes without requiring the backup nodes to speak with one another, it is removed in crash fault tolerance (CFT)-based consensus. Ongaro and Ousterhout (2014) proposed Raft, a consensus mechanism that is resilient to crash faults.

Three different node types make up the consensus: Leader, Follower, and Candidate. These nodes do, however, occasionally switch roles, and any node can take the lead by voting. Time is separated into terms, which are arbitrary lengths of time. The leader begins processing the client-submitted transactions. To keep the transactions in order, the leader verifies each one and gives each valid transaction a transaction index. A block of these transactions is created by the leader, who simultaneously sends out requests to each follower. Every follower replicates the block, and the leader receives an acknowledgement to verify the replication. The network is led by a single person.

The follower waits for a randomly chosen timeout period before entering the candidate state if the leader is absent or has not responded for a while. After that, the candidate asks other nodes for votes; if it gets a majority of the network's votes, it takes the lead. The node may receive a message from another node claiming to be the leader while it is waiting for votes. In that scenario, the node compares its term index to the node's alleged election term index. It must identify the other node as the leader if its word index is lower. It can, however, reject the message from the node claiming to be the leader and keep gathering votes if its term index is higher. This node is aware of the most recent changes if its term index value is higher. The entire procedure is repeated with a brief break chosen at random if no contender receives enough votes. A fresh vote will be started by the node given the shortest pause.

The crash tolerance problem is resolved by the Raft consensus algorithm. Up to 50% of the crash fault can be agreed upon by consensus. Nonetheless, it lacks resilience against assaults from malevolent nodes. Since all nodes are known and have network access permission, harmful tolerance is down, and the model's crash fault is considerably important. Additionally, Hyperledger Fabric nodes independently verify their accuracy before committing transactions to the ledger. This consensus exhibits high throughput and low latency, rendering it suitable for the system.

4.4 IPFS Off-chain Storage

It is a crucial component of the model, denoting any data storage external to the blockchain. Due to data storage constraints within the blockchain's distributed ledger, substantial data contributes to excessive chain growth. Utilizing off-chain storage mitigates the deterioration of Hyperledger Fabric's performance when managing large amounts of data within the blockchain [94]. Off-chain storage is efficient for Sanad with Hadith commentary data, which is too large for a distributed ledger.

Off-chain storage computation significantly reduces network consensus computation. Since most processing happens off-chain, the network can execute queries with reduced computational demands, as on-chain data merely retains a fingerprint of the off-chain information.

However, only the fingerprint hash is stored on the chain, jeopardizing the data's availability. In other words, if data is not included in the blockchain, its availability is considered at risk. Nevertheless, IPFS is decentralized and distributes and hosts data among nodes to ensure availability. Data can be retrieved from the nearest node within the network.

IPFS partitions files larger than 256KB into multiple 256KB blocks, referencing them through hash pointers. IPFS possesses numerous advantageous properties for various use cases, including the following:

No restrictions on file types: IPFS treats all files in the same manner as plain text and has no restrictions on file formats.

Easy data sharing: IPFS, as a distributed storage system, generates a hash number for every file or data stored. This protocol facilitates easy file exchange among various nodes within the network.

Tamper-proof: Information in IPFS is accessible via content addressing rather than location addressing, as in HTTP. Content identifiers are required to determine the location of files in content-based storage. IPFS generates a content identifier from the data's cryptographic hash, effectively acting as a fingerprint of the information. Obtaining an identical cryptographic hash is exceedingly challenging if the file's content is altered. Consequently, the inability to overwrite IPFS files ensures data integrity.

4.5 Summary

This chapter examined the model's design and architecture. It also discussed the structure of the fabric network and the transaction flow of Hyperledger Fabric to show how they work together to ensure secure and efficient operation. The integration of IPFS and its role in decentralized storage were also explained. The detailed analysis of these technologies provides a solid foundation for understanding the system's implementation.

5 Chapter Five

Implementation

5.1 Introduction

This chapter presents the implementation of a Hadith storage model that combines the capabilities of the Hyperledger Fabric blockchain with IPFS. Hyperledger Fabric version 2.5 was utilized for implementation, as it was the most recent version available when this research was written. Section 5.2 illustrates the deployment of the Hyperledger Fabric network, including the chaincode implementation. Section 5.3 presents the user's interactions with the system. Section 5.4 demonstrates the experiments conducted and the results obtained. Section 5.5 discusses that our model achieves decentralization, access control, integrity, availability, transparency, traceability, and scalability. Section 5.6 compares our model with other models. The source code for the implementations can be found at GitHub https://github.com/BashayerAlkalifah/DigitalHadiths-blockchain , see Appendix.

5.2 Deploy Hyperledger Fabric Network

Establishing the network necessitated the installation of several Fabric prerequisites, comprising cURL, Git, NodeJS v16.12.0, NPM v6.14.4, and the Go v1.22 programming language. Then, it required installing the Fabric binaries files, which included the essential directories /bin and /config. Finally, Docker Compose v1.14 and Hyperledger Fabricv 2.5 were installed.

The fabric network of our building comprises two organizations, each with one peer. All components are linked to a single channel, and a Raft consensus for three orders is implemented.

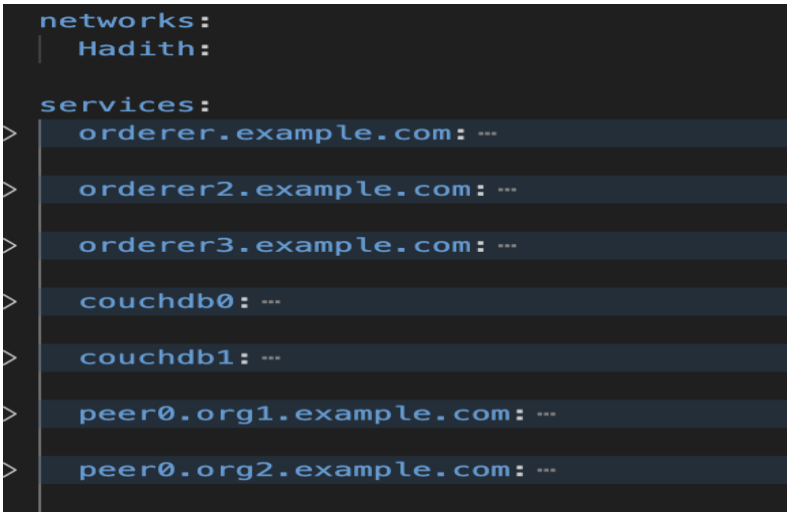

Figure 5.1 Definition of Bootstrapping the Network

The establishment of the network necessitated modifications to various configuration files and scripts to fulfill specific requirements, as detailed below:

1. **Configtx.yaml**: It contains configuration details along with channel transaction files necessary to create the genesis block. This file comprises multiple sections:

   I. Organizations: This section delineates the specific organizations along with their identifying information, as specified in the configuration file. The particulars encompass the name of the organization's Membership Service Provider (MSP) ID, tasked with managing all cryptographic functions, including issuance, verification, signing, and chaining; MSPDir, a directory housing the organization's cryptographic materials; and Anchor peers, utilized to maintain data synchronization among organizational peers by designating their port and host. The orderers must be delineated. Three orderers are involved in the implementation.

   II. Orderer: All parameters related to the orderer are detailed in this section. The parameters encompass the orderer consensus, host, and port addresses; BatchTimeout—denoting the duration the orderer must wait prior to batch creation; MaxMessageCount—indicating the maximum block size; PreferredMaxBytes—referring to the optimal maximum block size in bytes; and AbsoluteMaxBytes—representing the absolute permissible byte limit for a block.

   III. Application: It utilizes the genesis block's default settings.

IV. Profiles: It is deemed essential in the Configtx.yaml as it consolidates all prior configurations and delineates the network's structure. It comprises two components: the first pertains to channel configuration, while the second relates to the configuration of the genesis block.

2. **Fabric-ca**: In this instance, the CA supplied via Fabric was utilized to confer certificates to the three organizations; nonetheless, in practice, an intermediate CA from a reputable corporation must be employed.
3. **Crypto-config**: It contains the certificates for all participants.
4. **Docker-compose.yaml**: It contains the docker container for entities, including peers, orderers, and CouchDB, as illustrated in Figure 5.1. The configurations of these entities encompass domains, ports, paths, and other critical parameters necessary for proper network setup.

5.2.1 Chaincode

This subsection describes the chaincode functions for Add Hadith, Update Hadith, Approve and Reject New Hadith, and Approve and Reject Updated Hadith.

A. Add Hadith Chaincode

Authorized users add new Hadith records to the blockchain. Two scholars must approve the Hadith's "in progress" status before it becomes "active". Algorithm 1 delineates the flow of the Add Hadith chaincode.

Algorithm 1. Case Add Hadith

Input: ctx (context), hadithData

Output: Transaction ID or error.

```
1.  Get client identity using ctx.stub
2.  If registrationType != 'scholar' and  registrationType != 'StudentOfHadith',
3.    throw "Not authorized to perform this operation"
4.  If Hadith already exists
5.    throw “the Hadith already exists”
6.  Store Hadith in the blockchain world state
7.  Return the transaction ID
8.  Catch any errors, and throw a new error
```

B. Update Hadith Chaincode

The scholars update a hadith record. The updated Hadith will become "active" after two scholars approve it. Algorithm 2 defines the flow chaincode for updating the hadith.

Algorithm 2. Case Update Hadith

Input: ctx (context), hadithData

Output: Transaction ID or error.

1. Get client identity using ctx.stub
2. If registrationType != 'scholar'
3. throw "Not authorized to perform this operation"
4. Fetch the previous Hadith
5. If the previous Hadith has the status in progress
6. throw " Hadith is in progress "
7. Store update Hadith in the blockchain world state
8. Return the transaction ID
9. Catch any errors, and throw a new error

C. Approve and Reject New Hadith Chaincode

Only scholars can approve or reject a new hadith. Algorithm 3 removes rejected hadiths from the world state while maintaining them in the blockchain ledger. Activated hadiths cannot be deleted. In Algorithm 4, the system first validates approvals, then updates the hadith's status and records the transaction on the blockchain.

Algorithm 3. Case Reject Hadith

Input: ctx (context), HadithId , DeletedBy

Output: Transaction ID or error.

1. Get client identity using ctx.stub
2. If registrationType != 'scholar'
3. throw "Not authorized to perform this operation"
4. Fetch the Hadith by hadithId
5. If Hadith has the status active
6. throw “The Hadith is already active “
7. Delete Hadith and any associated approvals from the blockchain state
8. Return success message with the transaction ID
9. Catch any errors, and throw a new error

Algorithm 4. Case Approve New Hadith – Process 1

Input: ctx (context), HadithId , userStr

Output: ACTIVE or INPROGRESS or error.

1. Get client identity using ctx.stub
2. If registrationType != 'scholar'
3. throw "Not authorized to perform this operation"
4. Fetch the Hadith by hadithId
5. If Hadith has previousHadithId
6. throw "This operation is intended for approving new Hadith submissions only”
7. If Hadith has the status active

```
8.      throw “The Hadith is already active”
9.    Query approvals for hadithId
10.   ScholarCount = 0
11.   For each approval:
12.      If approval matches user's registrationType and orgId
13.          throw " This Hadith has already been marked as approved by your institution"
14.      If approval.registrationType == 'scholar', increment scholarCount
15.   If scholarCount > 0
16.      return ACTIVE
17.   Else, return INPROGRESS
18.   Catch any errors, and throw a new error
```

Algorithm 4. Case Approve New Hadith – Process 2

Input: ctx (context), ApprovalData

Output: Transaction ID or error.

```
1.    Get client identity using ctx.stub
2.    If registrationType != 'scholar'
3.      throw "Not authorized to perform this operation"
4.    If hadithStatus is active
5.      retrieve the existing Hadith
6.      update Hadith status
7.      store Hadith in the blockchain state
8.    Else remove hadithStatus from the approval data
9.       Store the approval data in the blockchain state
10.      Return the transaction ID
11.   Catch any errors, and throw a new error
```

D. Approve and Reject Update Hadith Chaincode

Scholars can start this process. The blockchain world state does not update the Hadith if two scholars reject the update. If two scholars approve the update, the old Hadith is removed from the world state and replaced with the updated one. Algorithm 5 delineates the approve And Reject Update Hadith chaincode flow.

Algorithm 5. Case Approve and Reject Update Hadith – Process 1

Input: ctx (context), HadithId , status, userStr

Output: ACTIVE or INPROGRESS or REJECTED or error.

```
1.    Get client identity using ctx.stub
2.    If registrationType != 'scholar'
3.          throw "Not authorized to perform this operation"
4.    Fetch the Hadith by hadithId
5.    If Hadith has no previousHadithId
6.      throw " This operation is to approve updates to an existing hadith”
7.    If Hadith has the status active
8.      throw “The Hadith is already active”
9.    Query approvals for hadithId
10.   approvalCount = status == 'approved’? 1: 0
11.   rejectionCount = status == 'rejected’? 1: 0
12.   For each approval:
13.     If approval.createBy == user's email
```

```
14.            throw " This Hadith has already been marked as approved or rejected by you"
15.        If approval.orgId == userOrgId
16.            If approvalStatus == 'approved' and userStatus == 'approved'
17.                throw " A scholar from your organization has already approved this Hadith."
18.            If approvalStatus == 'rejected' and userStatus == 'rejected'
19.                throw " A scholar from your organization has already rejected this Hadith"
20.    If approval.status == 'approved'
21.        increment approvalCount
22.    else if approval.status == 'rejected'
23.        increment rejectionCount
24.    If approvalCount >= 2
25.        return ACTIVE
26.    else if rejectionCount >= 2
27.        return REJECTED
28.    return INPROGRESS
29.    Catch any errors, and throw a new error
```

Algorithm 5. Case Approve and Reject Update Hadith – Process 2

Input: ctx (context), ApprovalData

Output: Transaction ID or error.

```
1.    Get client identity using ctx.stub
2.    If registrationType != 'scholar'
3.      throw "Not authorized to perform this operation"
4.    If hadithStatus is rejected
        delete update hadith and any associated approvals from the blockchain state
5.    If hadithStatus is active
6.      retrieve the Update Hadith
7.      update its status
8.          store update hadith in the blockchain state
9.          delete previous hadith and any associated approvals from the blockchain state
10.   Else remove hadithStatus from the approval data
11.     Store the approval data in the blockchain state
12.     Return the transaction ID
13.   Catch any errors, and throw a new error
```

5.3 Dashboard

This subsection shows the user's interactions with the system. Figure 5.9 shows the Postman API dashboard for role-based user registration. Users register by entering their email, name, password, registration type (Hadith student, Hadith scholar, or normal user), and institution ID. After successful registration, the system will display the message User successfully registered; please ask the administrator to activate the account. This message signifies that an administrative check is necessary to complete the registration. MongoDB saves the user's registration data and marks it inactive until further approval.

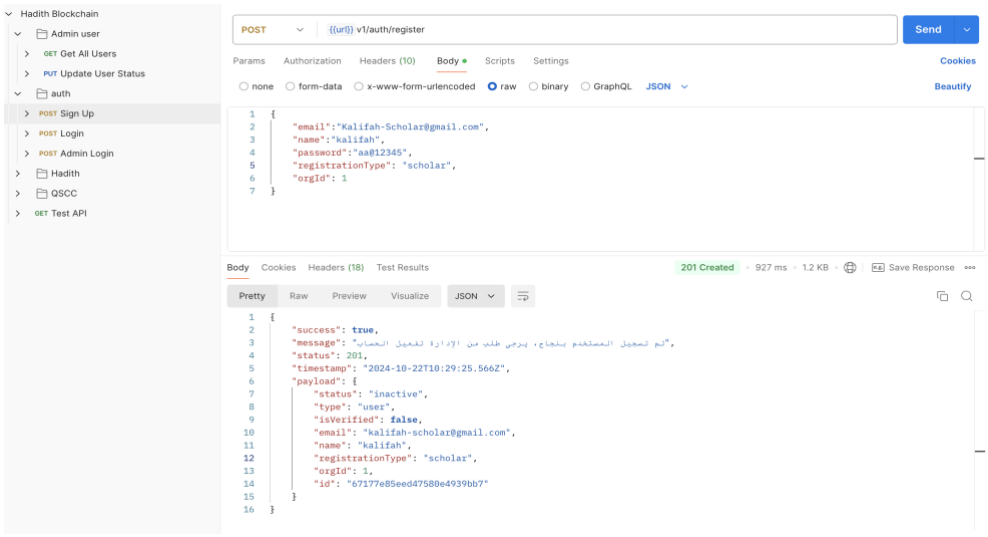

Figure 5.2 Sign-Up Page in Postman

The administrator of a relevant Hadith institution enters his login details, including an email address and a password. After logging in, the administrator can view all join requests from Hadith students, Hadith scholars, and normal users who wish to join the institution. In Figure 5.10, the response shows Kalifah as a scholar whose account is inactive and awaiting approval from the administrator.

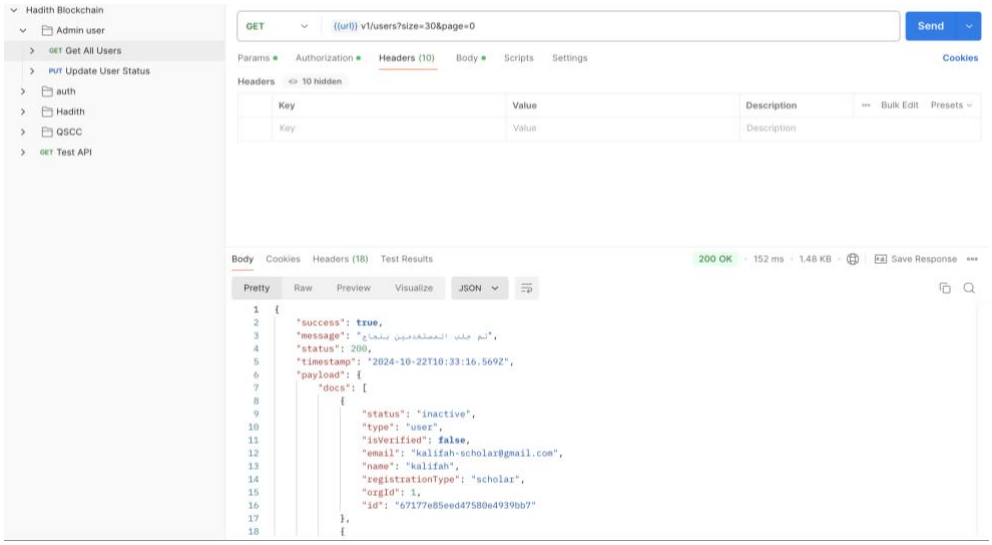

Figure 5.3 Get All Users in Postman

Figure 5.11 shows that the admin has accepted the Hadith scholar with the user ID ' 67177e85eed47580e4939bb7'. The status code 200 indicates that the request has been

processed and that the scholar can now use the platform. As seen in Figure 5.12, the scholar receives public and private keys for secure participation in the network.

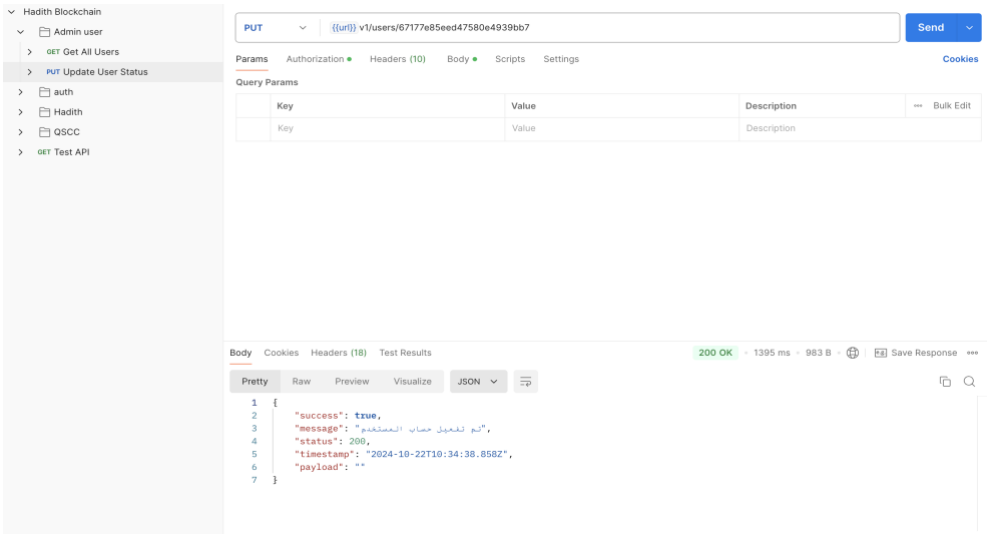

Figure 5.4 Accepting a User via User ID in Postman

Figure 5.5 Public and Private Keys for a User

Figure 5.13 illustrates the Hadith addition. The hadith student or hadith scholar enters essential information, such as the hadith text, the first narrator, the source, and the ruling, along with a document containing the full Sanad and hadith commentary. The system will store the document in IPFS as off-chain data and store the other information on the blockchain. After input, the system returns a success message and a unique hadith ID. If the hadith already exists, the system displays the message Failed to add Hadith: Hadith with ID f1dc616e7ead0ce1e1c0f3dd46aeeeb7a00834427acd293eb38205110ef011b9 already exists.

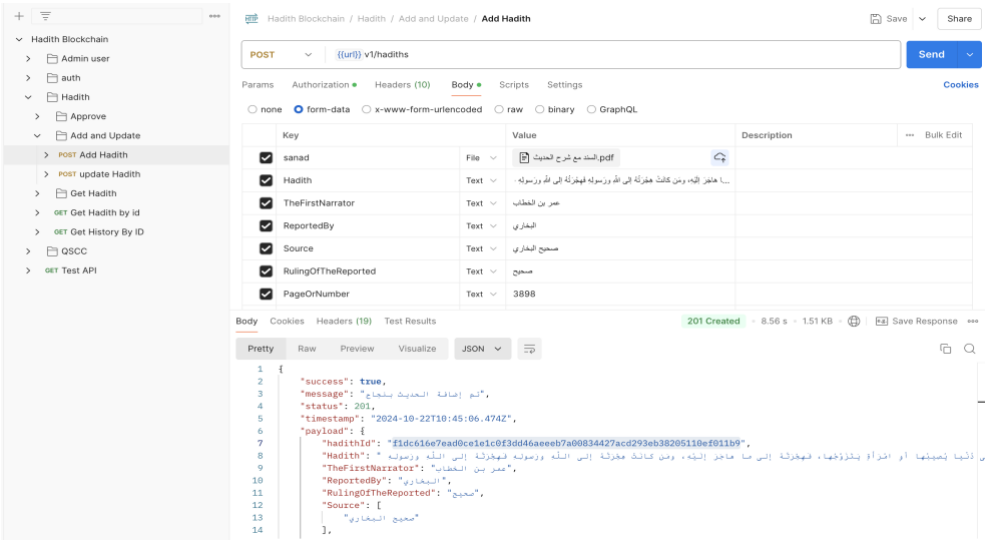

Figure 5.6 Adding a Hadith in Postman

Figure 5.14 shows how to retrieve details about a hadith after its creation. The Hadith is in progress until two scholars from different institutes approve it.

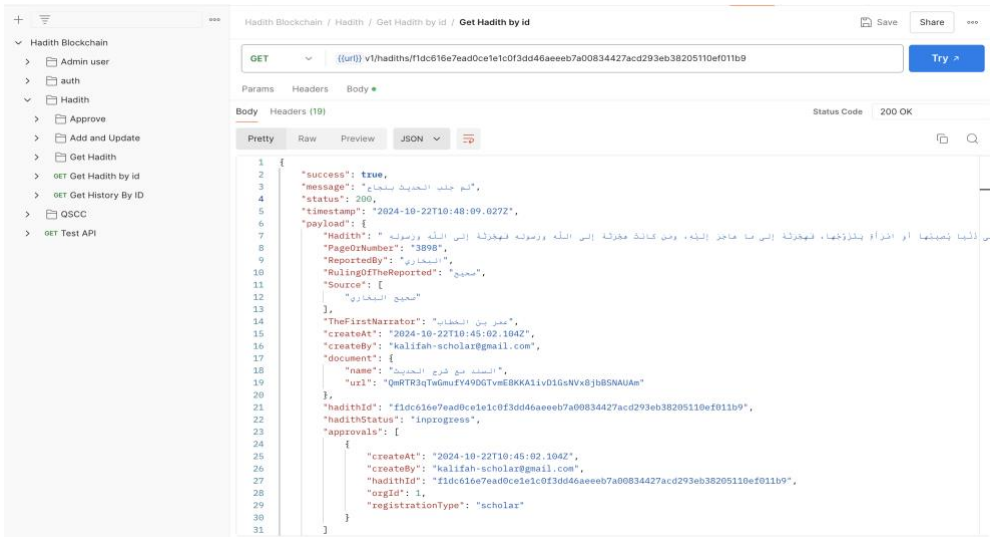

Figure 5.7 Retrieving Hadith by ID in Postman

In Figure 5.15, a scholar attempts to approve a hadith their institution already approved. If a Hadith student or normal user tries to approve a hadith, the system responds, "You are not authorized to perform this operation."

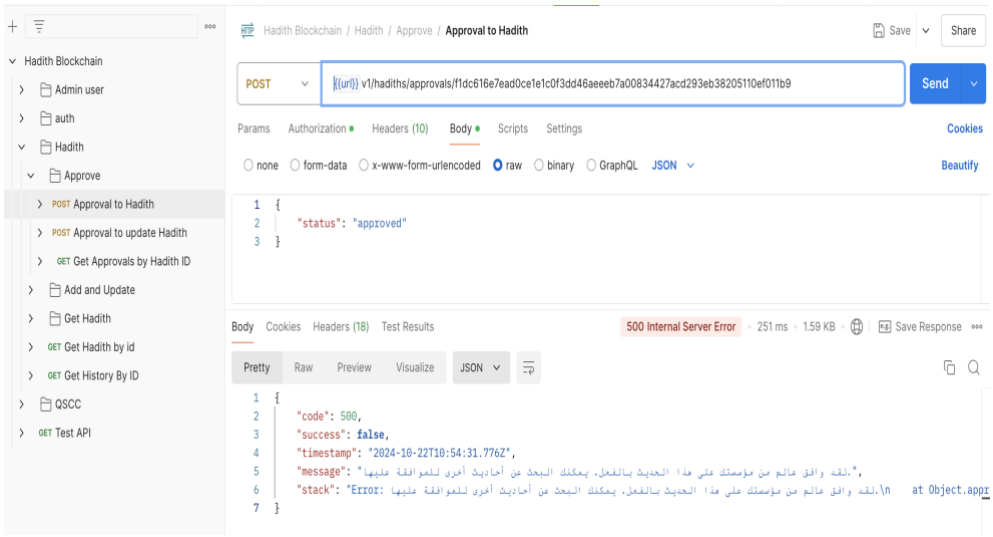

Figure 5.8 Error Response in the Case of Double Approval by the Same Institution

Figure 5.16 illustrates how obtaining both approvals will make the hadith active.

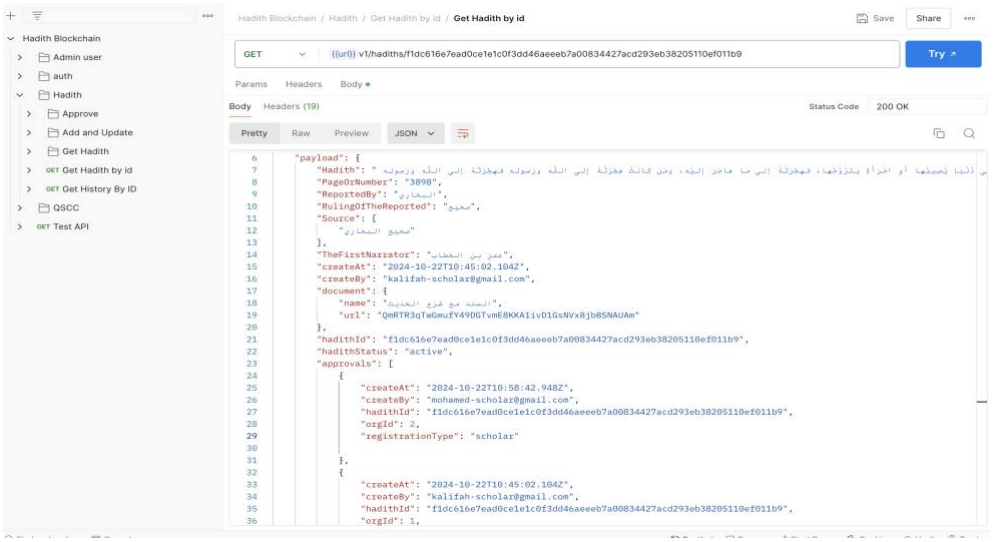

Figure 5.9 Active Hadith Status

Figure 5.17 shows a hadith history. This history log records the creation, updates, status changes, and approvals of individual hadith entries. The label "UPDATED" indicates the status update of a hadith.

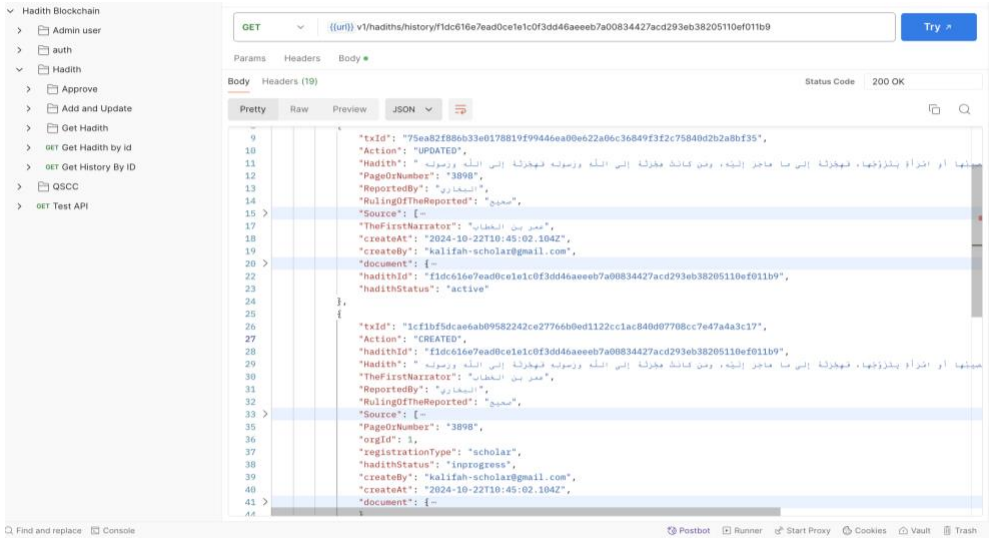

Figure 5.10 Retrieving Hadith History by ID in Postman

Figure 5.18 illustrates removing a previously added and deleted hadith from the World State database. However, Figure 5.19, displaying the retrieved hadith history from the blockchain, confirms that a scholar with the email address mohamed-scholar@gmail.com was responsible for the deletion.

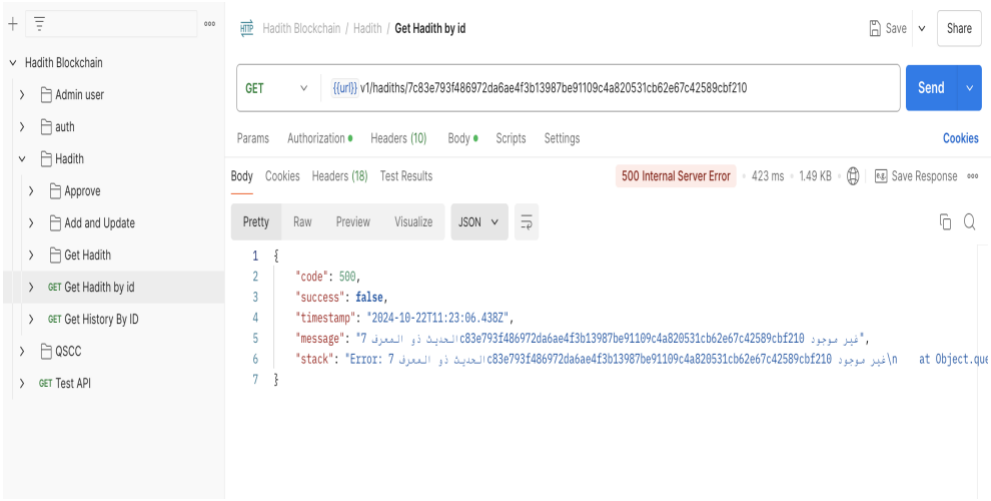

Figure 5.11 Retrieving Removed Hadith in Postman

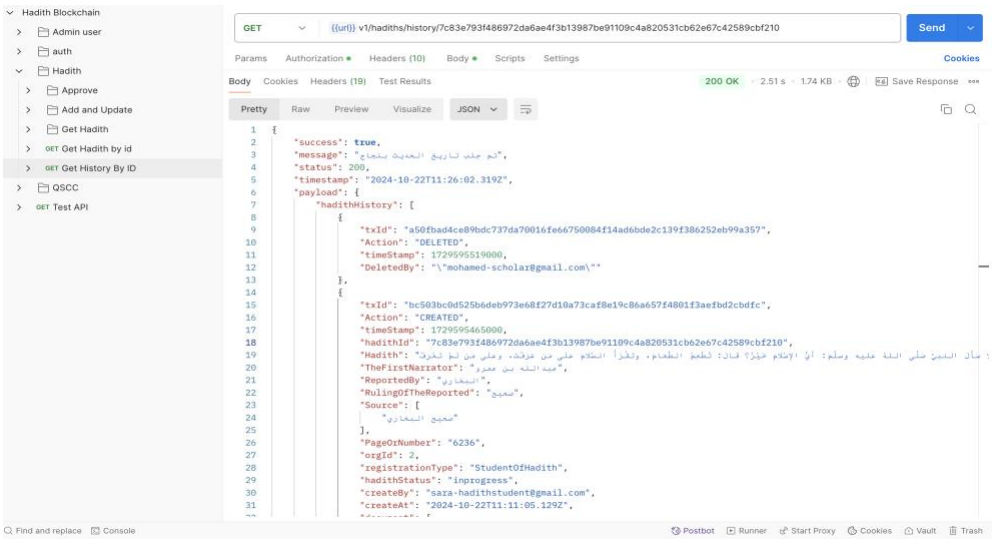

Figure 5.12 Retrieving Removed Hadith History in Postman

Figure 5.20 illustrates how to query a specific block using its block number. The response comprises the block header, which includes the previous and data hashes. The block data also contains all the transactions within it.

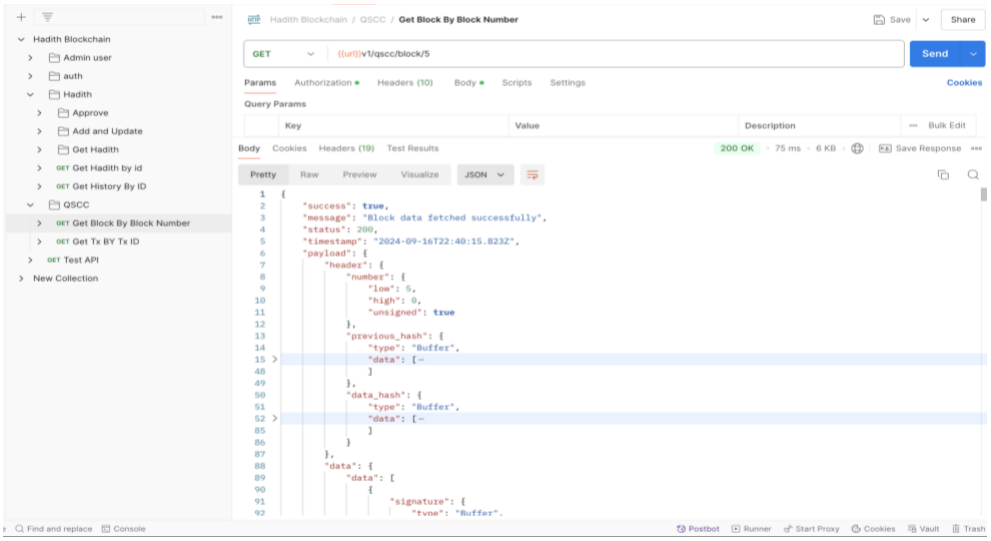

Figure 5.13 Retrieving a Block by Block Number in Postman

Figure 5.21 shows how to retrieve transaction details using a transaction ID. The response includes transaction endorsements and digital signatures from multiple institutions.

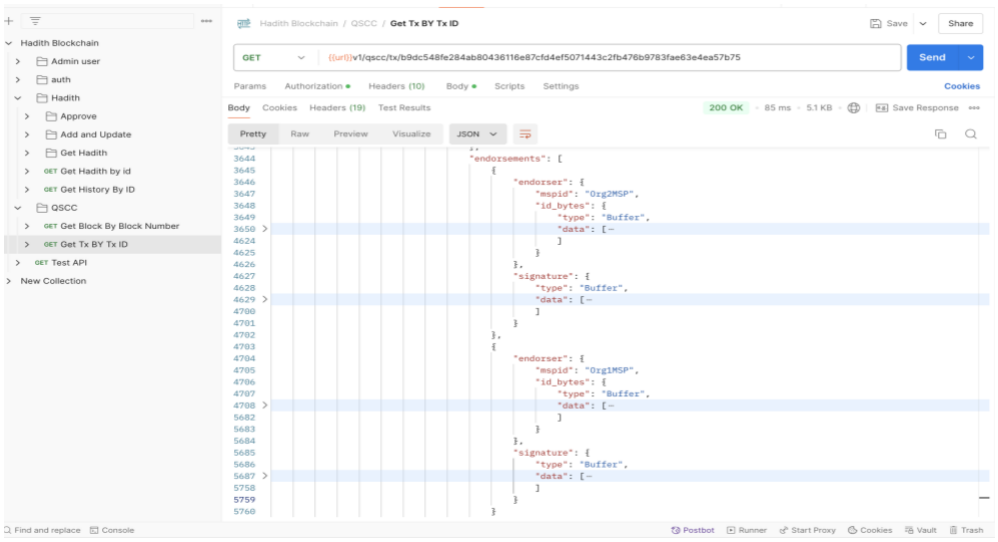

Figure 5.14 Retrieving a Transaction-by-Transaction ID in Postman

Finally, Figure 5.22 illustrates that any hadith scholar on the network can send a hadith update request to correct a mistake in a hadith. The scholar must provide essential information, such as the ID of the hadith he wants updated and a note explaining the reason for the update. The updated Hadith will become active after two scholars approve it. If a Hadith student or normal user tries to send a Hadith update request, the system will respond that you are not authorized to perform this operation, as illustrated in Figure 5.23.

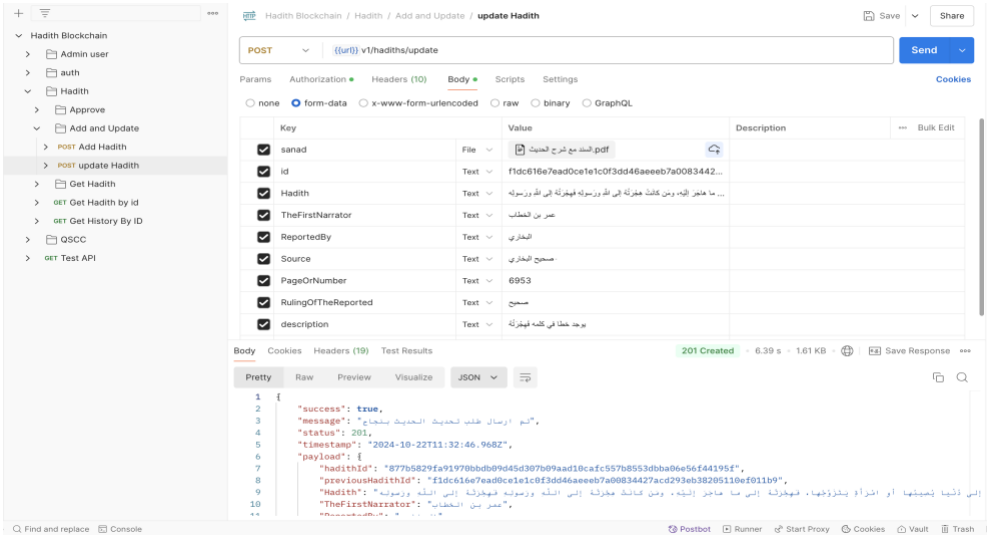

Figure 5.15 Hadith Update Request in Postman

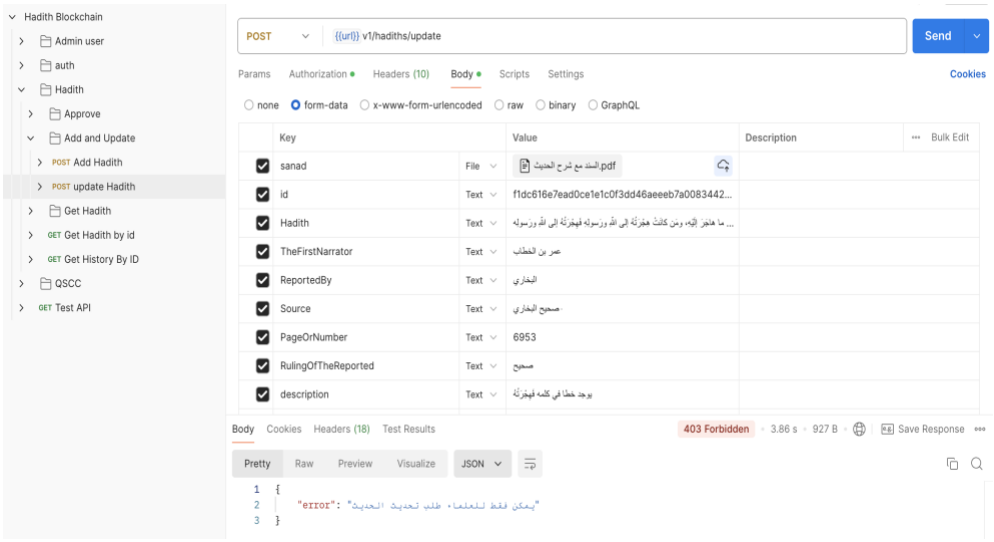

Figure 5.16 Error Response in the Case of Unauthorized User

5.4 Evaluation

This section is mainly about testing the performance of the system.

5.4.1 Experiment Environment Setup

The proposed model uses Hyperledger Fabric v2.5 blockchain and IPFS with Pinata as the IPFS node provider. Permissioned blockchains have facilitated the creation of various benchmarking frameworks, such as Hyperledger Caliper [95], Chainhammer by Kruger[96], Blockbench by Dinh et al.[97], and BCTMark [98]. This research selects Hyperledger Caliper due to its comprehensive documentation and extensive developer community. It is intended to enable users to assess the performance of a particular blockchain network across diverse scenarios. The Hyperledger Caliper service involves creating a workload for a designated system under test (SUT) and persistently monitoring its responses [95]

Hadith data simulation within Hyperledger Caliper was conducted using the datasets from Leeds University and King Saud University (LK) [99]. The LK dataset contains Hadith from renowned books. Nevertheless, as the dataset includes the complete Sanad (series of narrators), it was altered by keeping just the initial narrator. Furthermore, additional fields were added, namely a "Reported By" column and a "Source" column for every Hadith. Table 5.1 discusses the default experiment setup and workload.

Table 5.1 Setup and Workloads.

Parameter	Configuration
Number of Rounds	six
Total Transactions	1000 per rounds
Transaction Rates	50, 100, 150, 200, 250, 300 TPS
State Databas	CouchDB
IPFS Node Provider	Pinata
Blockchain	Hyperledger Fabric v2.5
Endorsement Policy	out of (2, 'Org1MSP.peer', 'Org2MSP.peer', 'Org3MSP.peer', 'Org4MSP.peer')
Ordering Service	Raft
Number of Clients	3
Batch Size	MaxMessageCount: 500 AbsoluteMaxBytes: 10 MB PreferredMaxBytes: 2 MB
Transaction Duration	2s

5.4.2 Performance Evaluation

The network was tested in four scenarios that examined different system behaviors. The first scenario examined how virtual machine (VM) configurations affected network performance. The second scenario explored the impact of increasing the number of institutes and nodes on network performance. The third scenario compared Go and JavaScript chain code performance. Finally, the fourth scenario examined network performance for write, read, and combined transactions. Each scenario offered valuable insights into the variables that influence network scalability.

A. Impact of Virtual Machine Configurations

The network was tested under three system configurations with different vCPUs, cores, and memory. The configurations tested were 4 vCPUs (2 cores), 4 GB RAM; 8 vCPUs (4 cores), 8 GB RAM; and 16 vCPUs (8 cores), 16 GB RAM, all running Ubuntu 20.04. The benchmarking process involved six rounds of Add Hadith, with send rates ranging from 50 to 300 TPS.

As depicted in Figure 5.24, The first configuration with 4 vCPUs limited network throughput and latency as the send rate increased. In Round 1, throughput was 48.6 TPS and peaked at 80.5 TPS in Round 6. Overall latency increased from 1.39 seconds in Round

1 to 1.81 seconds in Round 6, peaking at 1.94 seconds in Round 4. The second configuration with 8 vCPUs improved all metrics. Throughput increased significantly from 48.3 TPS in Round 1 to 222.6 TPS in Round 6. Average latency dropped from 1.32 seconds in Round 1 to 1.64 seconds in Round 6. The third configuration with 16 vCPUs performed best in all metrics. Throughput was the highest, from 48.6 TPS in Round 1 to 232.5 TPS in Round 6. The average latency was 1.27 seconds in Round 1 and 1.23 seconds in Round 6. Analysis of the three system configurations reveals a strong correlation between computational resources and network performance. Increased vCPUs and memory led to notable improvements in throughput and latency, particularly under higher transaction loads. Consequently, higher system configurations enhance network scalability.

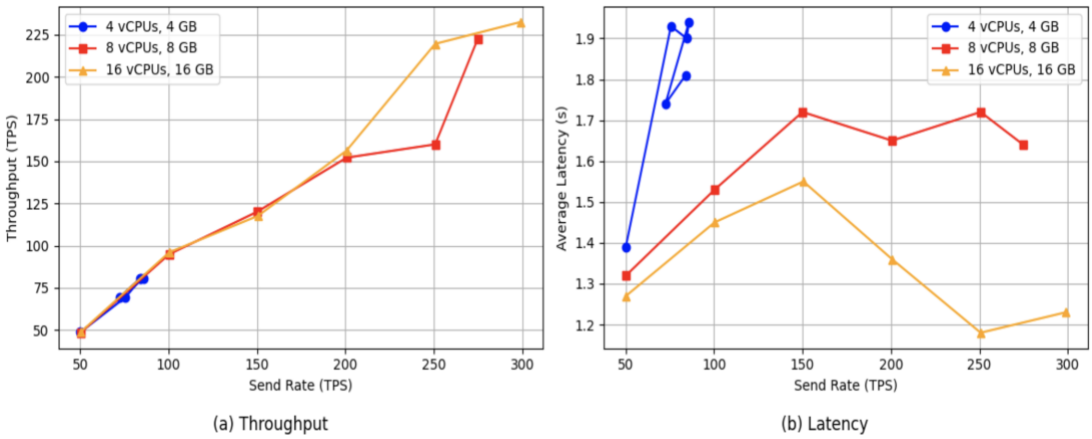

Figure 5.17 Workload Performance Results Across Different VM Configurations

B. Impact of Network Expansion

Within this subsection, network performance was evaluated using four different configurations: two institutes with two nodes, two institutes with four nodes, three institutes with six nodes, and four institutes with eight nodes, all employing three orderers. The AddHadith function was used to run benchmarks with send rates from 50 to 300 TPS. That configuration examined the impact of the Blockchain's size expansion on system performance.

As depicted in Figure 5.25, all configurations saw higher throughput as the send rate increased. The two institutes, two nodes setup had 48.6 TPS in Round 1 and 232.5 TPS in Round 6, with average latency decreasing from 1.27 to 1.23 seconds. Similar results were

observed in other setups, with only marginal latency increases as more nodes were added. The four institutes and eight nodes setup attained the maximum throughput of 234.5 TPS and the most optimal latency performance. The results indicate that the network scales effectively, maintaining high throughput and acceptable latency even as complexity increases.

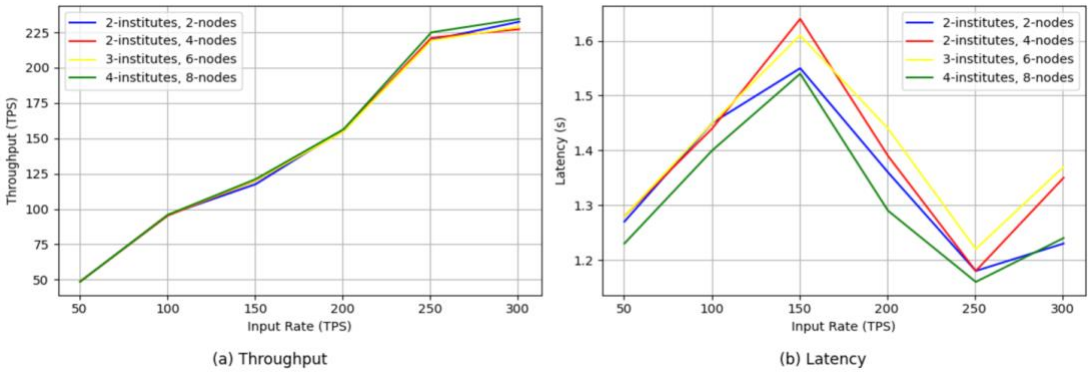

Figure 5.18 Workload Performance Results During Network Expansion

C. Impact of Chaincode Language

Figure 5.26 illustrates Go and JavaScript implementations' throughput and latency performance at varying send rates. Benchmarks were conducted using the AddHadith function. The results reveal that both languages perform similarly up to a send rate of 150 TPS, with only minor differences in throughput. However, beyond this threshold, Go demonstrates a substantial performance advantage. For instance, at a send rate of 300 TPS, Go achieves a throughput of approximately 232.5 TPS, while JavaScript remains slightly lower at around 228.8 TPS.

In contrast, the differences in latency are more pronounced. Go's latency peaks at 150 TPS and drops sharply at 300 TPS. On the other hand, JavaScript's 150 TPS latency is 1.49 seconds, lower than Go's. Go performs better than JavaScript at higher transaction rates, achieving a lower latency of 1.23 seconds at 300 TPS.

These findings suggest that Go outperforms JavaScript in throughput and latency, especially as the system load increases. Since Go and JavaScript chaincode

implementations have been developed for this system, either language could be used during deployment. Nonetheless, given these findings, Go may be more suitable.

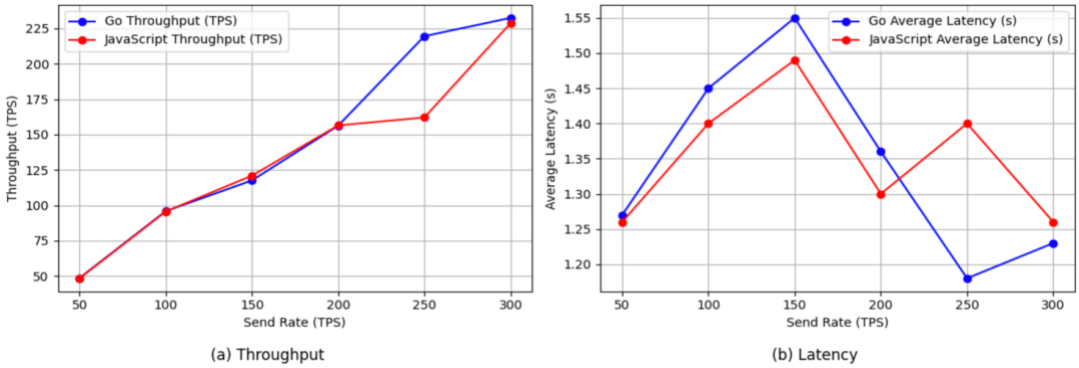

Figure 5.19 Performance Comparison of Go and JavaScript Chaincode

D. Impact of Transaction Types

This subsection tests writing, reading, and combined transaction modes. Transactions per round are 50, 100, 150, 200, 250, and 300 TPS. The maximum threshold for total transactions in each round is set at 1,000.

As evident from Figure 5.27, the initial testing scenario involves assessing writing transactions in which a ledger must be updated with hadith data by calculating transaction latency and throughput. The mathematical formula for writing transaction latency can be expressed as:

$$WT_L = (T_c * N_T) - S_T \quad (1)$$

The variable WT_L represents the duration of the transaction using the hadith network, T_c denotes the time it takes to confirm a transaction, N_T represents the change in the network threshold, as well as S_T represents the time the transaction is submitted. The throughput of transactions may be expressed according to:

$$WT_T = T_{CT} / T_{TS} * N_{CN} \quad (2)$$

WT_T is the writing transactions successfully each second, T_{CT} is the system's committed transaction, and T_{TS} is the completed transactions, which are the failed transactions that subtract the total N_{CN} committed transactions.

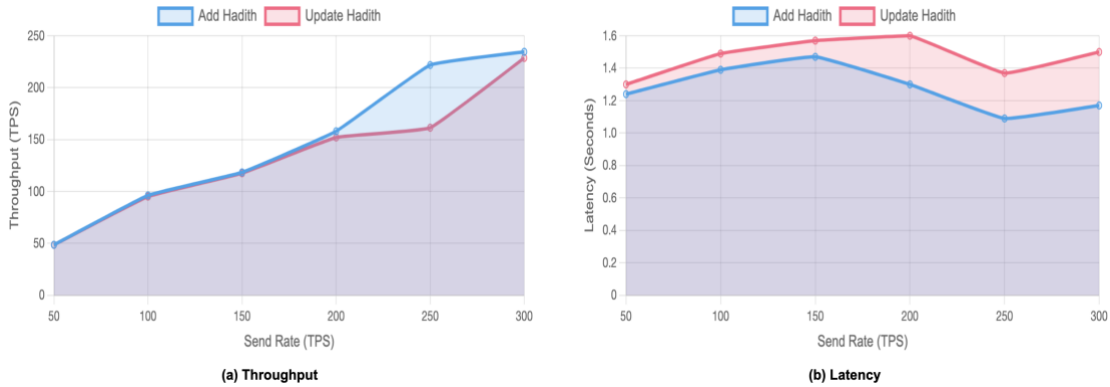

Figure 5.20 Network Performance in Writing Transactions Mode

As shown in Figure 5.27, the throughput of the `AddHadith` function increases with the workload, rising from 48.5 TPS to 234.5 TPS. Under heavier workloads, the average latency starts at 1.23 seconds and increases to 1.24 seconds.

Similarly, the `UpdateHadith` function follows a comparable trend. Its throughput is 48.5 TPS at low workload and 228.6 TPS at high workload. However, the average latency gradually rises from 1.30 seconds to 1.60 seconds with increasing workload.

The `AddHadith` function generally has lower latency and slightly higher throughput than `UpdateHadith`. This variation is because the `UpdateHadith` function needs extra read operations to verify the status of related records. However, both functions scale well, as shown by the continuous increase in throughput with transaction rates, while avoiding high network latency.

The second test scenario calculates transaction latency and throughput for reading or querying transactions. The mathematical formula for calculating the latency of reading transactions is:

$$RT_L = T_R - S_T \quad (3)$$

RT_L represents the latency time for reading transactions, S_T represents the time for submitting transactions, and T_R represents the response time during receipt. Reading

transaction throughput refers to the overall count of query transactions completed within a one-second timeframe and may be expressed according to:

$$RT_T = T_o - T_S \quad (4)$$

Reading transaction throughput (RT_T) is determined by subtracting the reading or query transactions (T_o) from the whole time measured in seconds (T_S).

Figure 5.28 shows that the `GetAllHadiths` function improved throughput from 24.5 TPS at a 50 TPS input rate to 41.3 TPS at a 300 TPS input rate. Despite pagination to manage large data, latency was 20.89 seconds throughout all rounds. This indicates high computational overhead and limited scalability due to the data volume.

However, the ` QueryHadithById ` function, which retrieves individual Hadith records, showed significant scalability. As input rates increased, throughput increased from 50.4 TPS to 231.8 TPS, with a 0.01–1.22 second latency.

The `QueryHistoryById` function, which retrieves the entire update history of a Hadith, performed similarly to `QueryHadithById`, except with slightly lower throughput and slightly higher latency. As input rates increased, throughput increased from 50.1 TPS to 231.6 TPS, with a 0.01–1.27 second latency. The difference in performance is due to the extra effort required to retrieve historical data.

Comparative analysis shows that ID-based queries (`QueryHadithById` and `QueryHistoryById`) outperform `GeAllHadiths` in throughput and latency. Even with pagination, `Get-All-Hadiths` faces scalability issues due to extensive data retrieval.

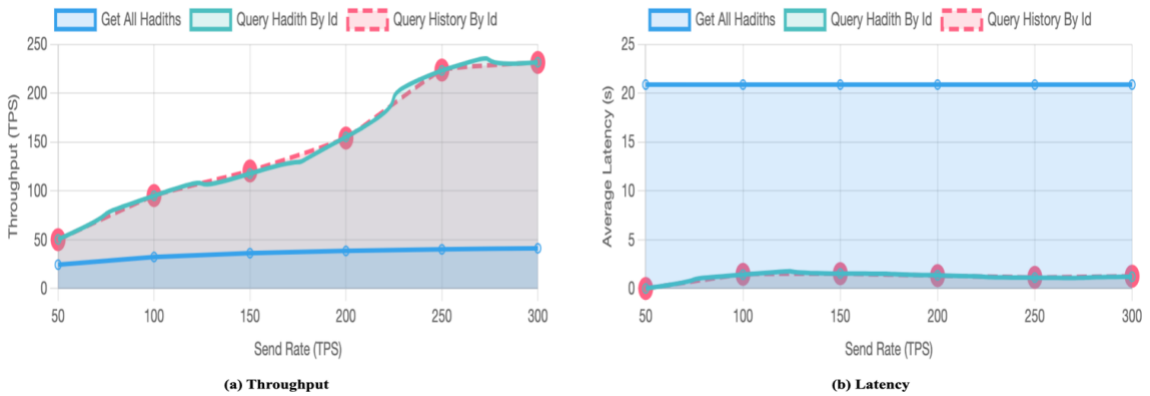

Figure 5.21 Network Performance in Reading Transactions Mode

Our third test scenario measures latency and throughput for combined transactions. This two-step process first checks the hadith's status to ensure that the approval is from different institutions and is still "In progress." The system must meet these conditions to approve and record the transaction, ensuring its validity and preventing duplicates. Figure 5.29 illustrates that latency for `approveAndRejectNewHadith` increases with load, peaking at 1.34 seconds. On the other hand, `approveAndRejectUpdateHadith` has a more stable and lower latency. For throughput, ` approveAndRejectUpdateHadith ` achieves 167.3 TPS, while `approveAndRejectNewHadith` achieves 179.8 TPS. Throughput may be lower in the 'approveAndRejectUpdateHadith' function due to complex conditions, such as deleting previous Hadiths from the world state.

In summary, both functions scale well, as shown by the continuous increase in throughput with transaction rates, while avoiding high network latency.

Figure 5.22 Network Performance in Combined Transactions Mode

Table 5.2 summarizes the performance evaluation.

Table 5.2 Summary of Performance Evaluation

Scenario	Findings
Impact of Virtual Machine Configurations	Higher vCPUs and memory improve throughput and reduce latency. Best performance observed with 16 vCPUs (8 cores), 16 GB RAM, reaching 232.5 TPS and 1.23s latency.
Impact of Network Expansion	Network scales effectively with more institutes and nodes. Maximum throughput (234.5 TPS) achieved with four institutes, eight nodes. Latency remained stable.
Impact of Chaincode Language	Go and JavaScript perform similarly up to 150 TPS. At 300 TPS, Go reaches 232.5 TPS, while JavaScript reaches 228.8 TPS. Go has better latency under high load.

Impact of Transaction Types (Write Transactions)	AddHadith: Throughput increases from 48.5 TPS to 234.5 TPS, latency from 1.23s to 1.24s. UpdateHadith: Throughput increases from 48.5 TPS to 228.6 TPS, latency from 1.30s to 1.60s. Extra reads cause higher latency for UpdateHadith.
Impact of Transaction Types (Read Transactions)	GetAllHadiths: Low throughput (24.5–41.3 TPS), high latency (20.89s). QueryHadithById: Higher throughput (50.4–231.8 TPS), low latency (0.01–1.22s). QueryHistoryById: Similar to QueryHadithById, slightly lower throughput (50.1–231.6 TPS), slightly higher latency (0.01–1.27s). ID-based queries perform better than full dataset retrieval.
Impact of Transaction Types (Combined Transactions)	approveAndRejectNewHadith: Higher latency (peaks at 1.34s), higher throughput (179.8 TPS). approveAndRejectUpdateHadith: More stable, lower latency, but slightly lower throughput (167.3 TPS) due to additional conditions like deleting previous Hadiths. Both functions scale well without excessive latency increases.

5.5 Addressing Hadith System Requirements

Section 3.4.1 compiles a list of system requirements the proposed model must fulfill. We will review these system requirements and explain how the solution meets them.

Decentralization: Hyperledger Fabric and IPFS store data in a distributed manner, resulting in a decentralized system that guarantees no one can manipulate it. An attacker can spread fabricated hadiths by stealing a scholar's private certificate, but that scenario is unrealistic in this system because two scholars must agree to activate a hadith. Furthermore, if a minority of channel institutions commit misconduct, the remaining institutions can vote to expel the disobedient entities from the channel.

Access Control: The system uses multi-layered access control to prevent unauthorized access. This means access control checks are done on every system layer, starting with the backend and then in the Fabric channel chain code. If a user tries to approve a hadith, the system responds, 'You are not authorized to perform this operation'.

Integrity: Blockchain stores information within blocks connected like a chain using the block's hash value. The connected chain is then shared among members via a distributed network. As a result, all the members (nodes) could quickly detect any alteration or manipulation of any transaction. Therefore, this unique characteristic of blockchain eliminates common misconceptions about retaining digital records, such as data substitution, deletion, or insertion. Consequently, even if a transaction is added incorrectly,

it cannot be removed or replaced; instead, a new block must be created containing the correct transaction data.

Furthermore, the participants could view both the newly added transaction and the first incorrect transaction in chronological order, together with the relevant timestamp. This implies that information can be checked anytime, even after being issued for years. IPFS data is uniquely identified. Thus, even a one-character change affects its hash. Hyperledger Fabric will write IPFS's hash value onto its immutable ledger. Based on this approach, the system ensures a high level of integrity.

Availability: Since all Hadith data is distributed across Hyperledger Fabric or IPFS, it is available as long as at least one node is online. This solves the issue of a single point of failure, which could bring the entire system down. Unlike centralized systems, the blockchain's availability does not rely on any entity, which might lead to far greater longevity and persistence.

In addition, there is no downtime during network updates. The Hyperledger Fabric permits institutions within the same channel to propose modifications to the channel parameters. Once the proposal receives sufficient votes, the channel undergoes uninterrupted reconfiguration. These channel updates allow institutions to be added or removed from the channel.

Transparency: The system guarantees high transparency via its blockchain architecture, which permanently logs all transactions and data alterations. Participants have access to detailed information about blockchain blocks containing all the transactions within them. The system also records each Hadith entry's history. Users can see both versions if a Hadith is added with an error and later corrected. They can also see which scholars approved the original and updated Hadith. This level of detail gives users full access to decision-making and changes.

Consequently, this system can offer superior transparency compared to centralized systems. No institution can revoke transparency, censor, or limit viewership participation in a publicly distributed ledger.

Traceability: The system tracks every action, including Hadith record creation, modification, and approval, for maximum traceability. A unique digital signature identifies

the participant responsible for each action. Once committed, no one can change or delete this digital signature, timestamp, or Hadith content on the blockchain.

For instance, the system logs the endorsement and transaction details if a scholar approves a Hadith without following the verification process. These details include the approval time, the Hadith version, and the scholar's digital signature. With this comprehensive log, participants can quickly identify and investigate system rule violations.

Scalability: The blockchain ledger solely retains significant hadith data. Consequently, the utilization of a blockchain ledger is minimized. The performance results presented in Section 5.4.2 demonstrate that the system is scalable, as it avoids low throughput and high network latency.

5.6 Comparison of System

This research compares the proposed system with alternative systems, as illustrated in Table 5.3. The centralized system faces significant limitations, including SPOF, lack of transparency, traceability, data persistence, and data integrity. In contrast, the system from the study [3] and the proposed model use decentralized architectures that eliminate SPOF and provide better robustness and data security through blockchain technology.

The proposed model is characterized by supporting scholars' approval of new and updated hadiths, which none of the other systems offer. It is also characterized by its access control and scalability, making it more comprehensive than the system from the study [3]. In addition, the proposed model provides more advanced operations, such as adding, updating, rejecting, querying a hadith, handling approvals, retrieving all hadith, and retrieving a hadith history. Additionally, it utilizes Hyperledger Fabric v2.5, the latest version at the time of this research's writing.

The proposed model benchmarks throughput and latency using Hyperledger Caliper, while the study [3] only measures average response time. The proposed model also includes a complete performance analysis. It tests the network in four different situations: changing the configurations of the virtual machines, adding more institutions and nodes, comparing the performance of Go and JavaScript chain code, and testing the network for write, read, and combined transactions. This analysis provides insight into the scalability of the network, which needs to be covered in the study [3].

The comparison results demonstrate the thoroughness of the proposed model, highlighting it as the most advanced solution.

Table 5.3 System comparison.

Characteristics	Centralized	[3]	proposed model
Architecture	Client Server	Peer to peer	Peer to peer
Authority	Centralized	Distributed/Decentralized	Distributed/Decentralized
Robustness	SPOF	No SPOF or control	No SPOF or control
Data Storage	Tables	Blocks	Blocks
Approval a New Hadith by Scholars	×	×	✓
Approval Update Hadith by Scholars	×	×	✓
Integrity	×	✓	✓
Access Control	✓	×	✓
Data Persistence	×	✓	✓
Transparency	×	✓	✓
Traceability	×	✓	✓
Scalability	-	Fairly scalable	✓
Functionality	Comprehensive functionality for a wide range of operations	Basic operations include adding, querying, and retrieving a hadith history.	Adding, updating, rejecting, querying a hadith, handling approvals, retrieving all hadith, and retrieving a Hadith history are all included.
Chaincode Support	-	JavaScript	Go and JavaScript
Blockchain	-	Hyperledger Fabric v1.4	Hyperledger Fabric v2.5
Performance Benchmarking	-	Benchmarked with official Apache; measured Average Response Time	Benchmarked with Caliper; measured throughput and latency
Performance Analysis	-	Basic performance analysis	Comprehensive performance analysis

5.7 Summary

This chapter covered the main components, starting with the setup of the Hyperledger Fabric network and the development of chain code. It then showed how system users perform various operations. Experiment results showed that the network scales efficiently. The chapter also discussed how the model meets the requirements of decentralization, access control, data integrity, availability, transparency, traceability, and scalability. Finally, the proposed model was compared with existing solutions to demonstrate its unique advantages.

6 Chapter Six

Conclusion

6.1 Introduction

This chapter summarizes the whole research, focusing on thesis contributions and highlighting directions for further research.

6.2 Thesis Contributions

The following list outlines the main achievements of this thesis:

1. This research leveraged the Hyperledger fabric blockchain to create a secure hadith storage model that addresses the risks of data integrity, SPOF, lack of transparency, and traceability in centralized architectures.
2. This research developed essential functions, such as adding, approving, updating, and rejecting hadith records. It also included querying individual hadith records, all hadith records, and accessing the entire hadith history.
3. The proposed model stored the data of Sanad and Hadith commentaries in IPFS and thus improved scalability.
4. The research has incorporated multi-layered RBAC mechanisms to ensure that only hadith scholars can authenticate and approve hadiths. RBAC has been implemented at every system layer, from the backend to the Fabric channel chaincode.
5. The study investigated how virtual machine configurations affect the performance of the Hyperledger Fabric blockchain network. Increasing the vCPUs and memory improved throughput and latency, especially with high transaction loads.
6. The system implemented chaincode in Go and JavaScript for performance analysis. Both implementations handled increasing transaction loads well, with Go showing slightly higher throughput and lower latency in high-demand scenarios.
7. Research has thoroughly evaluated the model's performance and shown that the system scales effectively with increasing network size and data volume, avoiding high network latency and low throughput.

   Figure 6.1 connects each contribution with the corresponding objectives and the research questions it addresses.

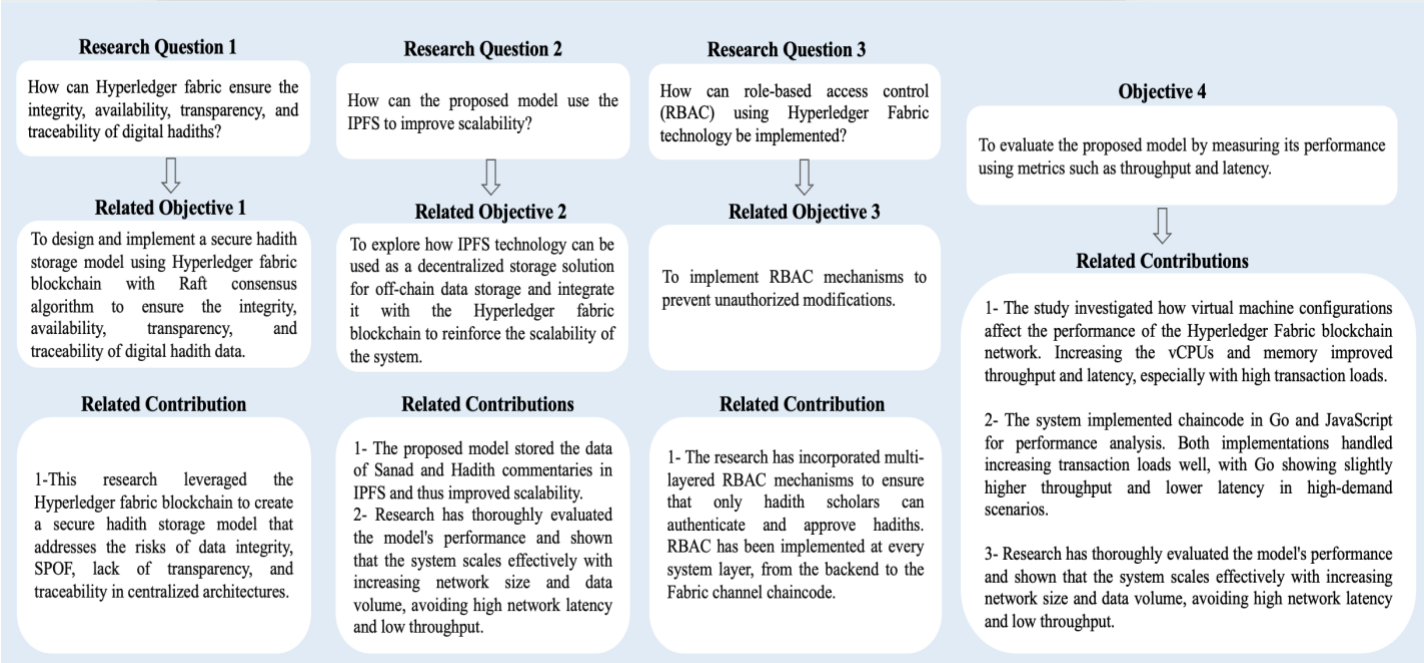

Figure 6.1 Connect Research Questions with Objectives and Contributions

6.3 Conclusion and Future Work

Integrating Hyperledger Fabric and IPFS introduces a new structure for managing Hadith data, ensuring integrity, transparency, traceability, scalability, and availability. In this research, the blockchain ledger solely retains significant hadith data. Consequently, the utilization of a blockchain ledger is minimized. The system enables participants to verify the authenticity of a Hadith. If a Hadith is not found in the blockchain, a student of Hadith can upload a new hadith and request its verification from hadith scholars. When two Hadith scholars from different institutions agree on the authenticity of a Hadith, it becomes active. If one scholar rejects it, the Hadith is removed from the world state before activation.

Additionally, any scholar on the network can submit a request to update a Hadith if an error is found. Compared to alternative blockchain solutions, the solution does not incur transaction fees. Experiments demonstrate that this model's throughput is good and can fulfill the demand. Furthermore, the solution architecture is scalable and effectively accommodates potential growth in institutes, nodes, and Hadith submissions. Its additional considerations regarding the prevention of unauthorized access are presented.

Finally, this structure showcases the broader potential of the Hyperledger Fabric blockchain in fields that require rigorous information authentication, such as managing religious opinions (Fatwas). In such cases, the system ensures that only authenticated scholars can respond to users, and their responses (Fatwas) are securely preserved for future reference, ensuring their authenticity and longevity for future generations.

Future work will focus on a user interface to improve system usability and accessibility. Additionally, artificial intelligence (AI) will be incorporated into the proposed model to expand its functionalities. AI could identify anomalous behaviors among scholars, such as frequent rejections of hadiths, and facilitate the direct submission of hadiths to scholars rather than students. Hadiths submitted by AI or Hadith students could be stored in MongoDB Cloud to optimize blockchain ledger use. Before uploading Hadiths to the blockchain, scholars could review and authenticate them in this temporary storage. This approach makes it easier for scholars to add new Hadiths while reducing the blockchain ledger load.

7 References

[1] S. Hakak *et al.*, "Digital Hadith authentication: Recent advances, open challenges, and future directions," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 6, Jun. 2022, doi: 10.1002/ett.3977.

[2] F. Haque, A. H. Orthy, and S. Siddique, "Hadith Authenticity Prediction using Sentiment Analysis and Machine Learning," in *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, IEEE, Oct. 2020, pp. 1–6. doi: 10.1109/AICT50176.2020.9368569.

[3] K. M. Awad, M. ElNainay, M. Abdeen, M. Torki, O. Saif, and E. Nabil, "A Secure Blockchain Framework for Storing Historical Text: A Case Study of the Holy Hadith," *Computers*, vol. 11, no. 3, p. 42, Mar. 2022, doi: 10.3390/computers11030042.

[4] K. Gaanoun and M. Alsuhaibani, "Fabricated Hadith Detection: A Novel Matn-Based Approach With Transformer Language Models," *IEEE Access*, vol. 10, pp. 113330–113342, 2022, doi: 10.1109/ACCESS.2022.3217457.

[5] H. F. Atlam, A. Alenezi, M. O. Alassafi, and G. B. Wills, "Blockchain with Internet of Things: Benefits, Challenges, and Future Directions," *International Journal of Intelligent Systems and Applications*, vol. 10, no. 6, pp. 40–48, Jun. 2018, doi: 10.5815/ijisa.2018.06.05.

[6] R. A. Abutaleb, S. S. Alqahtany, and T. A. Syed, "Integrity and Privacy-Aware, Patient-Centric Health Record Access Control Framework Using a Blockchain," *Applied Sciences*, vol. 13, no. 2, p. 1028, Jan. 2023, doi: 10.3390/app13021028.

[7] S. Kumar, A. K. Bharti, and R. Amin, "Decentralized secure storage of medical records using Blockchain and IPFS : A comparative analysis with future directions," *SECURITY AND PRIVACY*, vol. 4, no. 5, Sep. 2021, doi: 10.1002/spy2.162.

[8] L. Zhang, W. Zeng, Z. Jin, Y. Su, and H. Chen, "A Research on Traceability Technology of Agricultural Products Supply Chain Based on Blockchain and IPFS," *Security and Communication Networks*, vol. 2021, pp. 1–12, Nov. 2021, doi: 10.1155/2021/3298514.

[9] S. Chenthara, K. Ahmed, H. Wang, F. Whittaker, and Z. Chen, "Healthchain: A novel framework on privacy preservation of electronic health records using blockchain technology," *PLoS One*, vol. 15, no. 12, p. e0243043, Dec. 2020, doi: 10.1371/journal.pone.0243043.

[10] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy," *J Parallel Distrib Comput*, vol. 164, pp. 152–167, Jun. 2022, doi: 10.1016/j.jpdc.2022.03.009.

[11] M. Mahmud, Md. S. H. Sohan, S. Reno, M. A. B. Sikder, and F. S. Hossain, "Advancements in scalability of blockchain infrastructure through IPFS and dual blockchain methodology," *J Supercomput*, vol. 80, no. 6, pp. 8383–8405, Apr. 2024, doi: 10.1007/s11227-023-05734-x.

[12] A. Ayub Khan, A. A. Laghari, A. A. Shaikh, S. Bourouis, A. M. Mamlouk, and H. Alshazly, "Educational Blockchain: A Secure Degree Attestation and Verification Traceability Architecture for Higher Education Commission," *Applied Sciences*, vol. 11, no. 22, p. 10917, Nov. 2021, doi: 10.3390/app112210917.

[13] K. Awuson-David, T. Al-Hadhrami, M. Alazab, N. Shah, and A. Shalaginov, "BCFL logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem," *Future Generation Computer Systems*, vol. 122, pp. 1–13, Sep. 2021, doi: 10.1016/j.future.2021.03.001.
[14] C. Denis González, D. Frias Mena, A. Massó Muñoz, O. Rojas, and G. Sosa-Gómez, "Electronic Voting System Using an Enterprise Blockchain," *Applied Sciences*, vol. 12, no. 2, p. 531, Jan. 2022, doi: 10.3390/app12020531.
[15] Z. Touati-Hamad, M. R. Laouar, and I. Bendib, "Towards Blockchain-Based Document Authentication," *International Journal of Organizational and Collective Intelligence*, vol. 12, no. 3, pp. 1–15, Aug. 2022, doi: 10.4018/IJOCI.306693.
[16] H. Abubakar and S. Hassan, "A framework for enhancing digital trust of Quranic text using Blockchain technology," *Journal of Telecommunication, Electronic and Computer Engineering*, 2018.
[17] L. Ismail and H. Materwala, "A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions," *Symmetry (Basel)*, vol. 11, no. 10, p. 1198, Sep. 2019, doi: 10.3390/sym11101198.
[18] S. Wan, M. Li, G. Liu, and C. Wang, "Recent advances in consensus protocols for blockchain: a survey," *Wireless Networks*, vol. 26, no. 8, pp. 5579–5593, Nov. 2020, doi: 10.1007/s11276-019-02195-0.
[19] Brown and Jonathan, *Hadith: Muhammad's legacy in the medieval and modern world*. Simon and Schuster, 2017.
[20] F. Binbeshr, A. Kamsin, and M. Mohammed, "A Systematic Review on Hadith Authentication and Classification Methods," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 2, pp. 1–17, Mar. 2021, doi: 10.1145/3434236.
[21] M. A. Saloot, N. Idris, R. Mahmud, S. Ja'afar, D. Thorleuchter, and A. Gani, "Hadith data mining and classification: a comparative analysis," *Artif Intell Rev*, vol. 46, no. 1, pp. 113–128, Jun. 2016, doi: 10.1007/s10462-016-9458-x.
[22] A. M. Azmi, A. O. Al-Qabbany, and A. Hussain, "Computational and natural language processing based studies of hadith literature: a survey," *Artif Intell Rev*, vol. 52, no. 2, pp. 1369–1414, Aug. 2019, doi: 10.1007/s10462-019-09692-w.
[23] H. M. Abdelaal, A. M. Ahmed, W. Ghribi, and H. A. Youness Alansary, "Knowledge Discovery in the Hadith According to the Reliability and Memory of the Reporters Using Machine Learning Techniques," *IEEE Access*, vol. 7, pp. 157741–157755, 2019, doi: 10.1109/ACCESS.2019.2944118.
[24] J. Blecher, *Hadith commentary*. Oxford University Press, 2016.
[25] I. Yaqoob, K. Salah, R. Jayaraman, and Y. Al-Hammadi, "Blockchain for healthcare data management: opportunities, challenges, and future recommendations," *Neural Comput Appl*, vol. 34, no. 14, pp. 11475–11490, Jul. 2022, doi: 10.1007/s00521-020-05519-w.
[26] S. Hakak, A. Kamsin, O. Tayan, Mohd. Y. Idna Idris, A. Gani, and S. Zerdoumi, "Preserving Content Integrity of Digital Holy Quran: Survey and Open Challenges," *IEEE Access*, vol. 5, pp. 7305–7325, 2017, doi: 10.1109/ACCESS.2017.2682109.
[27] U. Bodkhe *et al.*, "Blockchain for Industry 4.0: A Comprehensive Review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020, doi: 10.1109/ACCESS.2020.2988579.
[28] P. Hegde and P. K. R. Maddikunta, "Amalgamation of Blockchain with resource-constrained IoT devices for healthcare applications – State of art, challenges and future

directions," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 220–239, Jun. 2023, doi: 10.1016/j.ijcce.2023.06.002.
[29] A. Raja Santhi and P. Muthuswamy, "Influence of Blockchain Technology in Manufacturing Supply Chain and Logistics," *Logistics*, vol. 6, no. 1, p. 15, Feb. 2022, doi: 10.3390/logistics6010015.
[30] G. Habib, S. Sharma, S. Ibrahim, I. Ahmad, S. Qureshi, and M. Ishfaq, "Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing," *Future Internet*, vol. 14, no. 11, p. 341, Nov. 2022, doi: 10.3390/fi14110341.
[31] G. Chen, B. Xu, M. Lu, and N.-S. Chen, "Exploring blockchain technology and its potential applications for education," *Smart Learning Environments*, vol. 5, no. 1, p. 1, Dec. 2018, doi: 10.1186/s40561-017-0050-x.
[32] Gartner, "The CIO's Guide to Blockchain," Gartner. Accessed: Oct. 11, 2024. [Online]. Available: https://www.gartner.com/smarterwithgartner/the-cios-guide-to-blockchain
[33] Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Satoshi Nakamoto, 2008.
[34] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," in *Advances in Cryptology-CRYPT0' 90*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 437–455. doi: 10.1007/3-540-38424-3_32.
[35] A. J, D. P. Isravel, K. M. Sagayam, B. Bhushan, Y. Sei, and J. Eunice, "Blockchain for healthcare systems: Architecture, security challenges, trends and future directions," *Journal of Network and Computer Applications*, vol. 215, p. 103633, Jun. 2023, doi: 10.1016/j.jnca.2023.103633.
[36] P. K. Ghosh, A. Chakraborty, M. Hasan, K. Rashid, and A. H. Siddique, "Blockchain Application in Healthcare Systems: A Review," *Systems*, vol. 11, no. 1, p. 38, Jan. 2023, doi: 10.3390/systems11010038.
[37] B. Shrimali and H. B. Patel, "Blockchain state-of-the-art: architecture, use cases, consensus, challenges and opportunities," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6793–6807, Oct. 2022, doi: 10.1016/j.jksuci.2021.08.005.
[38] A. Raja Santhi and P. Muthuswamy, "Influence of Blockchain Technology in Manufacturing Supply Chain and Logistics," *Logistics*, vol. 6, no. 1, p. 15, Feb. 2022, doi: 10.3390/logistics6010015.
[39] L. Ismail, H. Hameed, M. AlShamsi, M. AlHammadi, and N. AlDhanhani, "Towards a blockchain deployment at uae university: Performance evaluation and blockchain taxonomy," in *Proceedings of the 2019 International Conference on Blockchain Technology*, New York, NY, USA: ACM, Mar. 2019, pp. 30–38. doi: 10.1145/3320154.3320156.
[40] R. Schollmeier, "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications," in *Proceedings First International Conference on Peer-to-Peer Computing*, IEEE Comput. Soc, pp. 101–102. doi: 10.1109/P2P.2001.990434.
[41] B. Lashkari and P. Musilek, "A Comprehensive Review of Blockchain Consensus Mechanisms," *IEEE Access*, vol. 9, pp. 43620–43652, 2021, doi: 10.1109/ACCESS.2021.3065880.

[42] A. Raja Santhi and P. Muthuswamy, “Influence of Blockchain Technology in Manufacturing Supply Chain and Logistics,” *Logistics*, vol. 6, no. 1, p. 15, Feb. 2022, doi: 10.3390/logistics6010015.
[43] D. Burkhardt, M. Werling, and H. Lasi, “Distributed Ledger,” in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, IEEE, Jun. 2018, pp. 1–9. doi: 10.1109/ICE.2018.8436299.
[44] G. Habib, S. Sharma, S. Ibrahim, I. Ahmad, S. Qureshi, and M. Ishfaq, “Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing,” *Future Internet*, vol. 14, no. 11, p. 341, Nov. 2022, doi: 10.3390/fi14110341.
[45] M. Suvitha and R. Subha, “A Survey on Smart Contract Platforms and Features,” in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, Mar. 2021, pp. 1536–1539. doi: 10.1109/ICACCS51430.2021.9441970.
[46] R. Alajlan, N. Alhumam, and M. Frikha, “Cybersecurity for Blockchain-Based IoT Systems: A Review,” *Applied Sciences*, vol. 13, no. 13, p. 7432, Jun. 2023, doi: 10.3390/app13137432.
[47] A. G. Gad, D. T. Mosa, L. Abualigah, and A. A. Abohany, “Emerging Trends in Blockchain Technology and Applications: A Review and Outlook,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6719–6742, Oct. 2022, doi: 10.1016/j.jksuci.2022.03.007.
[48] A. H. Mohsin *et al.*, “Based Medical Systems for Patient’s Authentication: Towards a New Verification Secure Framework Using CIA Standard,” *J Med Syst*, vol. 43, no. 7, p. 192, Jul. 2019, doi: 10.1007/s10916-019-1264-y.
[49] P. Moriggl, P. M. Asprion, and B. Schneider, “Blockchain Technologies Towards Data Privacy—Hyperledger Sawtooth as Unit of Analysis,” 2021, pp. 299–313. doi: 10.1007/978-3-030-48332-6_20.
[50] M. N. M. Bhutta *et al.*, “A Survey on Blockchain Technology: Evolution, Architecture and Security,” *IEEE Access*, vol. 9, pp. 61048–61073, 2021, doi: 10.1109/ACCESS.2021.3072849.
[51] H. Yuan, S. Fei, and Z. Yan, “Technologies of blockchain interoperability: a survey,” *Digital Communications and Networks*, Aug. 2023, doi: 10.1016/j.dcan.2023.07.008.
[52] A. J, D. P. Isravel, K. M. Sagayam, B. Bhushan, Y. Sei, and J. Eunice, “Blockchain for healthcare systems: Architecture, security challenges, trends and future directions,” *Journal of Network and Computer Applications*, vol. 215, p. 103633, Jun. 2023, doi: 10.1016/j.jnca.2023.103633.
[53] A. Nedaković, A. Hasselgren, K. Kralevska, and D. Gligoroski, “Hyperledger fabric platform for healthcare trust relations—Proof-of-Concept,” *Blockchain: Research and Applications*, vol. 4, no. 4, p. 100156, Dec. 2023, doi: 10.1016/j.bcra.2023.100156.
[54] M. S. Rahman, M. A. Islam, M. A. Uddin, and G. Stea, “A survey of blockchain-based IoT eHealthcare: Applications, research issues, and challenges,” *Internet of Things*, vol. 19, p. 100551, Aug. 2022, doi: 10.1016/j.iot.2022.100551.
[55] R. Lai and D. LEE Kuo Chuen, “Blockchain – From Public to Private,” in *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2*, Elsevier, 2018, pp. 145–177. doi: 10.1016/B978-0-12-812282-2.00007-3.

[56] G. Lv, C. Song, P. Xu, Z. Qi, H. Song, and Y. Liu, “Blockchain-Based Traceability for Agricultural Products: A Systematic Literature Review,” *Agriculture*, vol. 13, no. 9, p. 1757, Sep. 2023, doi: 10.3390/agriculture13091757.
[57] W. Gavin, “Ethereum: A secure decentralised generalised transaction ledger,” in *Ethereum project yellow paper*, 2014, pp. 1–32.
[58] G. Lv, C. Song, P. Xu, Z. Qi, H. Song, and Y. Liu, “Blockchain-Based Traceability for Agricultural Products: A Systematic Literature Review,” *Agriculture*, vol. 13, no. 9, p. 1757, Sep. 2023, doi: 10.3390/agriculture13091757.
[59] C. Mauro, M. Schunter, and I. Askoxylakis, *Trust and Trustworthy Computing: 8th International Conference, TRUST 2015, Heraklion, Greece, August 24-26, 2015, Proceedings* . Springer, 2015.
[60] E. Androulaki *et al.*, “Hyperledger fabric: a distributed operating system for permissioned blockchains,” in *Proceedings of the Thirteenth EuroSys Conference*, New York, NY, USA: ACM, Apr. 2018, pp. 1–15. doi: 10.1145/3190508.3190538.
[61] R. Brown, J. Carlyle, I. Grigg, and M. Hearn, “Corda: an introduction,” in *R3 CEV, August*, Aug. 2016.
[62] C.Cachin, “Architecture of the Hyperledger Blockchain Fabric,” in *Workshop on distributed cryptocurrencies and consensus ledgers*, Jul. 2016, pp. 1–4.
[63] E. Androulaki *et al.*, “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” in *Proceedings of the 13th EuroSys Conference, EuroSys 2018*, Association for Computing Machinery, Inc, Apr. 2018. doi: 10.1145/3190508.3190538.
[64] P. Thakkar, S. Nathan, and B. Viswanathan, “Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform,” in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, IEEE, Sep. 2018, pp. 264–276. doi: 10.1109/MASCOTS.2018.00034.
[65] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” Jul. 2014, [Online]. Available: http://arxiv.org/abs/1407.3561
[66] L. Li, D. Jin, T. Zhang, and N. Li, “A Secure, Reliable and Low-Cost Distributed Storage Scheme Based on Blockchain and IPFS for Firefighting IoT Data,” *IEEE Access*, vol. 11, pp. 97318–97330, 2023, doi: 10.1109/ACCESS.2023.3311712.
[67] M. Lusetti, L. Salsi, and A. Dallatana, “A blockchain based solution for the custody of digital files in forensic medicine,” *Forensic Science International: Digital Investigation*, vol. 35, p. 301017, Dec. 2020, doi: 10.1016/j.fsidi.2020.301017.
[68] C. Stamatellis, P. Papadopoulos, N. Pitropakis, S. Katsikas, and W. Buchanan, “A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric,” *Sensors*, vol. 20, no. 22, p. 6587, Nov. 2020, doi: 10.3390/s20226587.
[69] A. Nedaković, A. Hasselgren, K. Kralevska, and D. Gligoroski, “Hyperledger fabric platform for healthcare trust relations—Proof-of-Concept,” *Blockchain: Research and Applications*, vol. 4, no. 4, p. 100156, Dec. 2023, doi: 10.1016/j.bcra.2023.100156.
[70] S. Sutradhar, S. Karforma, R. Bose, S. Roy, S. Djebali, and D. Bhattacharyya, “Enhancing identity and access management using Hyperledger Fabric and OAuth 2.0: A block-chain-based approach for security and scalability for healthcare industry,” *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 49–67, 2024, doi: 10.1016/j.iotcps.2023.07.004.

[71] P. Rani, R. K. Sachan, and S. Kukreja, “Educert-chain: a secure and notarized educational certificate authentication and verification system using permissioned blockchain,” *Cluster Comput*, vol. 27, no. 7, pp. 10169–10196, Oct. 2024, doi: 10.1007/s10586-024-04469-5.
[72] Abu-Bakar and M. Muhammad, “Shariah analysis of bitcoin, cryptocurrency, and blockchain,” *Shariah Analysis in Light of Fatwas and Scholars’ Opinions*, pp. 14–19, 2018.
[73] A. A. Alidin, A. A. A. Ali-Wosabi, and Z. Yusoff, “Overview of Blockchain Implementation on Islamic Finance: Saadiqin Experience,” in *2018 Cyber Resilience Conference (CRC)*, IEEE, Nov. 2018, pp. 1–2. doi: 10.1109/CR.2018.8626822.
[74] N. Khan, B. Kchouri, N. A. Yatoo, Z. Kräussl, A. Patel, and R. State, “Tokenization of sukuk: Ethereum case study,” *Global Finance Journal*, vol. 51, p. 100539, Feb. 2022, doi: 10.1016/j.gfj.2020.100539.
[75] S. Khan, “A Blockchain based Decentralized Zakat Collection and Distribution Platform,” in *Proceedings of the 2023 7th International Conference on Software and e-Business*, New York, NY, USA: ACM, Dec. 2023, pp. 9–13. doi: 10.1145/3641067.3641071.
[76] bin Khatiman, M. Aqmal, M. bin Ismail, and N. Yahya, “ Blockchain-based Zakat collection to overcome the trust issues of Zakat payers.,” *International Journal on Perceptive and Cognitive Computing*, pp. 53–58, 2021.
[77] M. Tieman, M. R. Darun, Y. Fernando, and A. B. Ngah, “Utilizing Blockchain Technology to Enhance Halal Integrity: The Perspectives of Halal Certification Bodies,” 2019, pp. 119–128. doi: 10.1007/978-3-030-23381-5_9.
[78] A. Alamsyah, N. Hakim, and R. Hendayani, “Blockchain-Based Traceability System to Support the Indonesian Halal Supply Chain Ecosystem,” *Economies*, vol. 10, no. 6, p. 134, Jun. 2022, doi: 10.3390/economies10060134.
[79] Munawar and A. Mugiono, “Framework for smart contract blockchain in halal traceability, integrity, and transparency,” *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 2875–2884, Jun. 2024, doi: 10.11591/ijece.v14i3.pp2875-2884.
[80] I. Surjandari, H. Yusuf, E. Laoh, and R. Maulida, “Designing a Permissioned Blockchain Network for the Halal Industry using Hyperledger Fabric with multiple channels and the raft consensus mechanism,” *J Big Data*, vol. 8, no. 1, p. 10, Dec. 2021, doi: 10.1186/s40537-020-00405-7.
[81] B. Sulistio, A. Ramadhan, E. Abdurachman, M. Zarlis, and A. Trisetyarso, “The utilization of machine learning on studying Hadith in Islam: A systematic literature review,” *Educ Inf Technol (Dordr)*, vol. 29, no. 5, pp. 5381–5419, Apr. 2024, doi: 10.1007/s10639-023-12008-9.
[82] M. Castro and B. Liskov, “Practical Byzantine Fault Tolerance,” *OsDI*, vol. 99, pp. 173–186, Feb. 1999.
[83] M. Hearn and R. G. Brown, “Corda: A distributed ledger,” Corda Technical White Paper, Aug. 2016. [Online]. Available: https://corda.net/wp-content/uploads/2019/08/corda-technical-whitepaper-August-29-2019.pdf
[84] ConsenSys, “Quorum whitepaper.” Accessed: Oct. 02, 2024. [Online]. Available: https://github.com/Consensys/quorum-docs/blob/master/Quorum%20Whitepaper%20v0.2.pdf
[85] D. Ongaro and J. Ousterhout, “In Search of an Understandable Consensus Algorithm,” 2014 USENIX annual technical conference, 2014, pp. 305–319.

[86] N. Z. Tomić, "A Review of Consensus Protocols in Permissioned Blockchains," *Journal of Computer Science Research*, vol. 3, no. 2, pp. 19–26, Apr. 2021, doi: 10.30564/jcsr.v3i2.2921.
[87] E. Daniel and F. Tschorsch, "IPFS and Friends: A Qualitative Comparison of Next Generation Peer-to-Peer Data Networks," *IEEE Communications Surveys & Tutorials*, vol. 24, pp. 31–52, Jan. 2022.
[88] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-Based, Decentralized Access Control for IPFS," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, Jul. 2018, pp. 1499–1506. doi: 10.1109/Cybermatics_2018.2018.00253.
[89] A. Ismail, M. Toohey, Y. C. Lee, Z. Dong, and A. Y. Zomaya, "Cost and Performance Analysis on Decentralized File Systems for Blockchain-Based Applications: State-of-the-Art Report," in *2022 IEEE International Conference on Blockchain (Blockchain)*, IEEE, Aug. 2022, pp. 230–237. doi: 10.1109/Blockchain55522.2022.00039.
[90] T. V. Doan, Y. Psaras, J. Ott, and V. Bajpai, "Toward Decentralized Cloud Storage With IPFS: Opportunities, Challenges, and Future Considerations," *IEEE Internet Comput*, vol. 26, no. 6, pp. 7–15, Nov. 2022, doi: 10.1109/MIC.2022.3209804.
[91] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, "Performance Analysis of Private Blockchain Platforms in Varying Workloads," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, Jul. 2017, pp. 1–6. doi: 10.1109/ICCCN.2017.8038517.
[92] E. Androulaki *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, New York, NY, USA: ACM, Apr. 2018, pp. 1–15. doi: 10.1145/3190508.3190538.
[93] G. Yang, K. Lee, K. Lee, Y. Yoo, H. Lee, and C. Yoo, "Resource Analysis of Blockchain Consensus Algorithms in Hyperledger Fabric," *IEEE Access*, vol. 10, pp. 74902–74920, 2022, doi: 10.1109/ACCESS.2022.3190979.
[94] T. Hepp, M. Sharinghousen, P. Ehret, A. Schoenhals, and B. Gipp, "On-chain vs. off-chain storage for supply- and blockchain integration," *it - Information Technology*, vol. 60, no. 5–6, pp. 283–291, Dec. 2018, doi: 10.1515/itit-2018-0019.
[95] Hyperledger, "Hyperledger caliper." Accessed: Oct. 02, 2024. [Online]. Available: https://hyperledger.github.io/caliper/
[96] A. Krueger, "chainhammer." Accessed: Oct. 02, 2024. [Online]. Available: https://github.com/drandreaskrueger/chainhammer
[97] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "BLOCKBENCH," in *Proceedings of the 2017 ACM International Conference on Management of Data*, New York, NY, USA: ACM, May 2017, pp. 1085–1100. doi: 10.1145/3035918.3064033.
[98] D. Saingre, T. Ledoux, and J.-M. Menaud, "BCTMark: a Framework for Benchmarking Blockchain Technologies," in *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, Nov. 2020, pp. 1–8. doi: 10.1109/AICCSA50499.2020.9316536.
[99] S. Altammami, E. Atwell, and A. Alsalka, "The Arabic-English Parallel Corpus of Authentic Hadith," 2020, doi: 10.5518/480.

Appendix

The source code for the implementations and the testing can be found at GitHub

https://github.com/BashayerAlkalifah/DigitalHadiths-blockchain

The Codebase: