Medical Devices Digital Threads and their Supply-Chain Management on Blockchain

by

Kaushal Sanjay Mhalgi

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2021 by the
Graduate Supervisory Committee:

Dragan Boscovic, Co-Chair
Kasim Selcuk Candan, Co-Chair
Maria Grando, Member

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

Blockchain technology is defined as a decentralized, distributed ledger recording the origin of a digital asset and all of its updates without the need of any governing authority. In Supply-Chain Management, Blockchain can be used very effectively, leading to a more open and reliable supply chain. In recent years, different companies have begun to use blockchain to build blockchain-based supply chain solutions. Blockchain has been shown to help provide improved transparency across the supply chain.

This research focuses on the supply chain management of medical devices and supplies using blockchain technology. These devices are manufactured by the authorized device manufacturers and are supplied to the different healthcare institutions on their demand. This entire process becomes vulnerable as there is no track of individual product once it gets shipped till it gets used. Traceability of medical devices in this scenario is hardly efficient and not trustworthy.

To address this issue, the paper presents a blockchain-based solution to maintain the supply chain of medical devices. The solution provides a distributed environment that can track various medical treatments from production to use. The finished product is stored in the blockchain through its digital thread. Required details are added from time to time which records the entire virtual life-cycle of the medical device forming the digital thread. This digital thread adds traceability to the existing supply chain. Keeping track of devices also helps in returning the expired devices to the manufacturer for its recycling.

This blockchain-based solution is mainly composed of two phases. Blockchain-based solution design, this involves the design of the blockchain network architecture, which constitutes the required smart contract. This phase is implemented using the secure network of Hyperledger Fabric (HLF). The next phase includes the deployment

of the generated network over the Kubernetes to make the system scalable and more available.

To demonstrate and evaluate the performance matrix, a prototype solution of the designed platform is implemented and deployed on the Kubernetes. Finally, this research concludes with the benefits and shortcomings of the solution with future scope to make this platform perform better in all aspects.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Dragan Boscovic for the continuous work to support my thesis study and related research. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Master's study. He has not only given me experience with research but also helped me expose myself to Industry and always pushed me for excellence.

I would like to thank Dr. Kasim Selcuk Candan and Dr. Adela Grando for their unwavering support and guiding me throughout my thesis work. Dr. Adela's experience in the Healthcare industry helped this project tremendously.

I would like to thank Chainrider team for helping through the Hyperledger Fabric setup. I would like to thank Google Cloud for providing resources on Google Cloud Platform that supported deployment and testing blockchain systems.

I would like to thank all members of the Blockchain Research Lab at ASU: Raj Sadaye, Manish Vishnoi, Sasa Pesic, Francis Mendoza, Arnav Dhiman for their contribution in discussions regarding consortium blockchains. I would like to thank Basam Alasaly, Jason Brynt for guiding me through the industrial supply chain management of the medical devices. I would especially like to thank Raj and Manish for their mentorship in Hyperledger Fabric and Kubernetes.

Finally, I must express my very profound gratitude to my parents for providing me the unfailing support and continuous encouragement throughout my years of study and through the process of researching. This accomplishment would have not been possible without them.

Thank you.

Kaushal Sanjay Mhalgi

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

A blockchain can be defined as a decentralized, distributed ledger, shared across multiple users to record immutable transactions. These stored records are called blocks and they are linked through a strong cryptographic algorithm such that any involved block can not be altered or tampered without alteration of subsequent blocks. The blockchain database is maintained using a peer-to-peer network that eliminates a need for a centralized governing authority to manage the network.

The first usage of blockchain technology example known is Bitcoin cryptocurrency. Satoshi Nakamoto (a person or group of people), published a paper describing a peer-to-peer network to serve as a public transaction ledger of the cryptocurrency Bitcoin.

Over the years, blockchain technology has been evolved to provide a secured platform to a wide range of applications. Started from cryptocurrencies, blockchain technology is being used in various applications like social media, supply chain management, payment systems.

Considering the industrial, economical, and logistical advantages of blockchain technology, it is expected to show great potential in the field of Supply Chain Management. Blockchain will drive improved transparency in the supply chain to help minimize fraud on high-value products as well as on critical medical devices.[27]

Medical devices are a vital part of the healthcare industry. Whether delivered to pharmacies, hospitals, or other healthcare professionals, supply chains for medical devices are necessary to ensure that goods arrive safely and securely. Manufacturers need to work with logistics providers who understand and have in-depth experience in this field, with a growing number of these items needing specialist handling.

1.1 Problem Statement

Since 1990, with the widespread use of Enterprise Resource Planning(ERP) systems, a great improvement in the exchange of supply chain knowledge has taken place. However, in large and sensitive supply chains involving complex transactions, visibility remains an issue.[14] Current issues in the field of supply chains may cause a disruption in the entire system and such issues should not occur when the data in the supply chain is sensitive such as medical devices.

Before moving towards the problems in the existing supply chain system, let's understand how ERP works to maintain the supply chain. ERP is a comprehensive framework for transaction management that combines multiple types of information processing capabilities and stores all information in a single environment.[5] For a general transaction involving a product manufacturer and a bank, there are three important flows: Information flow, inventory flow, and financial flow. The three flows can not be connected reliably by current ERP systems, manual audits, and inspections, making it difficult to avoid execution mistakes, enhance decision-making, and address supply chain disputes.[14]

The underlying infrastructure running the current supply chains isn't sustainable for the complex and large volumes of transactions and data integrity has become the primary issue in the system. Execution errors, such as product data errors and missing shipments, are also not detectable in real-time. Even though all types of flows are recorded by ERP systems, it can be difficult to determine which journal entries correspond to which inventory transaction. This issue can be found on a bigger scale for companies engaged in a lot of transactions each day across a large network of supply chains and products. These situations can be worsened by extremely complicated supply chain activities. For example, orders and shipments can not be neatly

coordinated, resulting in an order separated into different shipments and invoices, or multiple orders merged into a single shipment. Due to the lack of traceability, it is also difficult to identify who is at fault in case of data loss.

The safety and security of medical devices (including implanted cardiac pacemakers and medication pumps) are extremely important, given that devices can be compromised and controlled. Also, there is a need to reduce response times when recalls and field notices are issued by device manufactures and regulators. Blockchain-enabled distributed registries offer the potential to address those issues.

One of the most important issues dealing with the supply chain management of the medical devices and healthcare industry is the lack of universal standards for data interchange. There is no solution in the existing system, to track the product separated from its package. Although the FDA plans to fully secure electronic product tracing to facilitate a step-by-step supply chain for medical devices, some of the basics of quality control are still at an early age.[9]

Product manufacturers, suppliers, and healthcare institutions are concerned about the privacy of data as well as membership of the network. Hence consortium blockchain solutions are an ideal choice for implementing a supply chain application for 2 main features:

1. Data Permissioning through Access Control.

2. Selective Membership to pre-approved participants.

1.2 Issues in the current ERP-based Systems

ERP systems have been used in the supply chain management industry for a long period of time and they should be credited to bring substantial change in the supply chain field. However, in recent times due to the growing industries and huge

transaction counts occurring on the daily basis, these ERP systems lack to provide the following features which are important to maintain trust and accountability in the supply chain of the medical devices.

- **Scalability:** Scalability relates to the capacity of the software to adapt to evolving workloads and business development. While in most ERP definitions and guides, scalability is only briefly touched upon, it is one of the most important components of the supply chain. In today's situations as the supply chain industry of medical devices is growing at a fast rate, ERP systems may fail to scale to the level where they can provide the best performance. With growing industries and higher transaction rates, the ERP system reaches to end of its resources sooner than expected. Due to this lack of scalability feature, ERP systems in the supply chain of medical devices are unable to provide real-time and trustworthy service.

- **Flexibility:** ERP systems are designed to provide a centralized enterprise data repository that provides users with access to information in accordance with the processes they complete. This is achieved by building a comprehensive database that can house and organize data, distributing it to specific ERP applications that can be accessed by users. However, to make these ERP systems work fluent and flawless requires deep integration between backend data workflows. These problems arise from inherent shortcomings in flexibility.[21]

  Overall, flexibility in the ERP system and business adaptability are interconnected among each other. Locking into a legacy ERP suite to update the system with current requirements, may result in compromised flexibility.

- **Execution Errors:** With the growing industries and more organizations in the supply chain, ERP systems are vulnerable to make execution errors. These

execution errors can be related to the mistakes in the inventory data, shipment details data, missing shipments, or related to payments. ERP systems in the present times are unable to detect these errors in real-time. It is difficult and expensive in ERP systems to detect the origin of the mistake even after the mistake is discovered. These execution mistakes are more noticeable where industries involved in the supply chain are large and the transaction count is high. ERP systems are also unreliable when the data is sensitive such as healthcare data or the data about medical devices.[14]

- **Real-Time Auditability:** In case of execution mistakes such as inaccurate data about inventory or shipments, missing shipments, or payments in ERP systems, there are multiple approaches to improve the supply chain. One of the common approaches is to verify the transactions through auditing. In order to ensure compliance with contracts, auditing is important, but it is of limited help in enhancing decision-making to resolve organizational shortcomings. The reason behind this limitation is the lack of real-time transaction tracking and auditing. ERP system introduces additional latency in the supply chain that makes the system unreliable.

- **Security:** Data security has always been the major concern of ERP systems. in the ERP systems, transaction data is stored in the centralized database. Common ERP security risks are the following[11]:

  - **Outdated Data:** An open invitation to security nightmares like viruses, hackers, and malware is to run old, obsolete versions of your ERP program. Updating the ERP system is a way to patch the existing security concerns. However, updating the entire ERP system is an expensive and time-consuming process. Outdated ERP systems can lead to all sorts of

ugly data entry mistakes. In the case of an important medical device supply chain, inaccurate data added by the wrong or outdated ERP system version may cause a disturbance in the system.

  - **Poor Reporting Capabilities:** ERP systems lack data reporting functionalities which cause difficulty for users to access and analyze the data within their system. In such situations, most of the time users tend to use some other data recording tools which produce scattered data in the system. This causes the loss of traceability of all devices whose data is scattered.

  - **Unrestricted Data Access:** The important aspect to secure the data is to provide the access to only those who are authorized. For example, the sales representatives should not be authorized to the inventory data access. Current ERP systems, while capturing all transactional flows, are unable to manage and control the data access leading to the comprised privacy of the data.

- **Traceability of devices:** Traceability of the device not only means the ability of the manufacturer to trace a product through its delivery to the customer but also to have the ability to retrace a product backward through its manufacturing process. A traceability system can be important if a product is recalled, has failed after use, or if other development issues are found either before or after the product enters the supply chain. Although some ERP systems accommodate restricted types of traceability, it is difficult for an ERP system to satisfy a particular traceability requirement out of the box due to the variable data attributes that need to be collected (based on consumer demand, product type, and industry).[8] In the case of medical devices, the journey of the product not

only stops at its delivery to the healthcare institution but it might need trace-back in case of the products return to the manufacturer in case of its expiry. In such scenarios, ERP systems are incapable to trace the product or prove trust in traced data.

1.3 Why a blockchain solution for the supply chain management of medical devices

Blockchain technology offers security and traceability benefits in the supply chain management of medical devices that seemed to be lacking in the traditional ERP systems. The 2013 U.S. Drug Supply Chain Security Act allows pharmaceutical firms to recognize prescription drugs and trace them in order to protect customers from counterfeit, stolen, or dangerous goods.[14] Hence, blockchain technology where the digital thread of the product is stored from the manufacture of the medical device to its use in surgery or if expired, returned to the manufacturer through a shared ledger becomes the possible solution. The blockchain helps the company and its supply chain partners to track the product, identify all vendors involved with it, identify manufacturing and shipping batches associated with it, and effectively recall it if a company finds a defective product.[14] The use of the blockchain brings trust to the system. As the transactions in the blockchain are immutable, tampering of the records in the blockchain is prohibited and the security of the data is established.

Verifying transactions by audits is one common approach to improving the execution of the supply chain. This strategy, however, is of little help in strengthening decision-making to overcome organizational shortcomings. Another solution to reducing execution errors and enhancing supply chain traceability would be to label inventory with GS1 compliant RFID tags or electronic product codes (globally agreed guidelines for the handling of supply chain data) and then to combine the ERP systems of a company with those of its suppliers to establish a full transaction record.

However, this approach can introduce the following drawbacks:

- Integrating ERP systems are expensive.
- These systems are time-consuming.
- There is a lack of global standards in medical devices and the healthcare industry, hence the traceability of individual devices is difficult to achieve.

With the consortium blockchain in Hyperledger Fabric and using digital threads of the product, each device will have a Unique Device Identification(UDI) number for tracking. This enables each device to be tracked at all times in the supply chain. Sharing of information with the other participants in the network having adequate authorization can be implemented by deploying smart-contracts with sufficient endorsement policies. Consortium blockchain also eliminates the need for the governing party as the endorsement policies defined in the network enforce the transaction once they are validated. Hence, the consortium blockchain-based solution follows the principle of 'trust, but verify'

1.4 Hypothesis and Scope of Enabling Research

With Hyperledger Fabric as an underlying infrastructure, it is possible to build a distributed solution to a supply chain management of medical devices that can offer visibility and traceability. Apart from the consortium blockchain, our system is using the Digital Thread of the medical device to store the required product details(including manufacturing details and FDA details) and product shipment details(retrieved using the APIs of the logistic companies). The Digital Thread of the product is encrypted using a strong encryption algorithm and stored on the blockchain. Hence the data is not readily available and access control is implemented.

Our solution is deployed on Kubernetes that provides important features such as the automated recovery of the nodes and efficient load-balancing. These features help the system to be more available and robust on the top of which blockchain network is deployed. Our system can co-operate with the existing system to achieve privacy, confidentiality, and better performance. Two use-cases that are proposed and implemented:

- Submitting order, shipment tracking, and product tracking
- Returning of product in case of expiry

The solution is implemented to provide a solution to the above-mentioned use-cases using the common hyperledger fabric architecture and both test-cases are tested. The tests involved different characteristics of the HLF network to find the optimal configuration. Further, this configuration is used to compare the performance of both models in terms of response times for smart contracts. The proposed system is tested considering the following aspects:

- **Scalability:** With the growing healthcare industry and an increasing number of transactions, it is important to have a system that can be scaled to support the growing industry while maintaining the throughput of the system. We will be testing the scalability of the system by configuring the system with an increasing count of peers. Throughput of the blockchain network is checked with a higher number of peers to check how blockchain networks perform on large scale.
- **Load Balancing of the network:** Medical device supply chain may face a sudden high volume of transactions. For example, during the COVID-19 pandemic, the demand for respiratory masks and PPE kits increased suddenly.

The network should perform well during this scenario. Our network is deployed on the Kubernetes and hence the load balancing is checked by increasing the volume of transactions. It is further compared against the traditional docker deployment.

- **Size of the information stored in the blockchain:** The size of the information getting stored on the blockchain in a single transaction is an important parameter to test the network. As the data associated with the medical device may vary the size of information also varies. Our system is tested with the different block sizes to check the ideal size of information that should be stored on the blockchain in a single transaction.

1.5 Organization of the Thesis

The content of this thesis is organized as follows:

1. **Chapter 2:** Focuses on the background and related work in the supply chain field
2. **Chapter 3:** Describes the solution for two use-cases using Hyperledger Fabric and Kubernetes
3. **Chapter 4:** contains the Experimental Details. It contains information for experimental setup, data set, evaluation criteria, and performance of the system.
4. **Chapter 5:** concludes the thesis and describes the direction for future work.

Chapter 2

RELATED WORK AND RESEARCHES

2.1 The Impact of ERP on Supply Chain Management

Enterprise Resource Planning(ERP) is a transaction management that cooperates with other data processing abilities in the supply chain and stores the entire data in a single database.[5]

ERP systems have improved the supply chain industry to its maximum potential. However, the exponential growth and complex activities in supply chain fields limits the privacy and traceability features of the ERP.

Major benefits of the ERP systems:

- A transaction processing engine, allowing for the integrated management of data throughout the enterprise;
- ERP captures all three different types of flows, which helps to make the supply chain robust.
- ERP Systems allow the administration of an organization's resources in an integrated manner by automating most of the functions, to try to make the information available in real-time.[7]

Although, ERP systems have revolutionized the supply chain industry, they fail to solve the following issues:

- **Execution Errors:** When the scale of the supply chain gets sufficiently large, ERP systems tend to make mistakes that can be tracked and taken care of in real-time.

- **Privacy and Confidentiality:** ERP systems don't explicitly focus on the privacy of the data and hence records are vulnerable when shared.

- **Lack of traceability:** ERP does not achieve end-to-end tracking of the products in the supply chain. This lack of visibility may cause accountability issues in the supply chain.

- **High risk of failure:** ERP systems fail to incorporate process changes, upgrading software products, and adding new functionality, making ERP systems more vulnerable and inefficient.

Our solution provides a blockchain-based approach and uses an approach of digital thread that guarantees the end-to-end tracking of the product and thus provides traceability and visibility in the supply chain. As the solution is deployed using Kubernetes, issues like availability, and scalability of the system are also solved. Hence, rather than providing a replacement for ERP, we present a blockchain-based innovative solution to curb ERP's challenges for data privacy, sharing and integrity.

2.2 Supply Chains of Medical Devices

Medical devices, like most sophisticated consumer and industrial equipment, are the products of supply chains that can sometimes be lengthy, with multiple tiers of suppliers contributing to the final product. Quality issues can and do occur at each step of the process and with every link in the supply chain. But the buck stops with the device maker, who is ultimately held responsible for the finished product and its performance.

The lack of global standards for data interchange, work processes, and control capabilities is one of the most critical problems that the healthcare industry is facing. For example, while bar codes are now ubiquitous in consumer goods, unique identifiers

are not universal in the healthcare business. In response, to promote a step-by-step supply chain for drug products, the FDA plans to fully protect electronic product tracing. But in medical device systems, some of the basics of quality control are still at an early stage.[9]

Yet, amid those difficulties, there has never been a greater need to strengthen the working of supply chain relationships. The increasing network of healthcare facilities around the world makes it imperative to collaborate. Supply chain participants need to see each other as partners in a highly competitive environment in a joint effort to bring the right product to the right customer in the right market, right when it's needed.

To implement the system better, it needs a technological infrastructure that allows useful information to be exchanged frictionlessly in real time. It involves linking details to make end-to-end visibility possible between buyer and seller. It requires breaking down silos, using the same metrics, applying the same systems of performance management, and involving the same team of related workers from both partners.[23] Apart from that, it requires standardizing business processes on a frequent basis. It includes setting targets that are practical, sharing a feasible plan, and having the patience to achieve their goals.[25]

2.3 EtherTwin: Blockchain-based Secure Digital Twin Information Management

EtherTwin[24] is a ethereum-based decentralized application for secure information management using digital twins. EtherTwin offers secure information management, ensuring confidentiality through fine-grained access control and encryption, as well as providing integrity and availability based on the blockchain. EtherTwin eliminates the issue of information sharing of the Digital Twin data in the traditional system with the introduction of blockchain platforms and smart contracts.

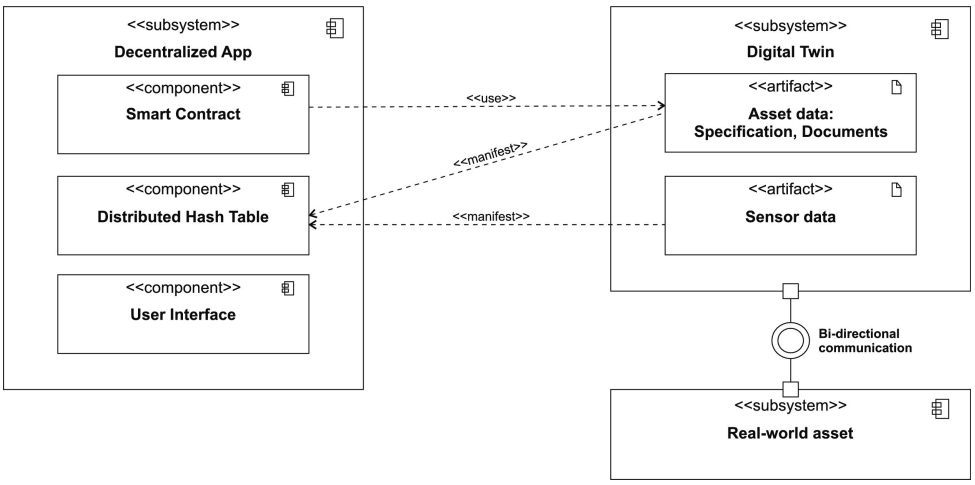


Figure 2.1: Component Diagram Describing the Digital Twin Sharing Context

The issues with EtherTwin involves the use of Ethereum blockchain that is a public blockchain that works on the principle of the proof-of-work. The proof-of-work consensus algorithm is time-consuming, computationally expensive, and vulnerable to 51% attack. Apart from this, EtherTwin only provides the solution of the digital twin data sharing rather than focusing on the digital thread of the product.

2.4 Leveraging Blockchain Technology to Enhance Supply Chain Management in Healthcare

In this paper, the author proposes a blockchain-based solution to manage various supply chain operations of medical devices. The benefits of this system include reducing or eliminating the frauds/errors, improving inventory management, and rapid problem detection. The paper also talks about the participation of different parties in the supply chain and trusted data sharing. The paper mentions, Unique Device Identifier(UDI) as the identifier of medical devices, that can be shared across the blockchain network due to its efficiencies and accountability around trust.[10]

However, the system was just an idea to explore the opportunities in the supply

chain management of the medical devices and hence, system specifications aren't mentioned. Also, need to use the HLF consortium blockchain to enhance the trust in the network.

2.5 Comparison of Proposed Solution with State of Art Systems

In this chapter, we reviewed multiple systems using decentralized blockchain technology in supply chain management to eliminate execution errors and improve the transparency of the system. Following are the similarities between these systems and the proposed solution:

- Blockchain technology is used for the security, confidentiality, and transparency of transactions.
- Digital twin is stored on the blockchain in encrypted format to share the data with other participants.

But apart from these similarities, the following are the innovative features provided in the proposed solution that were lacking in the state of the art systems:

- The proposed solution computes and maintains the digital thread of the medical devices that are stored on the blockchain to track the details of the device over its complete life-span. All details of the product such as its manufacturing, shipment, usage are recorded in encrypted format on the blockchain.
- The proposed solution is deployed on the Kubernetes which ensures the auto-recovery and security of the deployment nodes. Hence, the Kubernetes layer of deployment makes the system more available and scalable to use in the larger ecosystem.

Chapter 3

DESIGN AND METHODOLOGY

3.1 Types of Blockchain

Since the establishment of the blockchain technology, various advancement has given rise to the new generation of the blockchain to incorporate the newer challenges like Digital Identity Management, data governance, privacy and confidentiality, etc.[26] Following are the four types of blockchains:

- **Public Blockchain:** A public blockchain is a non-restrictive, permissionless distributed ledger system.[26] Anyone can be the part of these public networks as a node or a peer of the network. Once joined a peer can perform various operations such as accessing the records, append the block of transactions using proof-of-work(PoW) or mining, etc. Public blockchains are still considered safe as the blocks are immutable, that is tampering of data is avoided using the PoW consensus algorithm. Common applications of the public blockchain include the implementation of cryptocurrencies. The most common examples of the public blockchains are Bitcoin, Ethereum, Dash, etc.

- **Private Blockchain:** A private blockchain is a restricted or the permissioned blockchain, in most cases, designed for a particular organization or the company.[26] Private blockchains make use of different access-control methods such as participant certificates in order to make sure that only authorized members are allowed to be part of the network and thus only they can access the records, validate the transactions. Hence, private blockchains are similar to the

public blockchains in terms of operations and usability but deployed on a much smaller scale (Considering a particular organization or a company in mind). In most scenarios, private blockchains are deployed for asset management, supply chain management, digital identity management etc.

- **Consortium Blockchain:** A consortium blockchain is a blockchain that involves multiple organizations or companies as participants of the network to provide the broader level solution. It can be described as the semi-decentralized blockchain as some organizations may have a higher authority than the others. In a consortium blockchain, organizations can be divided according to their role or functions and each organization is responsible to maintain their participants along with their identities. Consortium blockchains are usually used by government organizations, banks etc.

- **Hybrid Blockchain:** A hybrid blockchain as its name suggests, is a combination of both public and private blockchain. Using a hybrid blockchain, it is possible to deploy the public decentralized and still maintain the access control over the specific data similar to the private blockchain. A hybrid offers the features of both public and private blockchains that is one can have a private permissioned system along with the public permissionless system in the same network. A hybrid also offers the flexibility to its nodes as a node can be a participant of one private blockchain along with multiple public blockchains. The transaction in a private blockchain of the hybrid blockchain is validated in the private blockchain to maintain its privacy however a user can release it in the public blockchain for its speedy validation. The public blockchain in the hybrid blockchain has the more number of nodes and hence it increases the hashing and speed of the transaction validation.

3.2 Architectural Components of Blockchain

Following are the core components of the blockchain architecture:

- **Node:** A node, sometimes also referred to as a peer is a computer or a machine in the blockchain network which holds the independent copy of the blockchain ledger.
- **Transaction:** A transaction can be defined as the fundamental building block of the blockchain. It stores all kinds of data or information in the blockchain that can not be tampered with or upgraded.
- **Block:** A block is a set of transactions arranged in a particular manner and distributed to all participating nodes in the blockchain network.
- **Chain:** A chain is a sequence of blocks in a particular order.
- **Miners/Validators:** Specific nodes that perform the operation of block verification before adding them to the existing blockchain.
- **Consensus (consensus protocol):** A set of rules or policies to carry out the functionality of the blockchain.

3.3 Hyperledger Fabric

Hyperledger Fabric (HLF) is an open-source distributed ledger technology (DLT) system that has been licensed at the enterprise level and is optimized for use in enterprise scenarios. It has a number of key advantages over other blockchain platforms. Unlike other blockchain platforms, HLF is a permissioned blockchain that means participants of the network are known to each other and hence there is trust

in the network. This network works under the governance model which ensures that participant of the network follows the policies defined by the network.[6]

Smart contracts written in general programming languages such as Java, Go and Javascript are supported by HLF, which means that no additional language is needed to learn how to write smart contracts. HLF offers various features such as privacy and confidentiality, pluggable consensus, and scalability. HLF has a highly scalable and configurable architecture that enables innovation, versatility, and optimization across a broad range of industries, including banking, finance, insurance, healthcare, human resources, supply chain, and even digital music distribution.

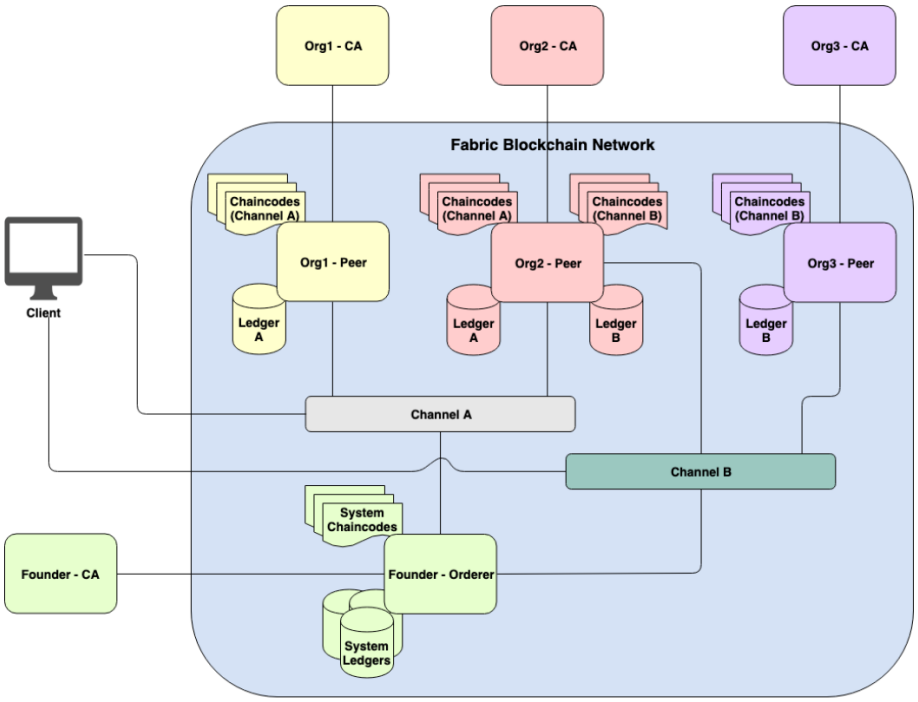


Figure 3.1: Hyperledger Fabric Architecture

Hyperledger Fabric introduces a new approach to verify the transactions and append the block into the chain that is 'execute-order-validate'. This approach divides the transaction flow into the following three steps:

1. **Execute** a transaction, verify its correctness followed by its correctness.

2. **Order** transactions using a consensus algorithm

3. **Validate** transactions against an endorsement policy defined in the network before appending the block to the ledger.

This approach eliminates the issues faced by the 'order-execute' model such as durability, flexibility, efficiency, and confidentiality.[22]

Architectural component of HLF and their specifications in the system:

- **Organizations:** An organization, also known as 'members' is a set of users of the blockchain network which is added to the network by adding its 'Membership Service Provider'. Every organization manages its users and hence, is responsible for them. An organization consists of peers, users, and clients.

  Organizational details of the systems are discussed in section 3.9.

- **Certification Authority**: A certification authority(CA) issues certificates to all users of the blockchain network. These certificates are X.509 certificates, digitally signed by the certificate authority. HLF has a built-in CA known as 'Fabric-CA' and it is up to an organization to have a CA or not. In the blockchain network, CA is hosted by a separate node that does the operations of the certification authority.

  Our system uses CA to maintain the confidentiality of the medical devices and their transactions. Whenever a device details need to be shared with a particular healthcare institution, its digital identity(certificate issued by the CA) can be added to the shared list, and hence the device details can be shared confidentially.

- **Membership Service Provider(MSP):** The MSP identifies which Root CAs and Intermediate CAs are agreed to establish the members of a trust domain by listing the identities of their members, or by identifying which CAs are allowed to issue valid identities for their members.

  An MSP define unique roles within the organization that an actor can perform (e.g. admin or a sales representative or a Order Reception staff-member) and set access rights in the network and channel context.

- **Peers:** In HLF, nodes are called peers. These peers can invoke a transaction and also maintain the state and copy of the ledger. Endorsing peers (also known as endorsers) defined in the network have the authority to endorse a transaction which is then checked against the endorsement policy defined for each smart contract for its validation.

  In this system, number of peers determines the scalability of the system. For this system, number of peers are initially kept four for each organization and then increased gradually to test the scalability of the system.

- **Channels:** A HLF channel is a private ledger shared between two or more specific organizations to maintain the privacy and confidentiality of the transactions. A channel in HLF is itself a separate blockchain on which only authorized peers can invoke or query the ledger. A channel is defined by its organizations, anchor peers of those organizations, the shared ledger, smart contracts deployed on it, and the ordering service node(s). A transaction is invoked or queried by the authorized peers on a particular channel.

  In our application, we create different channels to segregate data according to its usage. For e.g, the Digital thread of a particular medical device should be

stored on a separate ledger where the order information is stored. Different endorsement policies can be executed for different channels.

- **Chaincodes:** In the HLF network, a smart contract is called as a chaincode. It is an asset specifying business logic and the instructions for the transaction to change the asset. The rules for reading or modifying key-value pairs or other information from the state database are enforced by the chaincode. The chaincode must be mounted on the required peers in order to perform a transaction of the chaincode from those peers. Chaincode transactions once validated, are inserted into the ledger. In HLF, chaincode can be programmed using various languages such as Java, Go, Javascript.

- **Client:** Clients are the nodes invoking and broadcasting transaction proposals to the ordering service. These nodes are connected to the peer of its choice and used by the end-user through and front-end application.

- **Endorsement Policies:** Endorsement policies define the condition or agreement for a transaction to be valid. Endorsement policy can widely vary based on the requirements, like a peer from an organization can endorse a complete transaction or all organizations may need to endorse for valid transactions. When the ordering service sends the transaction to them for confirmation, each committing peer can check if the endorsements in the transaction comply with the endorsement policy. If this is not the case, the transaction is invalidated, and it has no effect on the world state.

- **Private Data Collections:** HLF offers the feature of Private Data Collections, which allows the further privacy of the data to endorse, commit, or query the private without creating a separate channel for it. In our systems, we have used

the private data collections to share the data of the medical device with only required entities. For eg. The device details will be available for the device manufacturer and shared with the healthcare institution that has purchased it.

- **Ordering Service**: Unlike permissionless blockchains such as Ethereum and Bitcoin, where the system relies on probabilistic consensus which eventually guarantees the ledger consistency but is still vulnerable to the forks(divergent ledgers). This problem is eliminated in the HLF with the use of a deterministic consensus algorithm. HLF proposes a separate node called orderer which forms the ordering service that ensures if a block is validated by the peer, it is guaranteed to be final and correct and hence will be appended to the blockchain. There are mainly three types of ordering services: Solo, Kafka, and Raft. This solution is implemented using the RAFT ordering service.

  Raft is a "crash fault-tolerant" (CFT) ordering service that employs a "leader and follower" model, in which the channel's leader is dynamically elected from the ordering nodes (known as the "consenter set"). The system will bear the failure of the ordering node, as long as most ordering nodes remain active(including the leader node).

3.3.1 Transaction flow in Hyperledger Fabric

Figure 3.2 represents the transaction flow in HLF. There are four main phases in HLF transaction flow, which are as the following:

1. **Transaction Proposal:** A transaction proposal is submitted by the client to execute an operation on the blockchain network. The transaction proposal has the client-ID, transaction payload, timestamp of the proposal, and the digital signature of the client. This transaction is submitted to one or more endorsing

peers for endorsement.

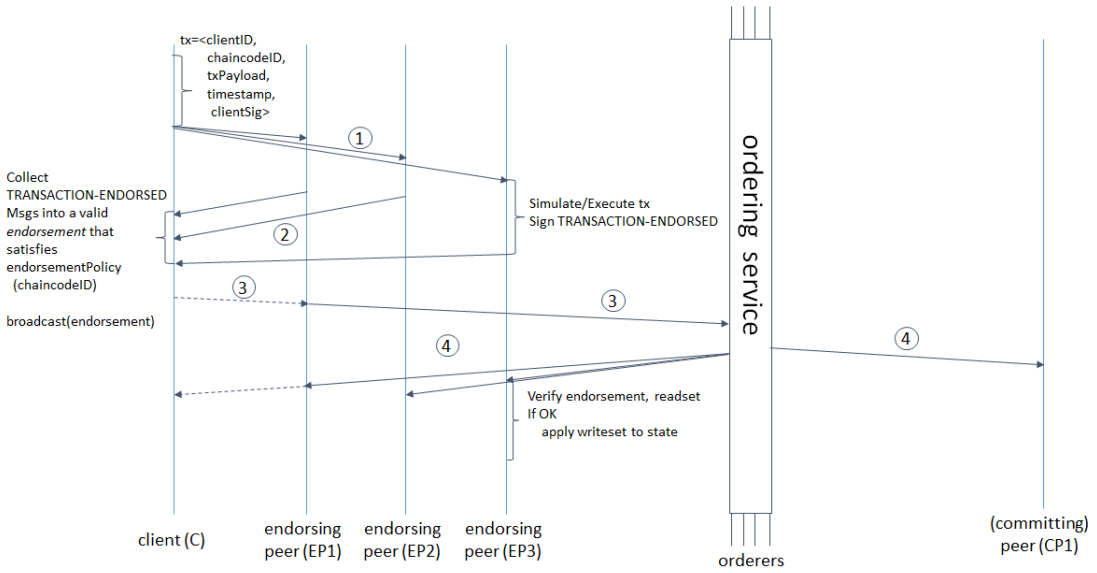


Figure 3.2: Transaction Flow in Hyperledger Fabric[6]

2. **Simulate and Sign:** Once the transaction proposal is submitted to the endorsing peers, endorsing peer signs this transaction with its digital key. Every peer has a state and copy of the ledger that is synchronized with other copies of the ledger. This copy is used for the transaction simulation and by the end of the simulation, a read/write set is generated which involves the updated key-value pairs. The response generated in this phase is the endorsement response that holds the cryptographic information of the peer, the transition, and the read/write set of the transaction. By the end of this phase, the local copy of the ledger is updated.

3. **Invocation Request:** An invocation request is a bundle of all endorsement responses received by the endorsing peers and further sent to the ordering service. The ordering service is responsible for the verification of all cryptographic material and checking if all endorsement policies are fulfilled that are mentioned in

the rules of the network. If policies are not satisfied, the request will be rejected and no changes are made to the shared ledger. However, such transactions are stored on the blockchain to avoid such attacks in the future.

If all endorsing policies are satisfied, however, there is a mismatch in the read/write set of invocation request and other key-value pairs, the invocation request is rejected as the peer invoking the request is out of sync. Otherwise, if all requests are validated and also the corresponding read/write sets are matched, the ordering service does the task of sending the transaction to all of the participant peers of the channel.

4. **Commit Phase:** It is the final phase in the transaction flow. In this phase, all peers of the channel update their copy of the ledger. In case of transaction flooding, the ordering system is responsible to order all the transactions in the correct way such that a divergent ledger is avoided.

3.4 Digital Twin

Digital twin is the virtual representation of the living or non-living entities. It is used as a collection of all the data related to the particular product. In recent times, the digital twin has been redefined as the following:
"A digital twin is a digital replica of a living or non-living physical entity. Data is shared seamlessly by bridging the actual and virtual domains, enabling the virtual entity to operate simultaneously with the physical entity."[18]

Digital twin consists of the following three stages:

1. The physical product

2. The digital product

3. The connection between the physical and digital products that is the flow of data that keeps the virtual product up-to-date.

Since the establishment of the idea of the digital twin, this prototype is divided into the following three types[16]:

i **Digital Twin Prototype (DTP):** The DTP is made up of designs, studies, and the physical product realization process. DTP occurs prior to the individual goods.

ii **Digital Twin Instance (DTI):** The DTI is the digital twin of the product generated after it's manufactured. This digital twin holds the entire data of the product.

iii **Digital Twin Aggregate (DTA):** The DTA is a set of DTIs with data and knowledge that can be used for physical product interrogation, forecasting, and learning.

3.5 Digital Thread

Digital thread refers to the digitization and traceability of the product over its entire lifetime. The digital thread links all the different capabilities in the digital twin back to the component designs, specifications, and software that goes into the digital twin's product.[15]

The digital thread and digital twin together provide as-designed specifications, validation and calibration records, as-built data, as-flown data, and as-maintained data.[20]

3.5.1 Digital Thread in Supply Chain Network

In external supply chains, multiple issues bring down the efficiency of the network such as the lack of the timely sharing of the data, lack of trust and accountability between different participants of the network.

These challenges can be addressed by a digital thread, which provides a continuous stream of data in a time-to-time manner and allows all participants to get deep insights into the chain. Digital thread also enhances the visibility and traceability in the supply chain as it allows to make the data visible across the network over the complete lifespan of the product.

In the case of medical devices, the digital twin and thread of the may contain information that is sensitive to be kept open in the public. Hence, there are various access control methods that can be used to provide privacy and confidentiality to the data. These digital twins can also be encrypted with secure AES-256 or other strong encryption algorithms to provide security to the data, although the digital thread is shared on the supply chain network.

3.6 Computing Digital Twin and Digital Thread

A Digital twin of the medical device is computed by collecting all required data and encrypted using the secured AES-256 encryption function. The encryption allows the digital twin to be shared with multiple participating entities of the network.

A Digital twin of the medical device is created and stored in the blockchain at the time of the device manufacturer. Hence, the digital twin is containing the following information:

- **Device Details:** These are the details of the medical product lot which gets assigned to it as soon as it manufactures. These details are assigned by the

device manufacturer. For example,

  - Device Number
  - Lot Number
  - Device Description
  - Unit Of Measurement(UOM)
  - Expiration Date
  - Device Specification (eg. Stent Diameter, Catheter Diameter, Catheter Length in case of VascularStent)

- **FDA Details:** These are the details specified by the FDA to manufacture the products. These FDA details are defined before the device lot is manufactured and must be followed by the manufacturers.

  - Version/Model
  - Primary DI Number
  - Public Device Record Key
  - MRI Required(Yes/Optional)

These above-mentioned details are collected and encrypted using AES-256 encryption into a digital twin which is stored on the HLF blockchain. AES-256 is a faster, more secure, and symmetric encryption algorithm. Privacy is enforced by sharing the encryption key with the required actors of the system. The keys are stored on the blockchain using private data components that make sure that the key of the digital twin doesn't fall into wrong hands.

As the device lot is ordered and delivered, those details related to the device are also attached to the digital twin and hence referred to as the digital thread. The

Digital thread of the device is also encrypted using similar methods. The details which are added to the device's digital thread are as following:

- **Order Details:** This includes order number, buyer details, Purchase-Order number, etc.
- **Shipment Details:** Such as shipment tracking id, shipment history, shipping address, etc.

3.7 NuCypher

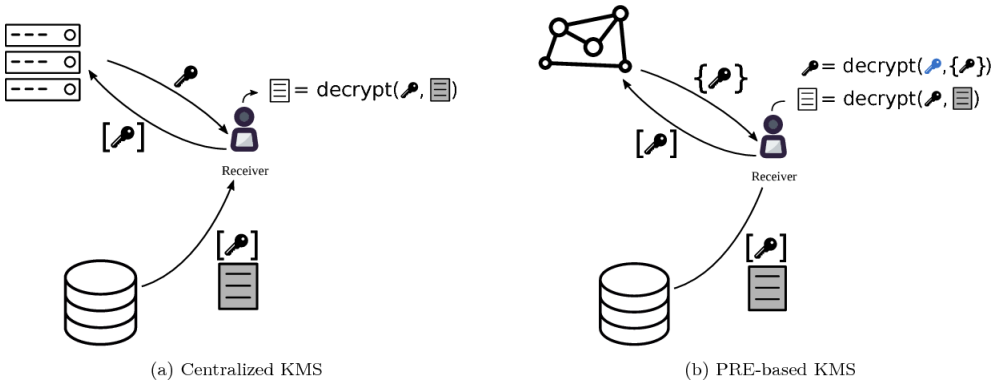
NuCypher[13] is a decentralized Key Management System(KMS) that provides features such as Proxy Re-Encryption and dynamic access control. It allows the data to be shared with a particular user for a certain period using Proxy Re-Encryption(PRE). By using proxy re-encryption, NuCypher ensures that unencrypted symmetric keys and re-encryption keys are not accessible once and keep data secure.

PRE is a special type of asymmetric cryptography where a third party re-encrypts a ciphertext from the sender so that receiver decrypts it, without knowing the message. PRE provides better security for the keys as compared to the traditional Public Key Encryption(PKE) as no private keys are needed to decrypt the message and they also can not be computed from the public keys.

3.7.1 Advantages of NuCypher

NuCypher as a decentralized KMS has the following advantages:

- **Versatile Data Sharing Policies:** NuCypher creates a policy-based data sharing using PRE that can be used on both public or private blockchain network. Data is encrypted using the receiver's key by a proxy node. NuCypher



(a) Centralized KMS (b) PRE-based KMS

Figure 3.3: Centralized KMS Vs PRE-based KMS[13]

also allows time-based data sharing that ensures the access to the data is revoked after a particular time period is over.

- **Unidirectional, and non-interactive:** NuCypher is a unidirectional and non-interactive system. unidirectionality makes sure that the receiver never needs to reveal its private key to decrypt the data and non-interactive means only the public key of the proxy is required. This feature ensures that the private keys of all entities are kept private.

- **Splitting trust across nodes:** By setting a policy, it is possible to split the trust among more than re-encryption nodes and specify a minimum number of nodes required to provide decryption keys. Greater availability and reliability is provided by splitting the trust among multiple nodes.

NuCypher allows the dynamic access management of the data through PRE. In our system, This feature of NuCypher is used to share the data securely outside the network. When a medical device is implanted or used for the patient, device details should be shared with the patient. As patients are not part of the blockchain network, it is not possible to share the data through HLF channels. NuCypher uses the PRE to encrypt the data using the proxy node and patient's public key to make it available

only to the patient.

NuCypher re-encrypts the data and generates the message kit along with sharing policies that are shared with patients. This method of encryption allows the device data to be shared for a specific period. Thus, dynamic access control is implemented in the system. For our system, we have used the testnet of NuCypher network that is deployed on the goerli ethereum testnet.

3.8 Kubernetes

Kubernetes is a lightweight, expandable, open source framework that enables both declarative configuration and automation for the management of containerized workloads and services. It has a broad ecosystem which is rapidly increasing. The facilities, support, and tools offered by Kubernetes are widely available.[4]

Kubernetes is a platform for reliably running distributed systems. It handles scaling and failover, as well as providing deployment patterns and several other functions.

Kubernetes offers the following features:

- **Service discovery and load balancing:** Kubernetes can expose a container using its DNS name or IP address. When traffic to a container is heavy, Kubernetes can load balance and distribute network traffic to ensure that the deployment is stable.

- **Automated rollbacks:** Given a particular state of the container, Kubernetes can bring the container from the current state to the desired state.

- **Self-healing:** Kubernetes provides a self-healing feature which make sure to auto-recover the stopped containers either by restarting or replacing them. Kubernetes does this task itself without the user's attention.

- **Storage orchestration:** Kubernetes allows an automatic storage mounting of any type including local storage, public cloud providers, and more.
- **Automatic bin packing:** Kubernetes runs containers of the specified RAM and CPUs on the nodes in the given cluster. To make use of the most of resources, Kubernetes will fit containers onto nodes.

3.8.1 Kubernetes Architectural Components and Specifications

Kubernetes architecture consists of the following important components:[17]

- **Node:** Kubernetes is built on a node, which can be a physical or virtual machine. Kubernetes launches containers within pods from nodes, which are worker machines.

  In our system, we have hosted three nodes, one for each HLF organization. additionally, five nodes are occupied by the ordering service and four additional nodes are up for four smart-contracts. Cluster manages the nodes and makes sure to arrange the new node if one gets down.

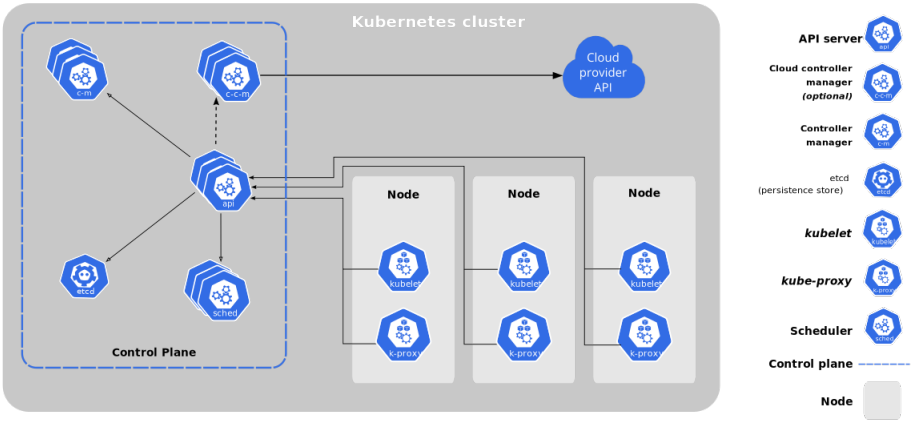


Figure 3.4: Kubernetes Components[3]

- **Cluster:** A cluster is a set of nodes grouped together. Clusters make sure that the network is self-healing. If one node fails, the application will be still accessible from the other node in the cluster. Also, in case of excessive load, the cluster balances the load among itself.

  We have used local minikube cluster to deploy the system. An Entire blockchain network can be hosted on one cluster. However, in the case of scaling the network up for multiple regions, more clusters can be created and joined. A Kubernetes cluster is made up of one or more Kubernetes-managed nodes. A container runtime, Kubelet (responsible for starting, stopping, and handling individual containers), and kube-proxy are all present on each node.

- **Master Node:** One or more master nodes running the Kubernetes control plane are contained in the Kubernetes cluster. There are various processes in the control plane, such as an API server, a scheduler, a controller manager, etc.

- **Pod:** The smallest deployable unit that Kubernetes can control is a pod. A pod is a logical group of one or more containers that share the same port space and IP address. A pod's main function is to support processes that are co-located, such as an application server and its local cache. Containers can find each other inside a pod through localhost and can also use standard inter-process communications to communicate with each other. If a node where the pod is running dies, the pod is deleted. It can then be replaced by a new pod with the same name but a different unique identifier (UID).

  In our system, all peers, certificates, CouchDB are deployed on the individual pods. Apart from this, all ordering nodes, as well as smart-contracts, are deployed on the individual pods.

All nodes, pods, and services in our system are deployed using helm charts. Helm

is an open-source package manager for Kubernetes. In helm, a chart is an organized collection of files that describes the related set of resources for the deployment. In our system, we have used the following structure to deploy the charts on the Kubernetes using helm:

- **Chart.yaml:** Basic information about chart such as name, description, dependencies.
- **values.yaml:** Environment parameters of the chart. For example, network details such as organizations of the network, number of peers, channels information, smart contracts information, etc.
- **templates:** A directory containing the templates of deployment which is combined with the values in values.yaml, produces the actual deployment manifest. In our system, templates include, create channels, join channel, install smartcontracts, etc.

3.9 System Design

In our system, the following are the three main organizations/actors:

- **Device Manufacturer:** Device manufacturers are the primary entities that design and develops the medical device and provides them to healthcare institutions. A device manufacturer will be putting the encrypted digital twin on the blockchain. There are three roles in this organization:
  - Sales Representative: A sales representative is the person whom healthcare institutions may contact if they run out of the stock. Sales representative has its stock of supplies and can sell them to the healthcare institution.

If the sales representative also runs out of stock, it needs to contact the manufacturer for the refill.

  - Order Reception Staff: This is the person whom the healthcare institutions directly speaks and submits order on behalf of the healthcare institution.
  - Device Manufacturer Admin: This role works as the administrative unit of the device manufacturer and hence manages the task of adding the digital thread on the blockchain, validating the order, etc.

- **Healthcare Institution:** Healthcare institutions is the organization where the device is used either used in surgeries or operations, or the device is stored for the future use. The digital thread will be shared with the healthcare institution once it purchases the device. Following are the two roles in the healthcare institution organization:
  - Department Supply Coordinator: These are the units that divide the provided medical devices to the corresponding medical department.
  - Material Manager: Material managers are the ones who manage the medical device till it gets used in the surgery or gets returned to the device manufacturer.

- **Operator:** The operator is the monitoring actor in the system. it executes various functionality that keeps the system running smoothly. It has also the job to run various shipment companies' APIs such as FedEx, UPS, USPS to fetch the shipment data of medical devices time-to-time.

HLF offers a feature of multiple channels to communicate with other participants privately. Each channel may hold a different asset and thus different smart-contracts

to manage the asset. Each channel can be considered as a different blockchain and hence HLF ensures that no two channels can share the data.

For our network, we have the following four channels:

- **Digital Thread Channel:** Digital thread channel is used to store and share the digital thread of the medical devices. This channel is shared with all three actors of the network. This channel also manages the shared key of the digital thread using the private data collections.

- **Order Channel:** The order channel stores the order details of all orders submitted by either healthcare institutions or sales-representative. This channel is shared with all three actors however, using private data collections it is made sure that, only required fields such as order no., shipment no. are shared with the operator.

- **Return Channel:** Return channel stores the product-return order and all details about it. Return order has a unique identifier as 'Return Materials Authorization' number which is used as a primary key. Return order also involves the shipment details and hence operator has the authority to write this channel.

- **Patient Channel:** Patients are the final user of the medical devices and hence it's their right to know about the product. Hence, this channel stores the digital thread encryption key encrypted using PRE and Healthcare Institution's key. Using both of these keys along with own key, a patient can view the details of the product. This channel is used to store only key information and thus shared with Healthcare institutions and the operator.

All smart-contracts on these channels are enforced by the endorsement policies defined by the network. These endorsement policies specifies the required MSPs, the

signatures of them should be present on the proposal, then only the transaction gets approved and changes to the ledger are committed by all the peers. In our system, the following are the endorsement policies for each of the channel:

- **Digital Thread Channel :**AND(DeviceManufacturerMSP.member, OR(HealthCareInstitutionMSP.member, OperatorMSP.member))
- **Order Channel:** AND(DeviceManufacturerMSP.member, HealthCareInstitutionMSP.member, OperatorMSP.member)
- **Return Channel:** AND(DeviceManufacturerMSP.member, HealthCareInstitutionMSP.member, OperatorMSP.member)
- **Patient Channel:** AND(HealthCareInstitutionMSP.member, OperatorMSP.member)

3.9.1 Submitting a Medical Device Order and its Shipment

The first use case that our system handles using blockchain solution is submitting an order for a medical device and its shipment to the healthcare institution. Medical devices that are purchased and shipped in this supply-chain are tracked end-to-end and their complete details are appended to the digital thread. There are no standards to record the medical device details and hence the digital thread is formed using unstructured details and some unique features of the product. In our systems, orders can be submitted by either a sales representative or a healthcare institution. Device manufacturers look into their device availability and approve or reject the order. Once, an order is created till it gets shipped their details are stored in the digital thread.

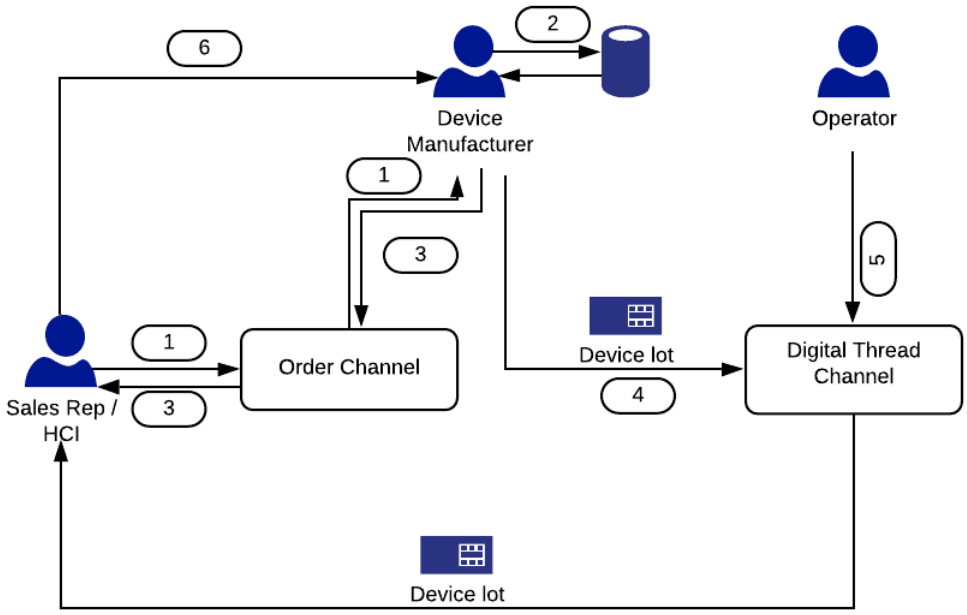


Figure 3.5: Order Submission and Product Delivery with Steps Numbered

1. **Order Creation:** Sales representatives or healthcare institutions submit the order of medical device(s).

2. **Device Lookup:** In this step, the availability of the ordered device is checked.

3. **Order Approval/Rejection:** If the device is available, then the order is approved and zero purchase order is generated which mentions the digital thread of the ordered product. In this step, the key of the encrypted digital thread is shared with the healthcare institution which purchased the product.

4. **Shipment begins:** This stage provides the operator with the correct tracking id and logistic details, also the digital thread is updated accordingly.

5. **Shipment Data:** The operator fetches the shipment data from the APIs and adds it to the device's digital thread.

6. **Order Complete:** Once a product is reached the destination, a sales representative or healthcare institution verifies the shipment. In case of errors or shipment failure, they can track the digital thread of the product shared with them.

As the transaction performed on the blockchain are verified and trusted, it gets easier to find the error or fraud in case of any. Also, maintaining the digital thread of the device helps us to find the exact point of failure.

3.9.2 Returning of a Medical Device

Medical device once expired needs to be returned back to the manufacturer for its recycle. In the existing system, there is no way to track the individual device by its expiry date, and hence becomes a difficult task for the sales representative. In our system, as the device is tracked even after its delivery to a healthcare institution (if it's used or not), it makes the task of searching the expired products easier for the sales representative. Sales representative collects the expired products and returns them back to the manufacturer using a via a logistic company.

1. **Lookup for expired devices:** A sales representative queries the blockchain for the expired devices in its nearby healthcare institutions.

2. **Results by the blockchain:** Blockchain provides the list of expired or ready-to-return devices in the nearby locations to the sales representative.

3. **Visit to the healthcare institution:** In this stage, a sales representative visits the healthcare institution that has expired products.

4. **Return initialization:** In this stage, a sales representative initiates the return procedure of the expired devices. Various details such as Return Material

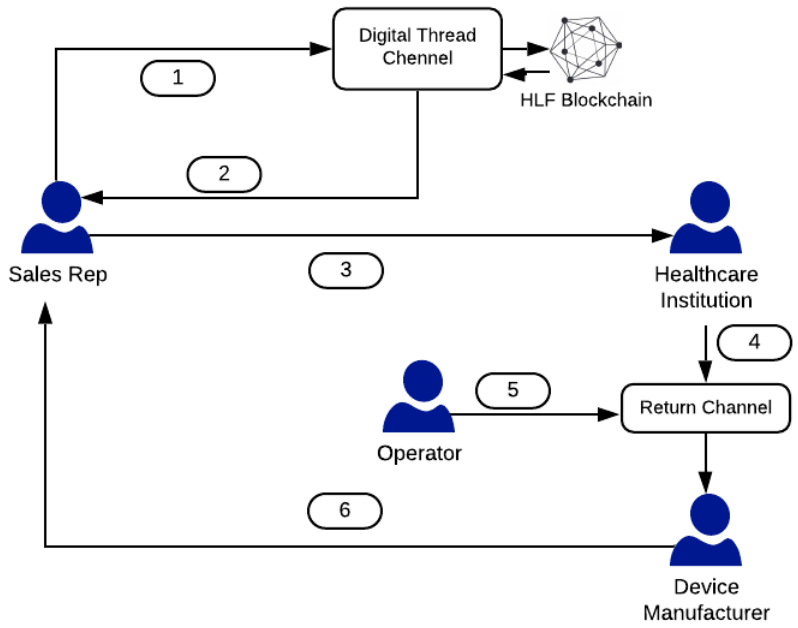


Figure 3.6: Return Order and Its Fulfilment with Steps Numbered

Authorization(RMA) number, return shipment tracking id are appended to the digital thread of the medical device.

5. **Shipment Details:** The operator adds the shipment details to the digital thread of the device time-to-time.

6. **Return Completed:** Once the device reaches the device manufacturer, the task is completed and the sales representative is notified.

3.9.3 Sharing of the Medical Device Digital Thread

Sharing of the medical device's digital thread access is a very important task as the digital thread has sensitive information about the medical device and should not be tampered with. Hence, the sharing of the digital thread is divided into two parts:

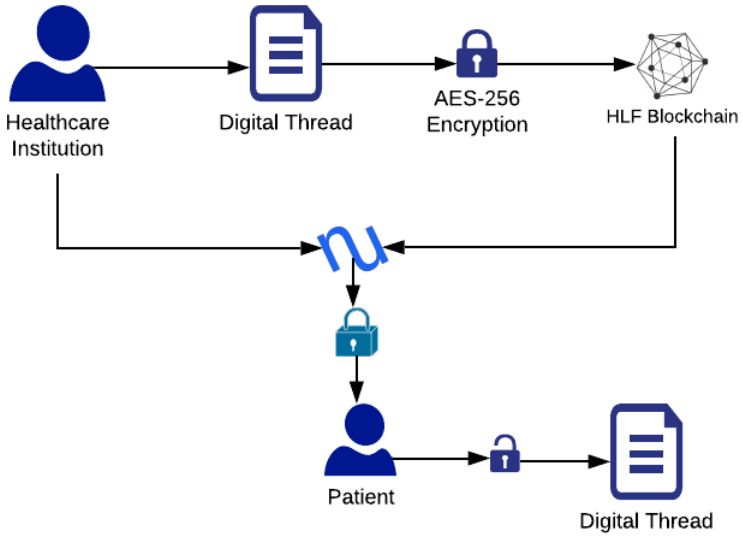


Figure 3.7: Sharing Digital Thread with Patients

1. **Sharing within the network:** Digital thread is shared with the device manufacturer users, healthcare institutions, or the operator according to their access levels.

2. **Sharing outside the network:** Digital thread is shared with the patients that are not part of the network.

Digital threads are encrypted using the AES-256 encryption algorithm by the device manufacturer and can be only decoded using the encryption key. As we saw in the above-mentioned use-cases, at different stages, the digital thread decryption key is shared with the required users of the blockchain network. In our system, sharing of the digital thread decryption key can be divided into the following types:

- **Sharing with Healthcare Institution:** When a healthcare institution purchases the medical device, its digital certificate is added to the digital thread

shared access and the decryption key is shared using the private data collections of the hyperledger fabric.

- **Sharing with Operator:** Operator is only able to access the shipment tracking id and the order id of the order. It also receives the decryption key using the private data collections.

- **Sharing with patients:** If a healthcare institution decides to share the data with patients, they can provide it using an additional security layer of NuCypher PRE. NuCypher re-encrypts the digital thread key and generates the policy using the patient's key. This policy is then used to encrypt the message. This message along with the healthcare institution is given to the patient which uses its key to decrypt the digital thread.

3.10 Deployment of the system in the current environment

Our system is the state of the art system that can be deployed to improve the traceability, visibility, and privacy of the medical devices supply chain that seems to be lacking in the current ERP-based systems.

ERP systems are based on centralized database systems which are vulnerable to privacy frauds on the other hand, our system is a decentralized system. Peers can join the network and have their own identity issued to them by the CA. Blockchain network also has endorsement policies that make sure that no fraudulent information is added to the blockchain.

Our system ensures the traceability of the medical devices using capture of digital thread and storing it on the blockchain. Privacy of the details is guaranteed by the strong encryption and sharing of the selective data. The major difference between the current ERP system and our system is tracking of the product even after its delivery

which makes the task of returning the expired product easier.

Our system is deployed on the Kubernetes which ensures the recovery of the nodes in case if it goes down, That makes our system more available and reliable. Kubernetes also improves the scalability of our system. If more nodes of other healthcare institutions, device manufacturers want to join the blockchain network, Kubernetes allows that without taking down the entire network.

As our system is decentralized and blockchain-based, a device manufacturer and healthcare institution needs to host their own nodes to be part of the system. The flexible design of our system allows multiple ordering services and distributed node system. This allows a more evenly distributed cost architecture among different organizations.

Chapter 4

EXPERIMENTS AND RESULTS

4.1 Experiments Description

We have implemented our system and tested its performance on various parameters to find the optimal configuration of the system. The purpose of the testing was to evaluate the system performance and compare it against the current working ERP systems. Currently, many device manufacturers and healthcare institutions use ERP supply chain management systems. The data provided by them is used as the baseline performance measure for our system.

4.1.1 Comparison Matrix

ERP systems manage multiple companies and a lot of transactions. Their key transaction standards for evaluating their performance are:

- **FDA Product Standards:** Every device sold in the United States must follow the FDA standards. The FDA has set some standards to evaluate, maintain and develop the supply chain of the medical devices that are followed by the ERP systems. The FDA standard mainly focuses on the traceability of the product and securing the supply chain using a standardized Unique Device Identifier. The standards were issued by the Office of Regulatory Affairs, Center for Drugs Evaluation and Research. However, the FDA offers only general guidance in advising device manufactures.[19]

- **CGMP Standards:** Current Good Manufacturing Practices(CGMP) are the quality systems for FDA regulated medical devices. CGMP sets the standards

that focus more on the shipment and delivery of medical devices. It provides the standards for shipment security, white-glove services, etc.

- **Alerts:** Alerts are sent to designated healthcare institutions or sales representatives based upon a shipment process. These alerts updates the required actors about the shipment.

To compare the current ERP system with our system, we are measuring the throughput given by our system while making an average number of transactions in ERP systems. We are also testing with more peers to check the scalability of the network. Also, we are tuning the block size of our blockchain to check, which configuration provides the maximum throughput.

We have also compared the NuCypher's PRE(Proxy Re-Encryption), and AES-256 encryption algorithm to find the best performance of all of them.

Other than that, we are testing the performance of handling extra load given by Kubernetes(used in our system) and comparing it against the existing docker deployments.

4.2 Experimental Data

Medical Device data is very sensitive and protected by FDA. Hence, the data used for this purpose is de-identified and it can't be associated with any entity. The data considered for this purpose covers device manufacture details, FDA details, and shipment details.

We have used the emulated data keeping the basic standards, received from different Medical Device Manufacturers. Some sample data was also generated for the testing purpose.

4.2.1 Description of Medical Devices Data Sharing

Although there are standards set by FDA and other regulatory bodies, those are quite generic standards and hence specification about entire data is not mentioned. That leads us to manage a very wide variety of data. Apart from this, as the supply chain is involved, various shipment data is also needed to be managed and shared.

We have used the digital thread for maintaining and sharing this data. Using the encryption method, any type of data can be stored in the blockchain. At the time of sharing, we can share the key of the encrypted digital thread with the required fields and the details will be shared.

For the testing of our system, we have emulated the shipment data to make the digital thread of the medical device. We have made sure that the type of the data is not consistent and tested the system with a wide variety of data.

4.2.2 Device Details Data Format

In our system, device details are dealt with in two formats:

- **Regular Format:** This includes mostly the JSON data which is converted from the device manufacturer's database using required methods. This data can not be stored and shared with other users as it becomes easily readable in this format.

- **Encrypted Format:** This data includes encrypted files of all data using the AES-256 algorithm. Also, sharing the data with the patient involves NuCypher's PRE encrypted files. Data is stored on the blockchain using this format.

4.3 Actors in Our System

Our system has three main actors. These actors cover different functionality of the medical devices supply chain. These actors also have roles in them to which users can be classified and given a functionality accordingly.

1. **Device Manufacturer:** The device manufacturer, encrypts the data and store it on the blockchain, seller of the devices
    - Sales Representative: Submits the order on behalf of healthcare institutions, initiates the return order
    - Order Reception Staff: approves the medical device order by checking the availability of the devices
    - Admin: Encrypts the data and stores the key on the blockchain
2. **Healthcare Institution:** Purchases the device by submitting the order, returns the device in case of unused and expired.
    - Department Supply Coordinator: Distributes the devices according to the department, updates the digital thread of the device.
    - Material Manager: Provides the device in case of its usage, hence updates the digital thread.
3. **Operator:** Adds the shipment details by tracking the id, can update the digital thread but can only read the tracking id of the digital thread.

The digital thread can also be shared with the patients using NuCypher PRE. However, only encryption details are stored in the blockchain. Hence patients are not part of the system network.

4.4 Hyperledger Fabric System

We have used the HLF permissioned blockchain to build the system. HLF has different components and few key components of the HLF architecture are:

1. **Certification Authority (CA):** Our system has three Certification Authorities(CA), one for each actor. It provides the following features:

   - issuance of Enrollment Certificates (ECerts)
   - registration of identities, or connects to LDAP as the user registry
   - certificate renewal and revocation

2. **Peers:** Peers are the very basic element of the HLF blockchain network. They host the smart contract, ledger and they do the task of committing the transaction, querying the ledger. For redundancy, resilience, and reliability, every organization hosts multiple peers. Each peer host the copy of the ledger and makes changes when the transaction is validated. We would vary the number of peers to check the scalability of the system.

3. **Client Peers:** Client Peers hosts the REST APIs, which allow the user front-end application to interact with the network and smart contracts deployed on the network. A client application talks to a client peer GRPC(Google's Remote Procedure Calls) protocol and submits transactions and queries to the peer.

4. **Ordering Service:** The ordering service for the network is RAFT. RAFT is a crash fault-tolerant consensus algorithm. It follows the "leader and follower" model where a follower is chosen by the election method. RAFT ordering is easier to set up than other 'Kafka' ordering service. In our system, we have one RAFT leader node with 4 follower nodes.

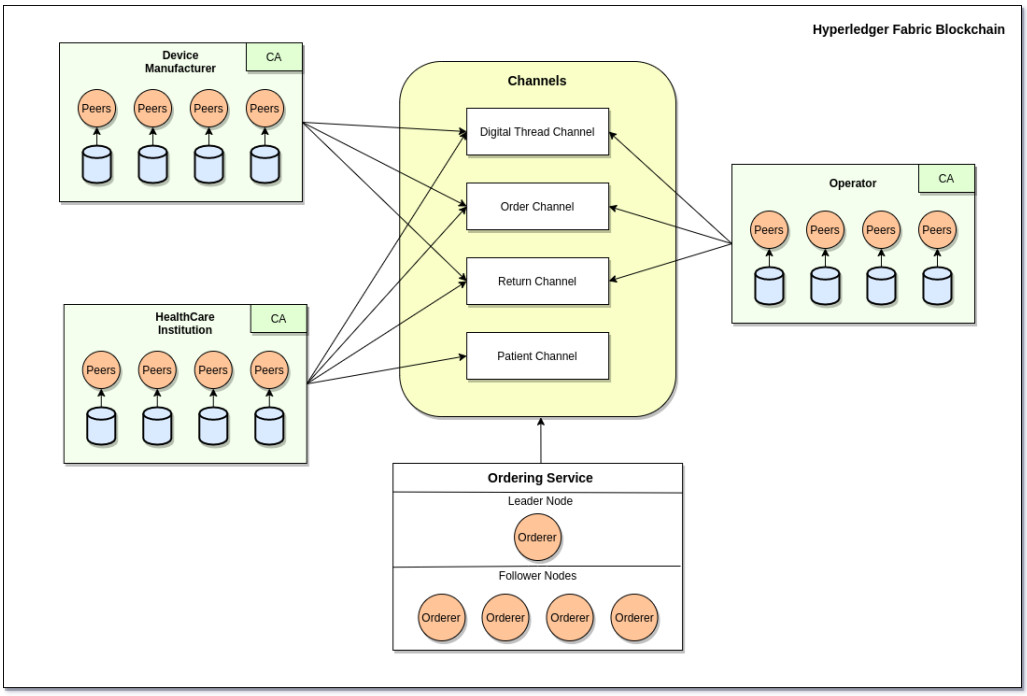


Figure 4.1: Blockchain Network with RAFT Ordering Service

5. **Channels:** We have used four channels to smoothly carry out the functionality of our system. Channel stores the assets info and smart-contracts manages the asset.

6. **CouchDB:** To provide simple and efficient storage and retrieval of ledger states, HLF offers the world state which is physically implemented as a database. The world state stores the latest values of the keys. Whenever a peer invokes a transaction, the read/write set is generated against the world state. To make the querying faster and efficient indexes can be added to the world state of the blockchain. Currently, HLF offers two options for the world state database that are LevelDB and CouchDB.

   CouchDB is an appropriate choice when the data is structured as JSON documents as CouchDB supports the rich queries and updates of richer data types.

In our system application, a CouchDB node is deployed for every running peer as every peer needs a separate world state. The indexes for the keys are based on device ID, order ID, and return-order ID.

4.4.1 Hyperledger Caliper

Hyperledger Caliper[2] is an open-source, blockchain benchmark tool. It allows users to test the performance of the different blockchain implementations to find the optimal configuration of the blockchain. The reports produced by the calipers provide the information about the latency and throughput of transactions in the network. Figure 4.2 shows the architecture for Hyperledger Caliper.

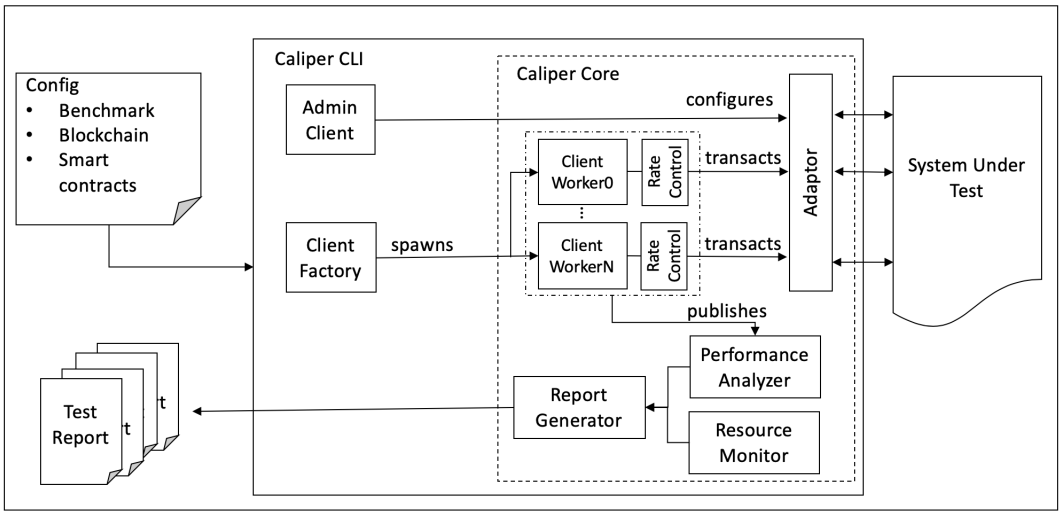


Figure 4.2: Hyperlerdger Caliper Architecture[1]

The configuration includes information about the benchmark to be run, the blockchain architecture to be tested, and the code for the smart contracts. These are fed to the interface, which creates two types of clients. The Admin client configures the system under test. This client is used to create channels, allow peers to join channels, and deploy chaincode. Multiple worker clients are also created which drives the test

load. These clients are driven using a rate controller, which impacts the rate at which transactions are submitted. The responses from these transactions are published to a performance analyzer. Resource monitor keeps track of the memory usage, disk I/Os, and CPU usage. All of these statistics are then compiled into a report.

4.5 Performance Testing

4.5.1 Description of System Under Test Configuration

We have tested our HLF system with a varying number of peers and each configuration with a varying number of block sizes. The ordering service used is RAFT. The following table contains all the system parameters that remain constant for all tests. The following parameters have been tuned with different numbers of peers and

Table 4.1: Global Parameters for Performance Testing

Parameter Name	**Value**
Max Block Size	100 MB
Orderer Type	RAFT
No. of Orderers	5
System Memory (Total)	236 GB
OS	Ubuntu 20.04 LTS

different block sizes to find out the optimal configuration.

- **Send Rate:** Send rate is a rate at which transactions are sent to the blockchain. Every configuration is tested with five different rounds with five different send rates which are: 25, 50, 100, 150, 200. send rate is counted in transactions per second. The send rates displayed in the result table are different as the table shows the achieved send rate.

- **Max Message Count:** Maximum number of transactions allowed in a single block.
- **Preferred Max Bytes:** Size of the blocks that are replicated across the network.

Table 4.2: Number of Peers During Testing

Device Manufacturers	Healthcare Institution	Operator	Total
5	5	2	12
10	10	4	24
20	20	8	48

Table 4.3: List of Batch Size Used for Testing

Message Batch Size Tested
50
100
200
250
500 (For Block Size 16 MB)

The performance metrics for the first set of experiments can be described as follows:

- **Max Latency:** Worst Case Transaction Response Time (measured in seconds).
- **Min Latency:** Best Case Transaction Response Time (measured in seconds).
- **Avg Latency:** Mean Transaction Response Time (measured in seconds).
- **Throughput:** Transactions Per Second processed by the system.

- **Successful Transactions:** Percentage of Transactions that were successful.
- **Failed Transaction:** Percentage of Transactions that were unsuccessful.

To test the implemented system, the following parameters were set under control to estimate the optimal performing hyperledger fabric network.

- **Orderer:** The ordering service used in the system is RAFT, a Crash-Fault Tolerant ordering service. Number of orderer nodes are five, to make sure that if one goes down, other would continue the system.
- **Number of Peer**: As per the table 4.2, the number of peers are set to 12, 24, and 48 respectively in three different configuration of the system. This parameter is dependent on the number of organization that are part of the network. As the invoked transaction is copied to all peers, the time taken to complete the transaction should increase as the number of peers are increased and hence the throughput is reduced.
- **Block size:** Three different block sizes are tested with above-mentioned peer configuration: 1MB, 8MB and 16MB. The block size is dependent on the size of the digital thread. As the block size is small, the time to complete the block with transactions is lowered and blocks are attached to the blockchain quickly increasing the throughput.
- **Transaction send rate and Batch size:** These parameters are configured in the caliper testing to test how the network performs with different traffic. Send rate is measured in the transactions per second and batch size refers to the number of transaction sent as a batch.

The following parameters were observed as a part of this experiment:

- **Throughput:** Throughput of the HLF network gives the measure of transactions per second completed by the network. This measure can help us to find the number of transaction a system could handle over a particular period. There are two types of transactions: Adding assets and Querying assets. Adding asset transaction would have a less throughput as it involves copying the transaction to all peers. On the other hand, querying asset transaction would comparatively give a higher throughput.

- **Latency:** Latency is the time required for the network to response the particular transaction. Caliper provides the latency in terms of following three measures: Max. latency, Min. latency and Avg. Latency. Latency also determines the speed and performance of the network.

Each experiment is followed by the results of the tested system configuration. Hyperledger caliper carries out this test by sending a fixed number of transactions at varying transaction rates. This experiment would allow us to estimate the recommended configuration to achieve the best performance from the Hyperledger fabric network.

Testing with Number of Peers: 12

Block size: 1 MB

Adding assets to Blockchain: Throughput given by the blockchain network while adding shown in Table 4.4.

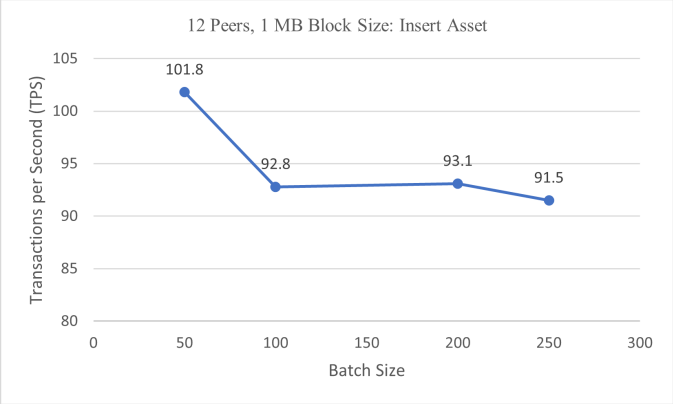


Figure 4.3: Hyperledger Fabric with 12 Peers and 1 MB Block Size, Adding Asset Performance

Querying Assets: Querying the ledger does not require endorsement policies to be followed and hence the time required to execute the transactions is comparatively lower than the adding assets transactions.

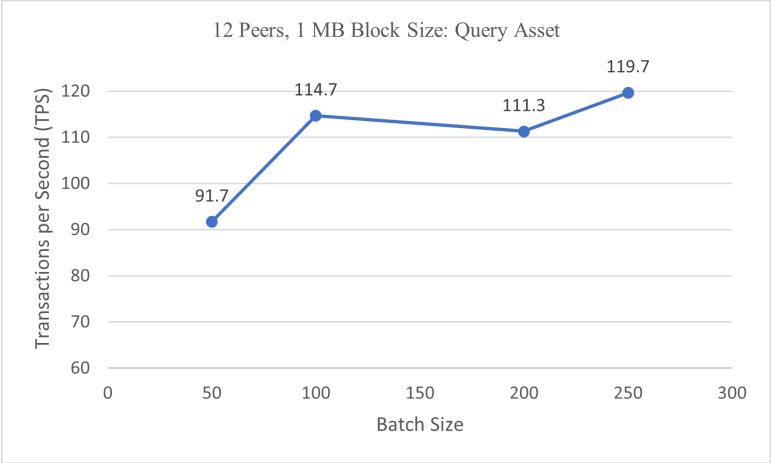


Figure 4.4: Hyperledger Fabric with 12 Peers and 1 MB Block Size, Querying Asset Performance

Results From the above tables and graphs, we can observe that:

Table 4.4: Performance Results for Adding Assets; 12 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	0.77	0.30	0.49	24.9
	100	0	50.2	0.80	0.33	0.56	49.1
	100	0	100.4	1.99	0.42	1.25	89.9
	100	0	150.6	3.44	0.49	2.38	101.7
	100	0	197.9	5.17	0.57	3.65	101.8
100	100	0	25.1	0.80	0.31	0.55	24.7
	100	0	50.2	0.95	0.39	0.70	48.7
	100	0	100.4	4.72	0.46	3.08	70.3
	100	0	149.9	5.75	0.63	4.03	84.0
	100	0	200.6	6.32	0.83	4.63	92.8
200	100	0	25.1	0.83	0.32	0.57	24.8
	100	0	50.2	0.94	0.42	0.69	48.6
	100	0	100.3	4.24	0.49	2.68	72.4
	100	0	150.0	5.40	0.57	3.99	86.7
	100	0	192.3	6.57	0.69	4.51	93.1
250	100	0	25.1	0.83	0.32	0.56	24.8
	100	0	50.2	0.91	0.37	0.69	48.5
	100	0	100.4	4.03	0.40	2.62	72.7
	100	0	150.5	5.51	0.49	4.00	85.8
	100	0	198.5	6.70	0.85	4.51	91.5

- Latency for all transactions for both adding assets or querying assets was quite low as the number of peers were low.
- Throughput given by the network was high for batch size 50 in case of adding assets on the other hand it was high for batch size 200 while querying assets. But in both cases, the change was very slight and overall good throughput was observed.
- No transactions failed during the testing as the ordering service is a crash-fault tolerant.

Table 4.5: Performance Results for Querying Assets; 12 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	0.75	0.23	0.39	24.8
	100	0	50.2	0.58	0.26	0.41	49.5
	100	0	100.4	5.85	0.91	3.78	74.5
	100	0	150.6	6.69	0.96	5.07	91.7
	100	0	197.9	8.59	2.69	7.14	84.7
100	100	0	25.1	0.72	0.23	0.39	24.8
	100	0	50.2	0.56	0.28	0.40	49.4
	100	0	100.4	0.82	0.26	0.49	96.1
	100	0	149.6	5.28	0.47	3.45	106.0
	100	0	188.7	5.74	1.04	4.46	114.7
200	100	0	25.1	0.73	0.24	0.39	24.8
	100	0	50.2	0.53	0.27	0.40	49.3
	100	0	100.4	0.68	0.27	0.48	96.7
	100	0	150.4	5.04	0.56	3.54	104.0
	100	0	199.6	5.87	0.96	4.65	111.3
250	100	0	25.1	0.70	0.24	0.39	24.8
	100	0	50.2	0.53	0.28	0.40	49.3
	100	0	100.4	2.58	0.39	1.50	94.7
	100	0	150.6	4.54	0.66	3.22	108.2
	100	0	200.7	5.34	1.00	4.16	119.7

Block size: 8 MB

Adding assets to Blockchain: Throughput given by the blockchain network while committing transactions shown in Table 4.6.

Querying Assets: Table 4.7 shows the results of the tests performed by caliper on 12 peer network and 8 MB block size for querying assets.

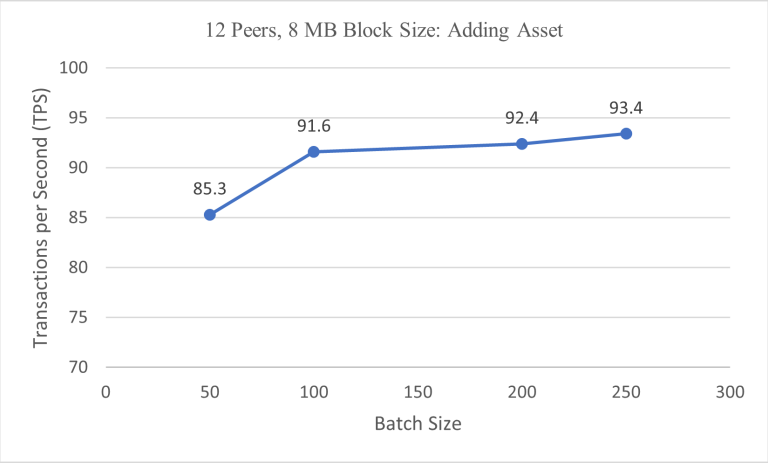


Figure 4.5: Hyperledger Fabric with 12 Peers and 8 MB Block Size, Adding Asset Performance

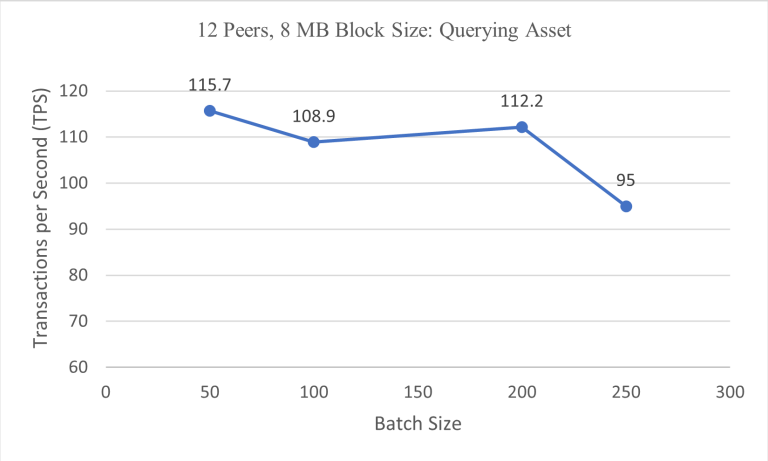


Figure 4.6: HLF Network with 12 Peers and 8 MB Block Size, Querying Asset Performance

Results

- Latency of the network is slightly increased in both cases than that was in block size 1 MB.

Table 4.6: Performance Results for Adding Assets; 12 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	0.78	0.33	0.56	24.8
	100	0	50.2	0.94	0.38	0.70	48.6
	100	0	100.4	3.73	0.49	2.42	75.2
	100	0	150.2	6.14	0.56	4.31	80.9
	100	0	198.2	7.22	0.73	3.65	85.3
100	100	0	25.1	0.85	0.32	0.59	24.8
	100	0	50.2	1.36	0.41	0.83	48.1
	100	0	100.4	5.55	0.48	3.62	67.4
	100	0	150.6	6.70	0.62	4.71	78.6
	100	0	200.8	6.68	0.71	4.72	91.6
200	100	0	25.1	0.83	0.34	0.58	24.8
	100	0	50.2	0.98	0.42	0.71	48.8
	100	0	100.3	3.94	0.46	2.70	73.9
	100	0	150.6	5.44	0.54	3.93	86.3
	100	0	200.7	6.43	0.84	4.54	92.4
250	100	0	25.1	0.87	0.32	0.58	24.8
	100	0	50.2	0.89	0.41	0.69	48.8
	100	0	100.4	3.99	0.54	2.70	73.1
	100	0	150.4	5.54	0.58	4.05	86.7
	100	0	197.7	6.45	0.79	4.50	93.4

- Throughput of the system is slightly reduced with the increased block size. However, the difference is very little. As shown in Figure 4.6, Throughput while querying is dropped when batch size is increased to 250.

- No transaction failure was observed even when the send rate is high in both adding assets and querying assets testing.

Table 4.7: Performance Results for Querying Assets; 12 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	0.69	0.24	0.39	24.8
	100	0	50.2	0.52	0.24	0.39	49.3
	100	0	100.4	0.65	0.26	0.44	94.8
	100	0	150.4	4.84	0.46	3.45	104.9
	100	0	200.6	5.25	1.13	4.11	115.7
100	100	0	25.1	0.73	0.22	0.40	24.8
	100	0	50.2	0.56	0.25	0.40	49.5
	100	0	100.5	0.81	0.28	0.52	96.2
	100	0	148.9	6.18	0.59	4.61	92.5
	100	0	199.4	5.81	0.84	4.72	108.9
200	100	0	25.1	0.69	0.24	0.39	24.8
	100	0	50.2	0.55	0.26	0.40	49.3
	100	0	100.4	0.71	0.25	0.48	96.7
	100	0	150.6	5.07	0.69	3.58	104.0
	100	0	200.8	5.70	1.32	4.55	112.2
250	100	0	25.1	0.76	0.25	0.40	24.8
	100	0	50.2	0.57	0.28	0.40	49.3
	100	0	99.8	1.28	0.41	0.77	95.0
	100	0	147.2	7.43	0.75	5.86	82.3
	100	0	169.2	9.29	1.98	7.88	77.9

Block size: 16 MB

Adding assets to Blockchain: Throughput given by the blockchain network while committing transactions shown in Table 4.8.

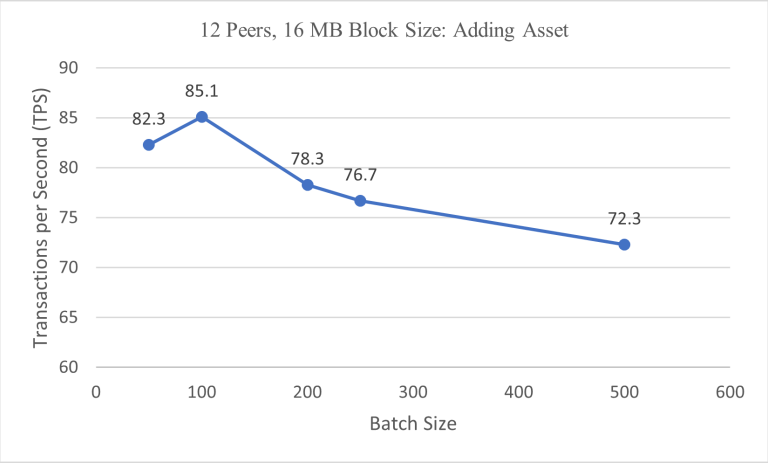


Figure 4.7: HLF Network with 12 Peers and 16 MB Block Size, Adding Asset Performance

Querying Assets: Table 4.9 shows the results of the tests performed by caliper on 12 peer network and 16 MB block size for querying assets.

Results

- Latency of the network while adding assets is comparatively lower than 1 and 8 MB block sizes. Whereas, while querying the latency is consistent with the previous two configurations.
- Throughput given by the system while adding assets was dropped but it was similar while querying assets. However, all transactions were passed at all send rates without failure which means, RAFT orderer is able to match the speed of incoming transactions.
- Although the transaction speed is lowered with a higher send rate, no transaction was failed during the testing.

Table 4.8: Performance Results for Adding Assets; 12 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	0.79	0.34	0.58	24.8
	100	0	50.2	0.95	0.39	0.70	48.7
	100	0	100.4	4.03	0.53	2.58	72.8
	100	0	150.6	5.85	0.54	4.18	81.8
	100	0	200.8	7.95	0.81	5.09	82.3
100	100	0	25.1	1.02	0.35	0.66	24.7
	100	0	50.2	1.08	0.42	0.79	48.5
	100	0	100.3	8.72	0.52	5.51	54.8
	100	0	150.6	7.78	0.77	5.92	72.0
	100	0	183.2	7.68	0.93	5.52	85.1
200	100	0	25.1	0.93	0.35	0.63	24.7
	100	0	50.2	1.40	0.38	0.79	48.7
	100	0	100.0	4.57	0.42	2.99	71.2
	100	0	149.7	7.57	0.89	5.45	75.1
	100	0	193.7	8.51	1.00	6.29	78.3
250	100	0	25.1	0.88	0.34	0.62	24.7
	100	0	50.2	1.12	0.38	0.74	49.0
	100	0	100.4	6.91	0.58	4.82	61.3
	100	0	150.2	7.80	0.98	5.88	73.3
	100	0	191.1	9.21	1.10	6.43	76.7
500	100	0	25.1	0.90	0.33	0.59	24.8
	100	0	50.2	1.03	0.37	0.73	48.4
	100	0	100.4	7.67	0.50	5.12	57.7
	100	0	149.2	9.42	0.79	6.61	67.3
	100	0	181.5	9.65	1.22	6.85	72.3

Testing with Number of Peers: 24

Block size: 1 MB

Adding assets to Blockchain: Throughput given by the blockchain network while committing transactions shown in Table 4.10

Table 4.9: Performance Results for Querying Assets; 12 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	0.72	0.24	0.40	24.8
	100	0	50.2	0.54	0.27	0.40	49.4
	100	0	100.4	2.56	0.32	1.76	87.0
	100	0	150.6	5.26	0.60	3.63	101.9
	100	0	200.5	5.59	0.97	4.59	107.6
100	100	0	25.1	0.72	0.24	0.39	24.8
	100	0	50.2	0.57	0.27	0.40	49.3
	100	0	100.4	0.73	0.26	0.49	96.2
	100	0	150.6	5.25	0.85	3.85	100.4
	100	0	200.3	6.76	1.16	5.38	103.3
200	100	0	25.1	0.74	0.25	0.40	24.8
	100	0	50.2	0.55	0.27	0.40	49.3
	100	0	100.4	0.76	0.30	0.52	95.6
	100	0	150.3	5.40	0.52	3.94	100.6
	100	0	197.8	5.86	1.28	4.67	112.2
250	100	0	25.1	0.68	0.24	0.40	24.8
	100	0	50.2	0.56	0.28	0.41	49.3
	100	0	100.4	1.59	0.31	0.75	95.7
	100	0	149.2	5.63	0.58	4.00	98.9
	100	0	200.6	6.40	1.44	5.31	103.4
500	100	0	25.1	0.75	0.24	0.39	25.0
	100	0	50.2	0.55	0.26	0.41	49.3
	100	0	100.4	1.49	0.25	0.59	93.2
	100	0	149.9	5.85	0.97	4.42	98.7
	100	0	189.1	6.91	1.83	5.58	101.9

Querying Assets: Table 4.11 shows the results of the tests performed by caliper on the 24 peer network and 1 MB block size for querying assets.

Results From the testing of the blockchain network, we can observe that:

- Latency of the network is significantly raised when the number of peers has doubled. However, the change was the most significant in the case of adding assets. That is because of the added number of endorsement policy signatures

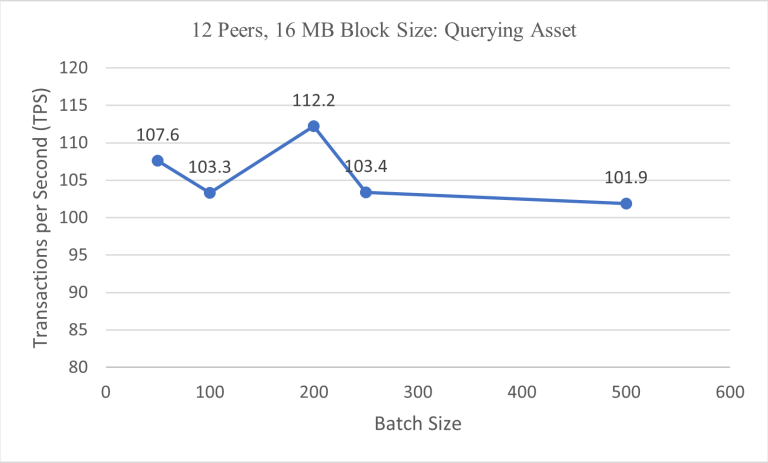


Figure 4.8: HLF Network with 12 Peers and 16 MB Block Size, Querying Asset Performance

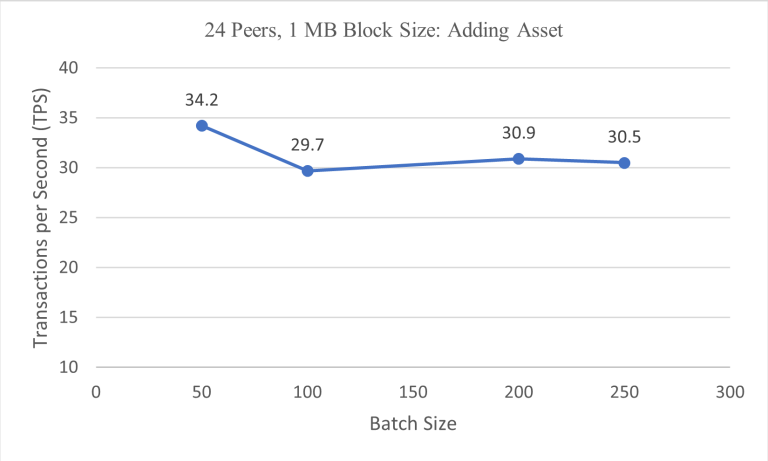


Figure 4.9: HLF Network with 24 Peers and 1 MB Block Size, Adding Asset Performance

and validations.

- Throughput given by the system is dropped in case of adding assets while it is slightly lowered in case of querying assets. This implies that adding assets is more time-consuming than the querying.

Table 4.10: Performance Results for Adding Assets; 24 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	33.73	0.86	20.03	13.7
	100	0	50.2	38.22	1.38	21.75	17.4
	100	0	100.5	27.87	2.14	18.05	27.7
	100	0	150.6	26.01	2.46	17.81	34.2
	100	0	200.9	29.37	4.74	17.93	32.1
100	100	0	25.1	32.66	0.76	18.97	13.9
	100	0	50.2	37.17	1.17	20.86	17.6
	100	0	100.4	30.39	1.70	18.97	26.8
	100	0	148.2	32.28	2.59	19.74	27.4
	100	0	200.8	31.00	3.60	19.05	29.7
200	100	0	25.1	34.10	0.76	18.83	13.6
	100	0	50.2	36.03	1.19	20.21	18.0
	100	0	100.4	28.71	1.78	17.75	26.9
	100	0	144.6	31.09	2.67	19.35	30.0
	100	0	200.8	29.36	4.19	19.16	30.9
250	100	0	25.1	35.44	0.86	20.62	13.5
	100	0	50.2	38.37	1.17	21.63	17.4
	100	0	100.4	33.46	2.02	20.52	24.5
	100	0	146.4	31.64	2.89	19.02	29.1
	100	0	200.5	30.99	4.27	19.22	30.5

- No transaction failure was observed while both Adding assets and Querying assets.

Table 4.11: Performance Results for Querying Assets; 24 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	1.05	0.33	0.49	24.9
	100	0	50.2	1.75	0.44	0.99	47.9
	100	0	99.9	9.81	1.05	6.11	75.2
	100	0	150.6	10.88	2.24	8.07	76.1
	100	0	196.3	11.25	5.87	8.45	83.5
100	100	0	25.1	1.22	0.34	0.51	24.9
	100	0	50.2	7.39	0.39	4.54	39.9
	100	0	100.4	10.29	0.88	6.57	69.2
	100	0	150.6	10.56	3.49	7.88	71.7
	100	0	200.8	12.51	2.80	8.71	74.9
200	100	0	25.1	1.42	0.34	0.51	24.9
	100	0	50.2	1.38	0.41	0.80	48.3
	100	0	100.0	9.74	0.94	6.37	70.3
	100	0	150.0	11.69	3.17	7.64	78.0
	100	0	200.6	11.20	3.41	8.59	79.2
250	100	0	25.1	1.45	0.33	0.51	24.9
	100	0	50.2	1.51	0.42	0.89	48.0
	100	0	100.4	9.10	0.92	6.07	71.3
	100	0	150.6	10.28	3.45	7.76	78.2
	100	0	200.8	11.54	3.57	8.63	78.6

Block size: 8 MB

Adding assets to Blockchain: Throughput given by the blockchain network while committing transactions shown in Table 4.12.

Querying Assets: Table 4.13 shows the results of the tests performed by caliper on the 24 peer network and 8 MB block size for querying assets.

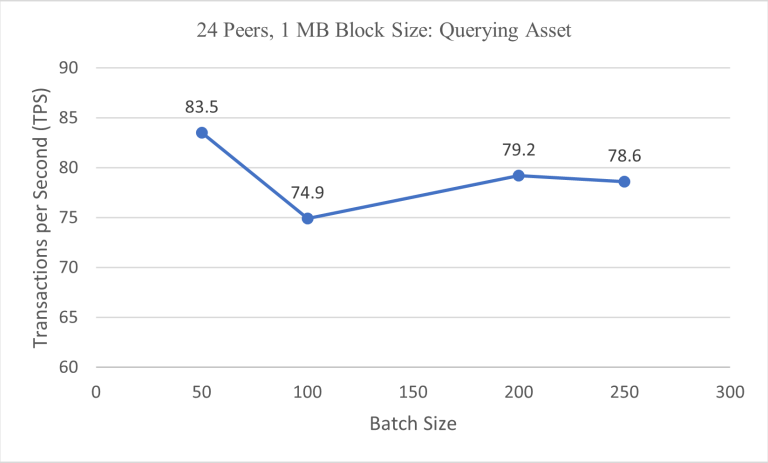


Figure 4.10: HLF Network with 24 Peers and 1 MB Block Size, Querying Asset Performance

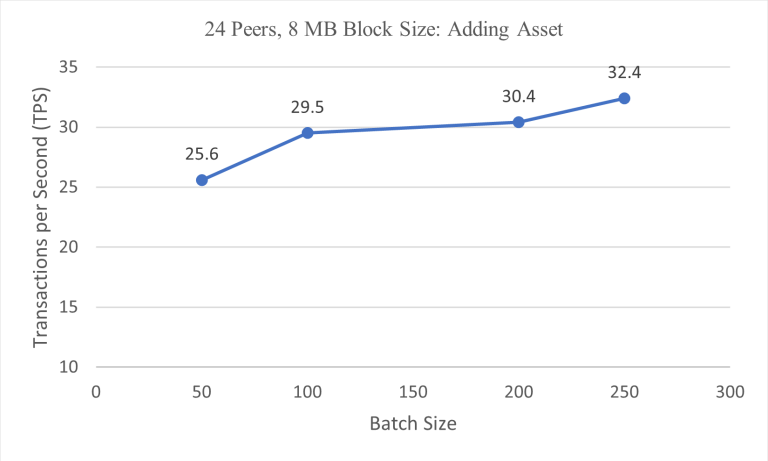


Figure 4.11: HLF Network with 24 Peers and 8 MB Block Size, Adding Asset Performance

Results From Table 4.12 and Table 4.13, we can observe that:

- Latency of the network is overall higher than the same 12 peers configuration. However, it is consistently lowered while increase in the batch size both while adding assets and querying assets.
- Throughput of the network is consistently getting higher when batch size is

Table 4.12: Performance Results for Adding Assets; 24 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	33.65	0.69	18.90	13.8
	100	0	50.2	36.63	1.38	20.78	17.9
	100	0	100.4	31.35	1.93	19.45	25.1
	100	0	150.5	36.19	3.03	20.84	25.6
	100	0	200.8	38.23	3.48	22.08	24.7
100	100	0	25.1	38.93	0.76	21.64	12.8
	100	0	50.2	39.33	1.31	22.15	17.0
	100	0	100.4	29.05	2.20	19.00	26.3
	100	0	148.8	29.74	2.60	18.38	29.5
	100	0	200.8	33.65	3.77	21.20	27.5
200	100	0	25.1	33.30	0.81	19.16	13.7
	100	0	50.2	38.25	1.15	21.94	17.2
	100	0	100.5	28.90	2.02	18.23	27.3
	100	0	150.3	29.76	2.72	18.96	30.4
	100	0	199.3	31.76	3.72	19.26	30.2
250	100	0	25.1	29.37	0.74	17.02	14.5
	100	0	50.2	38.75	1.07	21.23	17.3
	100	0	100.4	25.99	1.85	17.75	29.5
	100	0	150.6	29.43	2.55	19.62	29.6
	100	0	200.4	29.26	5.01	18.87	32.4

increased. Querying performance shows more throughput than adding assets as now endorsement policy validation is involved in querying.

- No transaction failure while both adding and querying assets signifies the capability of the RAFT orderer.

Table 4.13: Performance Results for Querying Assets; 24 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	1.23	0.34	0.50	24.9
	100	0	50.2	2.30	0.43	1.18	46.3
	100	0	100.5	10.95	1.04	6.87	61.7
	100	0	150.6	11.27	2.95	8.24	67.2
	100	0	200.7	12.35	3.90	9.06	67.3
100	100	0	25.1	1.04	0.34	0.49	24.9
	100	0	50.2	1.45	0.39	0.93	47.6
	100	0	100.4	10.64	1.08	7.12	66.2
	100	0	150.5	11.13	2.30	7.77	73.8
	100	0	200.4	12.49	4.89	8.32	76.4
200	100	0	25.1	1.10	0.34	0.50	24.9
	100	0	50.2	1.91	0.43	1.03	47.3
	100	0	100.4	9.64	0.94	6.43	72.6
	100	0	150.6	10.06	3.84	7.18	80.3
	100	0	199.9	10.19	3.11	8.17	81.4
250	100	0	25.1	1.44	0.34	0.51	24.9
	100	0	50.2	2.37	0.39	1.24	46.7
	100	0	100.4	11.57	1.28	7.37	66.5
	100	0	150.5	9.77	3.34	7.21	80.5
	100	0	200.6	10.52	3.56	8.28	80.2

Block size: 16 MB

Adding assets to Blockchain: Throughput given by the blockchain network while committing transactions shown in Table 4.14

Querying Assets: Table 4.15 shows the results of the tests performed by caliper on 24 peer network and 16 MB block size for querying assets.

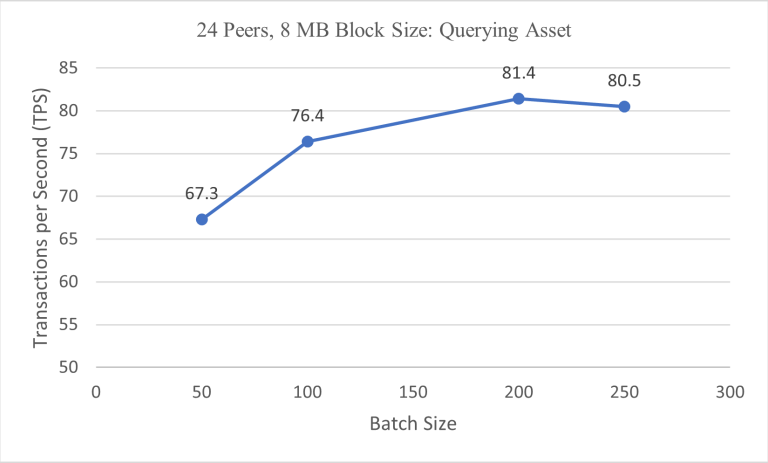


Figure 4.12: HLF Network with 24 Peers and 8 MB Block Size, Querying Asset Performance

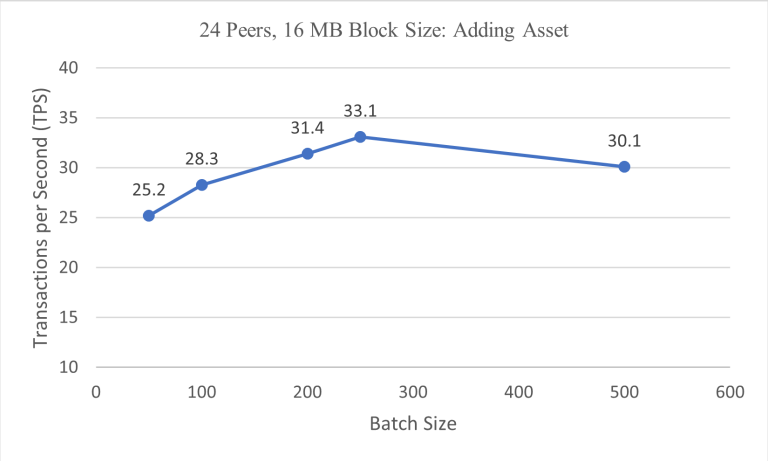


Figure 4.13: HLF Network with 24 Peers and 16 MB Block Size, Adding Asset Performance

Results From the performance matrices and graphs, it can be observed that:

- Throughput of the network is slightly lowered when compared to the previous two block sizes and the same number of peers. with an increasing number of peers, the throughput is improved except in the adding assets with 500 batch size, where the throughput is lowered.

Table 4.14: Performance Results for Adding Assets; 24 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	35.03	0.82	18.27	13.4
	100	0	50.2	38.84	1.24	22.14	17.2
	100	0	100.3	33.27	1.83	20.16	24.2
	100	0	149.5	35.28	2.85	20.95	25.2
	100	0	199.3	39.28	4.21	22.47	24.4
100	100	0	25.1	34.51	0.74	19.20	13.6
	100	0	50.1	35.88	1.16	20.05	18.0
	100	0	100.4	28.23	1.97	18.62	27.1
	100	0	150.6	31.45	2.43	19.68	28.3
	100	0	199.2	34.55	4.21	20.21	27.8
200	100	0	25.1	31.95	0.81	18.71	14.1
	100	0	50.2	37.90	1.17	21.02	17.5
	100	0	100.4	27.98	1.68	18.70	27.8
	100	0	150.6	29.50	2.58	18.49	31.1
	100	0	199.6	30.11	5.64	19.61	31.4
250	100	0	25.1	36.62	0.73	19.88	13.2
	100	0	50.2	39.34	1.21	22.74	17.0
	100	0	100.5	29.37	2.11	18.88	26.5
	100	0	150.5	27.38	2.15	17.41	33.1
	100	0	191.8	29.69	3.78	19.95	31.0
500	100	0	25.1	32.99	0.93	18.43	13.8
	100	0	50.2	37.08	1.16	21.24	17.7
	100	0	100.4	27.76	2.07	18.17	28.3
	100	0	150.6	28.51	2.69	19.24	30.1
	100	0	200.7	31.52	4.17	19.63	29.9

- Latency of the network is overall high with the worst that could be seen with batch size is set to 50. Latency is consistently lowered when increasing the batch size. This shows, higher number of messages per block shows good performance when the block size is high.

Testing with Number of Peers: 48

Block size: 1 MB

Table 4.15: Performance Results for Querying Assets; 24 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	1.04	0.34	0.50	24.9
	100	0	50.2	1.77	0.38	0.97	48.2
	100	0	100.4	9.97	1.00	6.94	63.0
	100	0	150.6	10.91	2.67	8.50	66.0
	100	0	200.7	13.22	4.18	9.42	66.2
100	100	0	25.1	1.22	0.33	0.50	24.9
	100	0	50.2	1.69	0.37	1.00	47.3
	100	0	100.4	10.10	0.91	6.42	68.2
	100	0	150.4	10.11	3.01	7.56	77.8
	100	0	197.4	10.56	5.71	8.55	74.1
200	100	0	25.1	1.38	0.34	0.50	24.9
	100	0	50.2	1.33	0.40	0.89	48.1
	100	0	100.5	9.45	1.16	5.98	71.9
	100	0	150.6	10.52	3.65	7.32	83.6
	100	0	199.0	11.52	5.78	8.86	77.5
250	100	0	25.1	1.17	0.34	0.50	24.9
	100	0	50.2	1.91	0.39	1.05	47.1
	100	0	100.4	9.82	1.03	6.21	72.3
	100	0	150.4	10.89	2.85	7.86	77.1
	100	0	196.8	10.67	5.96	7.82	83.3
250	100	0	25.1	1.05	0.34	0.49	24.8
	100	0	50.2	1.43	0.38	0.82	48.6
	100	0	100.3	9.55	0.82	6.22	72.1
	100	0	150.5	10.58	3.46	7.41	84.1
	100	0	200.7	10.27	2.90	7.80	86.5

Adding assets to Blockchain: Throughput given by the blockchain network while committing transactions shown in Table 4.16.

Querying Assets: Table 4.17 shows the results of the tests performed by caliper on 48 peer network and 1 MB block size for querying assets.

Results From the above tables and graphs, it can be observed that:

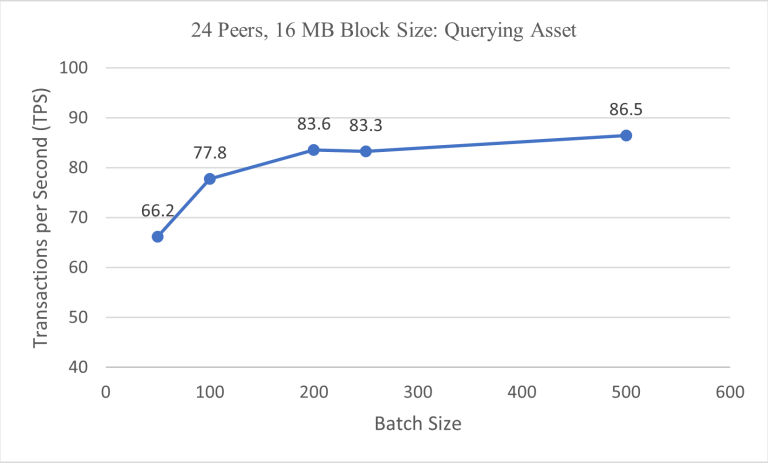


Figure 4.14: HLF Network with 24 Peers and 16 MB Block Size, Querying Asset Performance

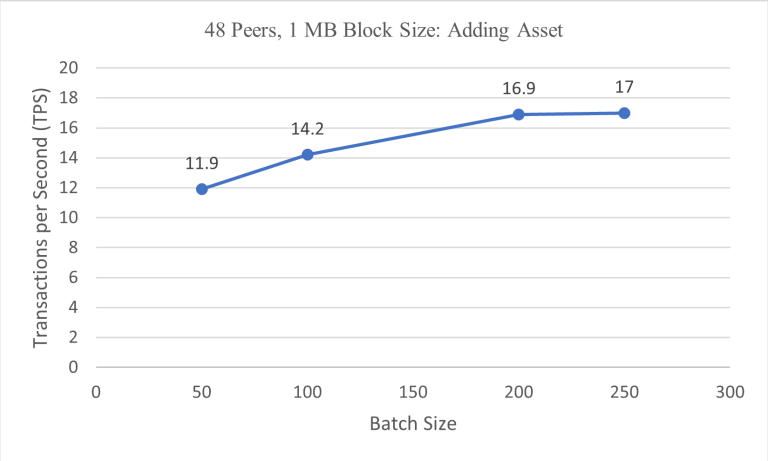


Figure 4.15: HLF Network with 48 Peers and 1 MB Block Size, Adding Asset Performance

- Latency of the network is noticeably raised when peers in the network are increased to 48. It is because of the transaction needs to get committed to all peers. However, the querying latency is comparatively low.
- Throughput of the blockchain network is dropped significantly for adding assets. However, it is better in the case of querying assets. It is visible that with

Table 4.16: Performance Results for Adding Assets; 48 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	73.65	2.71	44.90	10.2
	100	0	50.2	65.80	2.97	39.75	12.5
	100	0	100.4	81.17	5.05	47.03	11.6
	100	0	150.4	83.15	8.50	46.84	11.6
	100	0	200.8	83.02	13.82	49.57	11.9
100	100	0	21.6	91.51	3.07	57.58	8.8
	100	0	50.2	63.96	2.54	41.57	12.6
	100	0	100.4	64.11	4.46	40.54	14.1
	100	0	148.1	75.16	10.85	45.25	13.1
	100	0	200.7	68.25	11.00	41.12	14.2
200	100	0	25.1	79.01	3.45	50.81	8.7
	100	0	50.2	61.82	2.48	41.07	12.9
	100	0	100.4	53.60	4.22	37.27	16.9
	100	0	150.6	56.08	9.55	36.18	16.6
	100	0	200.5	60.86	8.26	37.90	16.3
500	100	0	22.1	91.55	2.37	61.76	8.3
	100	0	50.2	60.37	3.10	38.75	13.5
	100	0	100.4	61.78	3.99	40.36	15.1
	100	0	150.6	58.17	7.96	37.64	17.0
	100	0	200.8	60.24	11.69	35.59	16.3

increasing batch size, throughput is improved.

- No transactions were failed during the testing of both adding assets and querying assets.

Table 4.17: Performance Results for Querying Assets; 48 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	16.99	2.14	8.23	23.7
	100	0	50.2	14.42	0.92	8.84	36.3
	100	0	100.4	25.85	4.57	15.81	35.8
	100	0	150.6	23.37	8.37	16.81	40.2
	100	0	200.8	23.16	9.59	16.46	40.4
100	100	0	25.1	31.58	2.54	16.52	22.5
	100	0	50.2	13.95	0.88	8.70	37.3
	100	0	100.5	20.91	5.19	14.93	41.4
	100	0	150.6	23.92	6.11	18.21	39.3
	100	0	200.9	26.24	8.98	17.20	37.4
200	100	0	25.1	18.89	2.52	10.06	22.8
	100	0	50.1	17.74	0.97	11.01	34.4
	100	0	100.4	22.41	3.05	14.78	39.9
	100	0	150.6	23.80	7.66	17.20	40.7
	100	0	200.2	20.66	9.03	16.28	46.6
250	100	0	25.1	39.16	3.10	22.53	18.9
	100	0	50.2	13.56	1.98	8.60	37.4
	100	0	100.4	25.83	8.23	17.66	37.1
	100	0	150.6	22.11	4.19	16.89	43.5
	100	0	200.6	24.61	9.49	17.09	39.7

Block size: 8 MB

Adding assets to Blockchain: Throughput given by the blockchain network while committing transactions shown in Table 4.18.

Querying Assets: Table 4.19 shows the results of the tests performed by caliper on the 48 peer network and 8 MB block size for querying assets.

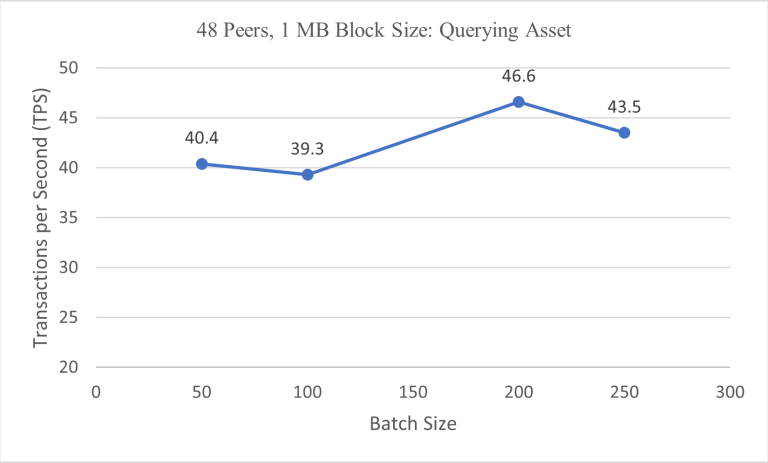


Figure 4.16: HLF Network with 48 Peers and 1 MB Block Size, Querying Asset Performance

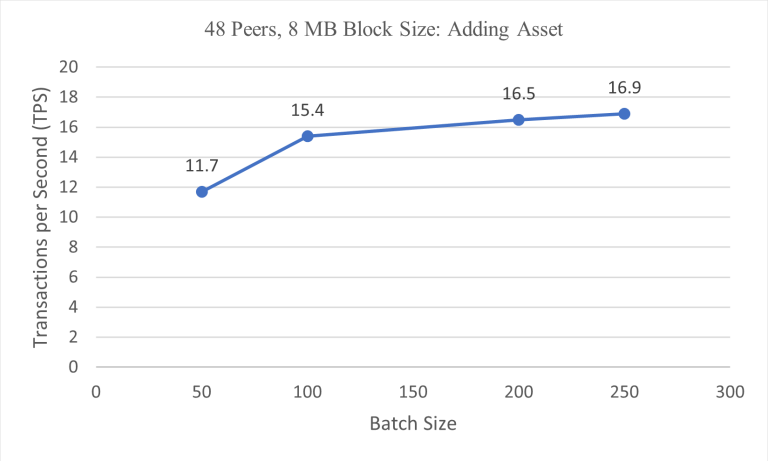


Figure 4.17: HLF Network with 48 Peers and 8 MB Block Size, Adding Asset Performance

Results From performance matrices and graphs, it can be observed that:

- Latency of the network is higher compared to the 1 MB block size and with increasing batch size, latency is lowered while both adding assets and querying assets.
- Throughput of the network with 48 peers is very high compared to the 24 peers

Table 4.18: Performance Results for Adding Assets; 48 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	97.16	2.16	63.75	7.5
	100	0	50.2	71.26	2.72	42.11	11.7
	100	0	100.4	80.62	5.33	47.64	11.4
	100	0	150.6	85.02	7.61	47.69	11.4
	100	0	200.5	89.99	12.47	50.17	11.0
100	100	0	25.1	82.74	2.14	50.32	8.2
	100	0	50.2	57.55	2.41	39.02	13.6
	100	0	100.1	60.85	6.93	38.31	15.4
	100	0	149.3	78.97	7.32	46.43	12.5
	100	0	200.7	69.56	9.16	42.15	14.1
200	100	0	25.1	75.15	2.93	52.66	9.1
	100	0	50.2	53.67	2.65	33.08	14.4
	100	0	100.3	65.12	4.20	41.87	14.0
	100	0	150.6	62.44	8.04	40.42	15.7
	100	0	200.9	59.62	8.05	37.71	16.5
250	100	0	22.3	84.99	2.26	58.32	9.8
	100	0	50.2	57.71	3.42	36.19	14.7
	100	0	100.4	57.23	4.61	36.90	15.7
	100	0	149.5	57.40	6.90	39.06	16.9
	100	0	200.6	57.63	12.28	41.53	16.6

network. Throughput of the network is improving with increasing batch rate except in the case of batch size 250.

- The failure rate while adding assets as well as querying assets is 0% that implies the network is taking time but still working without failure.

Table 4.19: Performance Results for Querying Assets; 48 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	14.04	2.72	8.40	21.5
	100	0	50.2	15.07	1.14	9.02	36.1
	100	0	100.3	23.61	4.24	14.72	39.5
	100	0	150.3	23.13	8.36	14.62	42.3
	100	0	200.4	27.02	8.03	21.04	35.5
100	100	0	25.1	21.43	2.66	12.91	22.3
	100	0	50.2	17.27	0.86	10.29	35.5
	100	0	100.4	21.38	6.99	14.52	40.5
	100	0	150.5	25.86	5.78	17.50	37.6
	100	0	200.6	21.06	9.40	15.20	46.8
200	100	0	25.1	18.39	2.63	10.04	22.8
	100	0	50.2	12.67	1.32	7.99	39.5
	100	0	100.3	22.72	4.54	16.02	40.1
	100	0	150.6	23.42	7.71	17.69	40.5
	100	0	200.7	21.20	9.36	15.76	44.8
250	100	0	25.1	30.24	2.87	15.87	22.1
	100	0	50.2	18.32	1.28	11.94	35.1
	100	0	100.5	22.93	4.84	16.87	39.6
	100	0	150.2	34.29	8.74	17.29	40.1
	100	0	200.8	22.49	9.39	16.40	43.4

Block size: 16 MB

Adding assets to Blockchain: Throughput given by the blockchain network while committing transactions shown in Table 4.20.

Querying Assets: Table 4.21 shows the results of the tests performed by caliper on the 48 peer network and 16 MB block size for querying assets.

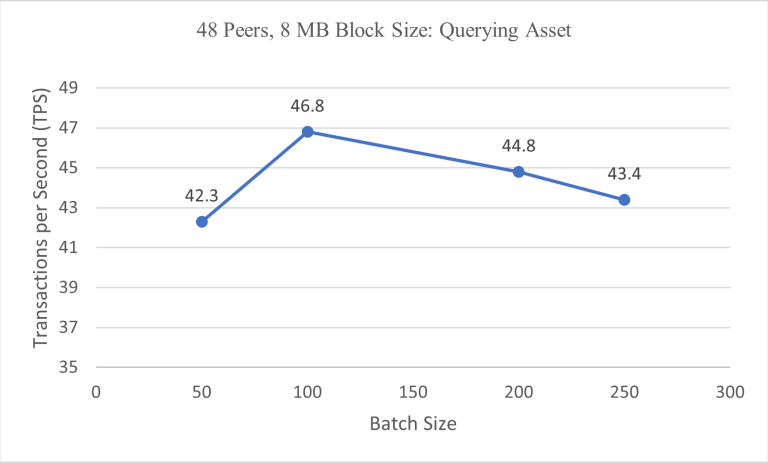


Figure 4.18: HLF Network with 48 Peers and 8 MB Block Size, Querying Asset Performance

Figure 4.19: HLF Network with 48 Peers and 16 MB Block Size, Adding Asset Performance

Results From the performance matrices and graphs, it is visible that:

- Latency of the network to add assets is highest for 200 batch size and it is comparatively higher than the previous two configurations of 48 peers network. Also, Querying assets performance of the network is slightly improved.
- Throughput of the network is lower than that with fewer peers network. How-

Table 4.20: Performance Results for Adding Assets; 48 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	21.5	79.95	3.70	51.75	9.3
	100	0	50.2	67.78	2.97	41.29	11.8
	100	0	100.4	88.17	4.78	48.69	10.8
	100	0	150.4	93.45	8.88	51.27	10.6
	100	0	199.5	87.88	8.72	48.21	11.2
100	100	0	22.8	89.10	2.69	58.96	8.6
	100	0	50.2	68.98	2.55	41.83	12.4
	100	0	100.4	67.82	5.36	40.81	13.9
	100	0	150.6	67.03	7.01	38.22	14.5
	100	0	200.8	71.27	8.46	41.15	13.8
200	100	0	25.1	72.07	2.18	50.15	11.2
	100	0	50.2	50.84	5.92	31.57	16.0
	100	0	100.4	66.48	4.38	40.34	14.3
	100	0	150.3	68.34	7.32	42.85	14.4
	100	0	200.3	57.85	8.37	34.68	16.7
250	100	0	25.1	67.13	2.58	50.00	10.3
	100	0	50.2	57.00	3.69	36.56	13.5
	100	0	99.2	61.73	4.71	39.97	14.9
	100	0	150.6	64.86	6.88	41.78	15.0
	100	0	200.9	62.11	8.87	41.27	15.7
500	100	0	25.1	65.88	2.08	51.11	11.1
	100	0	50.2	57.39	2.59	38.80	13.4
	100	0	99.8	67.04	4.65	41.56	14.0
	100	0	150.6	62.98	9.20	41.86	17.0
	100	0	200.6	65.33	8.98	40.70	14.9

ever, it is comparatively improved than the 1 MB and 8 MB block size.

- No transactions are failed even with a high send rate which implies that the RAFT ordering system is a crash-fault tolerant.

4.6 Kubernetes

Kubernetes is an open-source container orchestration system for automating computer application deployment, load-balancing, and scaling. A Kubernetes deployment

Table 4.21: Performance Results for Querying Assets; 48 Peers Network

Batch Size	**Success (%)**	**Fail (%)**	**Send Rate**	**Max Latency**	**Min Latency**	**Avg Latency**	**Throughput (TPS)**
50	100	0	25.1	14.92	2.03	8.82	21.7
	100	0	50.2	14.45	1.33	8.59	37.0
	100	0	99.0	21.27	2.77	14.30	40.0
	100	0	150.5	26.85	4.04	19.16	35.9
	100	0	200.7	23.21	7.26	15.26	42.6
100	100	0	25.1	21.19	2.02	11.33	22.1
	100	0	50.0	15.55	1.57	9.93	37.4
	100	0	100.5	20.83	5.65	14.56	41.6
	100	0	150.6	27.33	4.44	21.03	34.8
	100	0	199.8	21.52	9.80	15.44	44.6
200	100	0	25.1	17.57	2.43	8.96	22.4
	100	0	50.2	13.40	0.91	7.99	39.4
	100	0	99.4	24.18	3.69	16.75	37.6
	100	0	150.6	19.92	8.24	14.47	48.3
	100	0	156.9	24.14	9.88	18.85	40.0
250	100	0	25.1	38.72	7.26	22.98	19.4
	100	0	50.2	17.55	1.01	10.23	36.4
	100	0	100.4	28.52	4.90	19.48	33.7
	100	0	150.5	28.87	15.33	23.09	33.1
	100	0	200.0	24.82	9.43	19.03	38.6
500	100	0	25.1	13.73	2.98	7.78	21.8
	100	0	50.2	16.77	1.24	9.23	38.6
	100	0	100.4	20.10	2.57	14.40	46.1
	100	0	150.4	33.75	6.89	26.17	29.0
	100	0	200.2	21.83	10.53	17.22	43.2

can help load balancing, and scaling the network with extra peers in the system.

4.6.1 Performance Testing and System Under Test

The load balancing performance of the Kubernetes network is compared against the docker network with making the system flood with too many input requests continuously. **Locust.io** is a python based library that allows to conduct the load-balancing and stress testing of the containerized deployments such as Kubernetes and Docker.

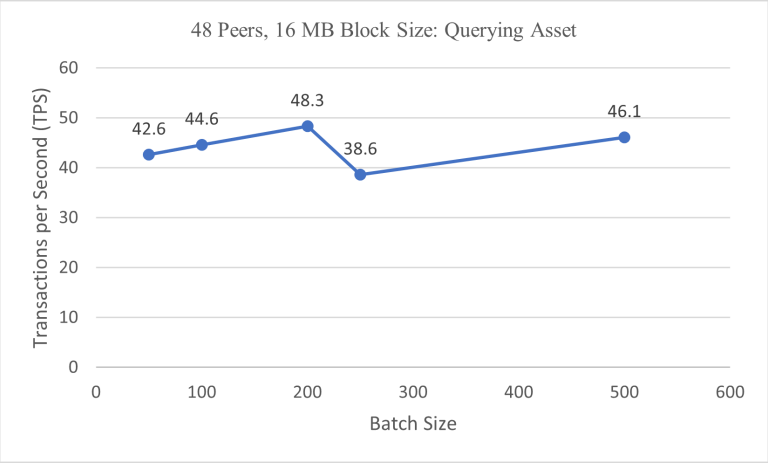


Figure 4.20: HLF Network with 48 Peers and 16 MB Block Size, Querying Asset Performance

System Under Test: Table 4.22 specifies the system parameters used for the testing of the Kubernetes network.

Table 4.22: System Description for Kubernetes Load-balancing Testing

Parameter Name	**Value**
Kubernetes	Minikube
Master Node	1
Worker Nodes	20
OS	Ubuntu 20.04 LTS
System Memory	16 GB RAM and 256 GB ROM

The following parameters were set under the control while conducting the test:

- **Number of Containers:** To conduct the experiment, I have set the number of containers to 20 and Hyperledger peer is deployed on the each peer. Kubernetes will make sure that the number of peers remain constant throughout the testing which docker containers can not.

- **Container specs:** To conduct the experiment, both Kubernetes pods and Docker containers are given a same memory and OS. It is also periodically checked if the containers are able to utilize their resource properly or not.

The following parameters were estimated as a result of these experiment:

- **Throughput:** Throughput is the count of Requests handled by the Deployment. As the throughput is captured in Requests per Second, this parameter can be used to estimate the load-balancing and scalability of the Deployment.
- **Latency:** Latency in the Deployment testing is the time required to response the request. This parameter is measured in millisecond and calculated in the following three formats: Min Latency, Max Latenct, Avg Latency.

Load-Balancing testing: Table contains the results obtained by performing load-balancing testing for 45 minutes on both docker and Kubernetes deployments.

Table 4.23: Kubernetes and Docker Performance for Load-balancing

Deployment	**Requests**	**Fails**	**Avg Latency**	**Min. Latency**	**Max. Latency**	**Throughput (RPS)**
Kubernetes	197271	0	10.21	2.86	96.10	100.4
Docker	18902	0	5.01	1.49	70.71	11.2

Results From the above experiments, it can be observed that:

- Kubernetes handled a significantly higher number of requests without failure than the docker network with the same configuration and throughout the testing period, Kubernetes was able to maintain the higher throughput.

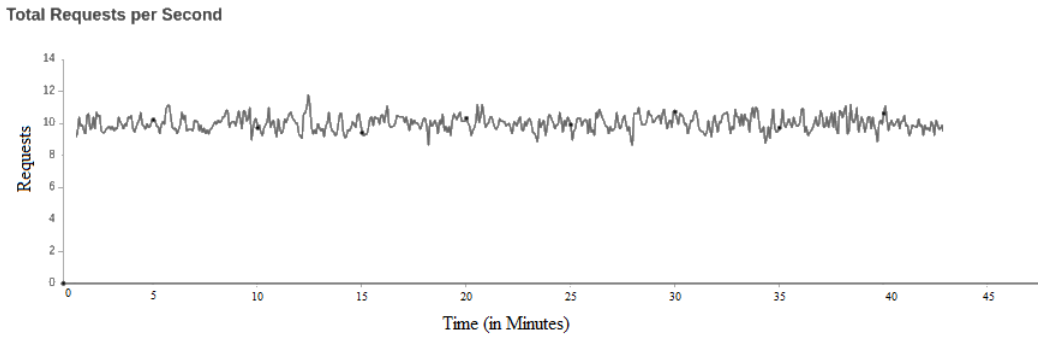


Figure 4.21: Requests per Second achieved by Docker for 45 Minutes

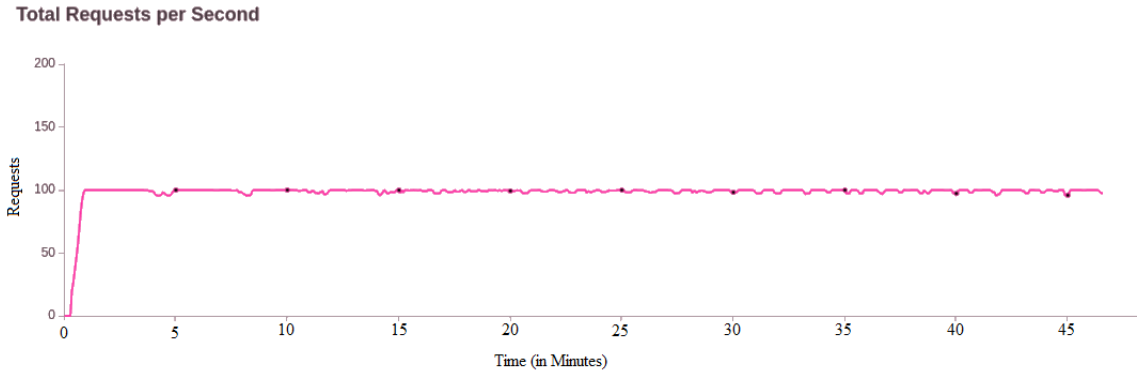


Figure 4.22: Requests per Second achieved by Kubernetes for 45 Minutes

- The latency of the docker system was slightly better however the difference was very small(approximately 5 ms).

4.7 Encryption Algorithm Testing

The security and confidentiality of the data are highly dependent on the encryption algorithm, but at the same time, it is important to provide the encrypted or decrypted data at a faster speed to improve the throughput. Our system uses AES-256 which is one of the strongest symmetric encryption algorithms to share the data among the participants and Nucypher PRE to share the data outside the network.

The following parameters were set under the control to conduct the time-it testing:

- **File Size:** To conduct the performance testing of encryption algorithm, Files of different sizes varying from 1 MB to 1000 MB are sent for encryption and decryption. As the size of the files is increased, the time taken for the operation should linearly increase as both encryption algorithms are Block Encryption Algorithms.

- **NuCypher Configuration:** NuCypher PRE is a decentralized encryption algorithm. It's nodes are deployed on the Goerli Ethereum Testnet. For this testing, I have deployed 50 nodes on the testnet. More nodes means the re-encrpyted ciphertext is distributed on more nodes and would increase the decryption time as the text needs to be retrieved from the more nodes.

The following parameters were estimated as a result of this experiment:

- **Encryption Time:** Encryption time is the time taken by the algorithm to encrypt the given file. Encryption time of the both algorithm is computed and compared against each other to find the better one to generate the homogeneous solution.

- **Decryption Time:** Decryption time is the time taken by the algorithm to decrypt the encrypted file. In most cases, it is observed that the decryption time is larger than encryption time.

4.7.1 Performance Testing

Timeit is a python based benchmarking library used to test the execution speed of both the AES-256 and NuCypher PRE encryption algorithm.

File Size	NuCypher PRE		AES-256	
	Encryption Time (sec)	Decryption Time (sec)	Encryption Time (sec)	Decryption Time (sec)
1	0.12	0.02	0.02	0.01
100	0.85	0.88	0.09	0.10
200	1.57	1.76	1.62	1.53
300	2.10	2.67	2.38	2.40
400	2.84	3.56	3.27	3.46
500	3.55	4.50	4.10	4.13
600	4.30	5.38	5.06	5.19
700	4.96	6.25	5.70	5.87
800	6.00	7.12	6.58	6.72
900	6.31	8.02	7.51	7.77
1000	7.14	8.91	8.21	8.86

Table 4.24: NuCypher and AES-256 Performance

Results Both NuCypher and AES-256 took approximately the same time and showed a good performance for both encryption and decryption.

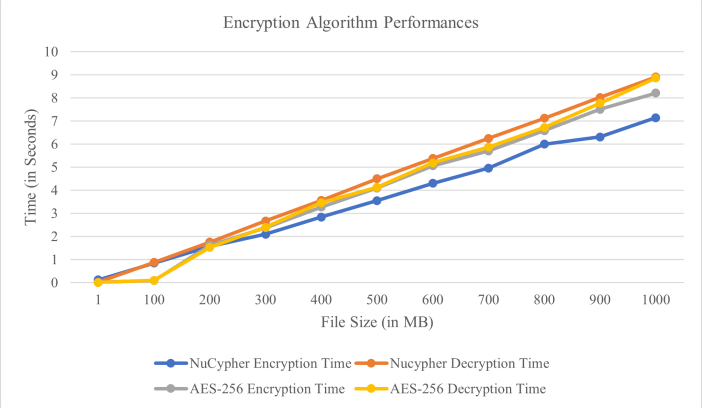


Figure 4.23: NuCypher and AES-256 Performance

Chapter 5

CONCLUSION

5.1 Performance Summary

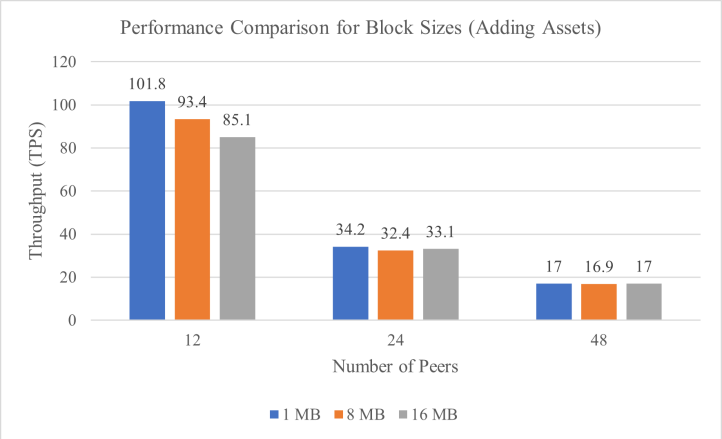


Figure 5.1: Performance Comparison According to Block Sizes(Adding Assets)

Summarizing all test experiment results for adding assets on the HLF network, we can conclude that:

- The throughput of the HLF blockchain network keeps dropping as the number of peers increases. Network with a lower number of peers offered the throughput of 101.8 TPS whereas that with the network with a higher number of peers achieved 17 TPS. This significant change is observed as the higher number of peers take more time to validate endorsement policies and getting endorsement signatures.

- Block size of 1 MB provided higher throughput with a lower number of peers however, as the number of peers increased, 16 MB block size configuration pro-

vided the equal throughput. Block size of the network can be cor-related with the size of the information in the record. This is used to determine the optimal digital thread size to store in the blockchain. smaller sizes are performing best in all peer configurations. However, 16 MB block size is showing a good performance in the scaled network.

- While performing adding assets with all configurations, no transaction failure was observed. This is because the ordering service used is the RAFT ordering service. RAFT ordering service had five orderer nodes in case of failure. Thus it can be concluded that the system is the crash-fault tolerant system.

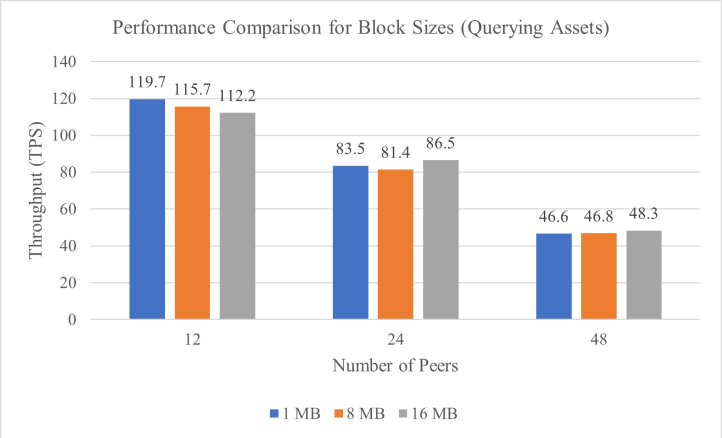


Figure 5.2: Performance Comparison According to Block Sizes(Querying Assets)

Summarizing all test experiment results for querying assets on HLF network, we can conclude that:

- The throughput of the network for querying assets keeps dropping for an increasing number of peers. However, the drop rate is very low as compared to the adding assets throughput drop rate. As querying does not require the endorsement policies, a sufficient amount of time could be saved and hence higher throughput is achieved.

- Throughput is higher for 1 MB block size in case of a lower number of peers but for higher peer configuration, higher throughput is achieved with a block size of 16 MB. It can be concluded that for a scaled network, a large block size configuration performs better than the small block size configuration. Hence the digital thread size can be large in the scaled network. However, the optimum performance for small networks can be observed when the size of the digital thread is small.

From the above performance results, we may conclude that, although HLF blockchain performs without failure in a scaled network, the throughput is affected significantly and thus it is limited to permissioned, business-centric solutions only. To achieve the best performance of the network, the number of peers should be low and size of the digital thread should be small as well.

Apart from Hyperledger Fabric blockchain performance testing, we performed the tests to test the scalability of Kubernetes and encryption algorithms.

- Kubernetes capability of handling requests is high compared to the dockerized network. Throughput of the Kubernetes network is 100 RPS whereas, that of the dockerized network is 10 RPS. The latency of the Kubernetes throughput is 5 ms higher than the dockerized deployment that is extremely slight compared to the Throughput of the Kubernetes deployment.

- AES-256 and NuCypher PRE both performed encryption and decryption on files in linear order. NuCypher PRE showed faster speed on all files than AES-256 and shown the potential to replace the AES-256 to form the homogeneous solution.

5.2 Existing ERP Systems vs our Blockchain-Based System

From the discussion in Section 5.1, we can conclude that our consortium blockchain-based system to improve the supply chain of medical devices has shown great potential. Following are the key improvements presented by our system compared against the existing ERP systems.

- **Traceability:** As we have discussed earlier, it is important in the supply chain to keep complete track of products, and in the case of medical devices, as the data is sensitive, traceability becomes more important. Our system with the help of digital threads provides end-to-end traceability.

  In the existing ERP-based solution, tracking the medical product was a difficult task due to the lack of the Unique Identifier and thus a device manufacturer or other participants may lose the track of the products and this leads to execution errors. In our system, a digital thread is used to store every detail of the product over its life-span and it's stored on the blockchain which provides trust and accountability in the system. Thus, the system becomes trustworthy, transparent, and visible.

- **Privacy:** Our system follows the privacy standards set by the FDA regarding medical device supply chain management and fulfills the minimum requirements.

  In our system, all detail of the medical device including its shipment details are encrypted using a strong encryption algorithm and then stored on the blockchain. Hence, data is not readily available on the network. When the data needs to be shared, only the required part of the information is shared by providing a key via private data collections of the HLF. The information

can also be shared with patients when required using NuCypher PRE. Patients are not currently part of the blockchain network but the platform could be generated later.

- **Scalability:** Supply chain accounts for 40% of the overall costs of the medical devices market.[12] Scalability refers to the level of expansion a system can go without compromising the throughput. Scalability is the major issue in the ERP systems that leads to various execution errors in the system. In our system, a number of peers define the scalability of the network. We have tested the network with 12, 24, and 48 peers and measured the throughput for both adding and querying the asset. Summarized results are shown in the table 5.1 From the data in table 5.1, it can be concluded that with a higher number of

	12 Peers	**24 Peers**	**48 peers**
Adding asset to the blockchain	101.8	34.2	17.0
Querying asset from the blockchain	119.7	86.5	48.3

Table 5.1: Throughput(in Tps) Compared for Different Number of Peers

peers, the throughput of the system is lowered. However, memory restrictions should be kept in mind as all peers were deployed on the local Kubernetes system which affects the throughput of the system. When deployed on the cloud Kubernetes, more optimized performance can be observed.

With the above discussions and comparisons, we can conclude that our system is a practical alternative to the present system. Some advantages of our system could be:

1. Our system can improve the traceability and accountability of medical products. Products are tracked even after their delivery.

2. HLF based decentralized system and Kubernetes allows the maximum availability of the system.

3. Privacy and confidentiality of the data is maintained.

4. The recycling ratio of the products by returning them to the manufacturer in case of expiry will be improved.

5.3 Future Work

- **Patient Involvement:** Currently, patients are not positioned as the actor in our system. Data is shared using NuCypher whenever required. Confidentiality can be improved by enrolling patients on the blockchain network and generating required policies when data needs to be shared with patients.

- **Involving More Healthcare Institutions:** In current scenario, the system is tested using the emulated data as the sensitive data of medical devices is not available easily. In future, this system can be tested with more realistic scenario where more healthcare institutions are on-boarded.

- **Justifying the Encryption Performance Results:** It was observed in the experiments that the NuCypher PRE algorithm performed better. It would a topic to the future work to understand that How NuCypher PRE being a decentralized performed faster than a symmetric AES-256 encryption algorithm.

- **Use of IPFS File Management:** As per the observations, smaller digital thread size shows a good performance in the smaller networks. However, issue may rise such as data loss while reducing the size of the digital thread. To avoid this issue, we can use the InterPlanetary File System(IPFS), a peer-to-peer file storing system. This will help to store the entire digital thread in a file and

the hash of this file will be stored on the HLF for reference. In this way, better performance for small networks can be achieved.

- **Reduced System Update Time:** It takes more time to install the system updates in case of distributed network as all peers need to make changes. This process can lead to pausing the system for a while till it updates. Hence, if we could make sure to keep the system available by providing additional peers while previous nodes are being updated, then the maintenance period can be avoided and better availability can be ensured.

REFERENCES

[1] "Architecture", URL https://hyperledger.github.io/caliper/v0.2/architecture/.

[2] "Hyperledger caliper", URL https://hyperledger.github.io/caliper/.

[3] "Kubernetes components", URL https://kubernetes.io/docs/concepts/overview/components/.

[4] "What is kubernetes?", URL https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/.

[5] Akkermans, H. A., P. Bogerd, E. Yücesan and L. N. Van Wassenhove, "The impact of erp on supply chain management: Exploratory findings from a european delphi study", European Journal of operational research **146**, 2, 284–301.

[6] Androulaki, E., A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains", in "Proceedings of the thirteenth EuroSys conference", pp. 1–15 (2018).

[7] Azevedo, P. S., M. Romão and E. Rebelo, "Advantages, limitations and solutions in the use of erp systems (enterprise resource planning)–a case study in the hospitality industry", Procedia Technology **5**, 264–272.

[8] C, R., URL https://www.supplychainmarket.com/doc/meeting-the-challenges-of-product-traceabilit-0002.

[9] Ciemcioch, S., "Overcoming challenges in healthcare supply chain management", URL https://www.warehouseanywhere.com/resources/healthcare-supply-chain-management/.

[10] Clauson, K. A., E. A. Breeden, C. Davidson and T. K. Mackey, "Leveraging blockchain technology to enhance supply chain management in healthcare: an exploration of challenges and opportunities in the health supply chain", Blockchain in healthcare today **1**, 3, 1–12.

[11] Commercient, T., URL https://www.commercient.com/3-big-erp-security-concerns/.

[12] Ebel, T., K. George, E. Larsen, K. Shah and D. Ungerman, "Building new strengths in the healthcare supply chain", McKinsey & Company .

[13] Egorov, M. and M. Wilkison, "Nucypher kms: Decentralized key management

system", ArXiv **abs/1707.06140**.

[14] Gaur, V. and A. Gaiha, "Building a transparent supply chain", URL https://hbr.org/2020/05/building-a-transparent-supply-chain.

[15] Gould, L. S., "What are digital twins and digital threads?", URL https://www.autobeatonline.com/articles/what-are-digital-twins-and-digital-threads.

[16] Grieves, M. and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems", in "Transdisciplinary perspectives on complex systems", pp. 85–113 (Springer, 2017).

[17] Gupta, G., "Components of kubernetes architecture", URL https://medium.com/@kumargaurav1247/components-of-kubernetes-architecture-6feea4d5c712.

[18] Jacob, J., "Introduction to digital twin", URL https://blog.amt.in/index.php/2020/07/21/introduction-to-digital-twin/.

[19] Johnson, J. A., "Fda regulation of medical devices", .

[20] Marine Magnet, T., "Top 10 digital twin execution system enables self-organised production withnbsp;real time learning control", URL http://www.marinemagnet.com/status-updates/top-10-digital-twin-execution-system-enables-self-organised-production-with-real-time-learning-control.

[21] News, A., "Erp flexibility is essential, but not as simple as it seems", URL https://abas-erp.com/en/news/erp-flexibility-and-upgradability-%E2%80%93-embracing-change-sustainable-technology-abas.

[22] Nijssen, S. and P. Bollen, "The lifecycle of a user transaction in a hyperledger fabric blockchain network part 2: order and validate", in "OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"", pp. 150–158 (Springer, 2018).

[23] Poonian, N., "Overcoming obstacles to supply-chain collaboration", URL https://www.supplychainbrain.com/blogs/1-think-tank/post/30400-overcoming-obstacles-to-supply-chain-collaboration.

[24] Putz, B., M. Dietz, P. Empl and G. Pernul, "Ethertwin: Blockchain-based secure digital twin information management", Information Processing & Management **58**, 1, 102425.

[25] Saric, A., "Great medical devices require high-functioning supply chains", URL https://www.medtechintelligence.com/column/great-medical-devices-require-high-functioning-supply-chains/.

[26] Team, D., "Types of blockchains - decide which one is better for your investment needs", URL https://data-flair.training/blogs/types-of-blockchain/.

[27] Wang, Y., J. H. Han and P. Beynon-Davies, "Understanding blockchain technology for future supply chains: a systematic literature review and research agenda", Supply Chain Management: An International Journal .