HyperLedger Fabric Induced Supershop Supply Chain Management Model for Securing Product Authenticity

by

Md Solaiman Hossain
18101202
Nafid Faham
18101107
Sakib Hassan Chowdhury
18101127
Adel Rahman
19101667
Mohammed Fazle Mubin
18101055

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

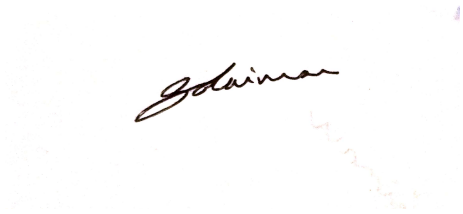
Department of Computer Science and Engineering
Brac University
May 2022

Declaration

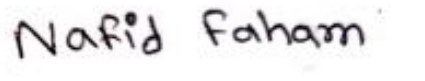
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

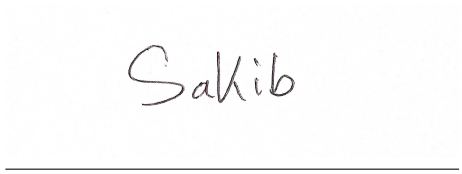
Student's Full Name & Signature:



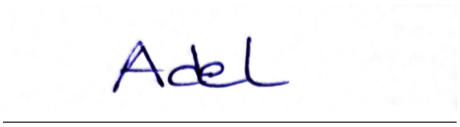
Md Solaiman Hossain
18101202



Nafid Faham
18101107



Sakib Hassan Chowdhury
18101127



Adel Rahman
19101667

Mohammed Fazle Mubin
18101055

Approval

The thesis/project titled "HyperLedger Fabric Induced Supershop Supply Chain Management Model for Securing Product Authenticity" submitted by

1. Md Solaiman Hossain(18101202)
2. Nafid Faham(18101107)
3. Sakib Hassan Chowdhury(18101127)
4. Adel Rahman(19101667)
5. Mohammed Fazle Mubin(18101055)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 29, 2022.

Examining Committee:

Supervisor:
(Member)

Muhammad Iqbal Hossain, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Blockchain is a relatively new technology but has so much to offer. That is why in order to maintain proper security and controlled transparency to Supershop System we are going to induce Hyperledger Fabric System in our blockchain platform. This paper will include the use of blockchain technology for the Supershop System, and different research works to back up the technology we are using. The diagrams and flowcharts are given in the paper to help understand the making of the system in a pictorial manner. The different hardware and prerequisites needed are mentioned in the paper along with the description of the baseline implementation of the system. This paper also shows how the network is made and how multiple organizations may fit in this system. After the creation of a proper network with the Hyperledger Fabric system, this paper shows that it is not only possible to make a Supershop System with the help of a blockchain but also make it more secure and maintain authenticity of products. The work of this research is to ensure the consumers absolute authenticity by maintaining transparency in our Supply chain management with the help of blockchain technology. The main goal is to make sure that the product passed from the manufacturer to the distributor then to the retailer finally to the consumer is authenticated by the blockchain. A central authority will be managing the quality control of the product. The main actors of the supply chain will have their digital signature which will be checked and transactions will be approved through a smart contract when needed. All of the information regarding the product transaction between these actors will be recorded in the ledger. This will help to make sure that the chain was authentic and to check any alteration of the product. The consumer will have the opportunity of checking the authenticity of the product as well through the chain authenticated and recorded in the ledger of the blockchain. To sum up, this paper will show how the supply chain management of a Supershop can be built with the help of HyperLedger Fabric for Securing the Product Authenticity.

Keywords: Blockchain; HyperLedger Fabric; Supply Chain; Smart Contract; Supply Chain Management

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Dr. Muhammad Iqbal Hossain sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$B2B$ Business to Business

$B2C$ Business to Customer

$C2C$ Cash to Cash

DA Distributed Application

DFS Distributed File System

EMR Electronic Medical Record

FDA Food and Drug Administration

IBM International Business Machines

KPI Key Performance Indicator

MSP Membership Service Provider

PHR Physical Health Record

PoS Proof of Stake

PoW Proof of Work

SCM Supply Chain Management

TPS Transaction Per Second

Chapter 1

Introduction

1.1 An Intro to Blockchain Technologies

Blockchain has been popular in recent times. It is a system that is of recent origin but the use of it is very significant. Many applications take real-time data and usually these used to be shared and stored on centralized systems. The clouds and database for storing the information of the customers and employees and even the interaction and transaction between the customers and the employees need safekeeping. In Blockchain, there is basically a ledger where the given information is collected and stored. However, this stored data is then secured with some sort of hash encryption (SHA256 for example) and once it is encrypted and stored with the necessary information of the sender, the information becomes unchangeable. This is a big reason why using blockchain is so secure than other systems of data storing. There has been a lot of research within the last decade about implementing blockchain on such systems for decentralized access controls, immutability, and security. There are so many uses of blockchain but in recent years its use has started to spread in common systems for example in supply chain, database security, cloud security, etc. However, such systems are prone to cyber-attacks as well. The system will be secured at the gateway level with Blockchain and will be decentralized so that even if one organization is breached, it will not affect the other organizations. For the data management system, we will make sure that the data is encrypted and again decentralized so that any sort of attack can be easily tackled and avoided with the use of Blockchain technology.

1.2 HyperLedger Fabric on Supershops

In the case of systems like Supershops, which have sensitive data and daily transaction information need proper security and privacy which is why blockchain is an important necessity in the modern world to ensure confidentiality and scalability. Since there are many vendors in a supply chain of a Supershop there will be a risk of a data breach if all the information is kept centrally. By building a proper network through blockchain technology it is possible to maintain the different vendors and the related sensitive information.

So, we will be using Hyperledger Fabric instead of other traditional blockchain systems as it is open-source and easier to implement. Moreover, Hyperledger is cheaper

than other blockchain alternatives as well as offers a decentralized security system that offers privacy to the user. As Super Shops are basically business organizations, there are also competitions where there is a lot of information that is needed to be within the organization.

This paper will be based on making a safe and secure network made of blockchain to run a Super Shop. There are many vendors who act as different organizations from the view of Blockchain. They will be given different roles in the network so that they are not centralized. The information of the supply chain of the Supershop will be stored in the blockchain and the organizations can be controlled with smart contracts through Hyperledger Fabric. This will basically make a secure supply chain network with the help of Hyperledger Fabric.

1.3 Research Problem

An ever-growing problem for any brand is counterfeiting. Almost any industry is immune to its threat and super shops are no exception to it. The global cost of counterfeit goods is rapidly increasing every year causing a severe impact on the revenues and as well disturbing the safety of the consumers. Therefore, to ensure the authenticity of a product is of great importance in a super shop model. The supply chain in a super shop largely depends on its producer and distributor which is why the problem of super shop chain management is gaining more and more importance recently given the recent attention for quality and safety. The major problems are to ensure product traceability, which requires a comprehensive perspective of the many stages of distribution starting from harvesting. The majority of today's supply chains, on the other hand, are managed by centralized systems which means it depends on a centralized authority for sharing any kind of information. These systems are usually non-transparent posing a serious threat to the security and reliability of the traceability data. As the systems are monopolistic in nature, it is really easy to make fraud and corrupt data while having the issue of not being scalable. For dealing with such issues, blockchain technology can play a vital role. In our case, we are using the Hyperledger Fabric based blockchain framework which is based on a decentralized network. This eliminates the chance of being highly dependent on a single centralized entity. As the data is decentralized, it is really easy to find the manipulated data or product information as all the other entities in the chain will have its information. Moreover, as our network is distributed, the problem of scalability can also be solved by means of using this technology. Our model highly depends on a smart contract, which automates a lot of the things for us which makes us easier to maintain traceability data while maintaining the authenticity of the products in a supply chain. This is where our model makes a huge difference to counterfeiting in a supershop supply chain.

1.4 Research Objectives

The main aim of this research is to create a model of a blockchain based supershop supply chain management system using Hyperledger Fabric which will provide traceability to the system and allow us to implement the applications of blockchain

technology on this wide domain. We have used Hyperledger based blockchain fabric as it is a private permissioned blockchain and most importantly open sourced. The proposed model will allow the members of the system to store and manage product related traceability data in a distributed way. As the whole model greatly depends on the smart contract, the system will be able to store and manage traceable information accordingly. Another aspect of the research is to maintain the quality and integrity of the products along with its safety as it is an important element of the supply chain. Moreover, the model that we aim to create would be vastly scalable which means we would be able to deploy this system prototype in any production ready environment without losing any authenticity of the products. This model will evolve over time dynamically as more components are added to it and it is one of the main keys of our research model as super shop chains gradually increase. Thus, our main objective is to create a scalable, traceable, efficient, and cost-effective system where the authenticity of products of a supershop system is maintained without having a possibility for the data to get tampered with. The framework of the system and the methodologies are further discussed in the following sections.

Chapter 2

Literature Review

2.1 Related Works

Using human-centric computing, Yin et al. [10] examined and assessed local privacy protection. They combined privacy protection with machine learning to achieve task categorization using logistic regression in their work. The goal of using noise addition is to limit the disturbance of the original data while also safeguarding the data's privacy. The goal of feature selection is to highlight the influence of noisy data on the classifier so that it can generate a larger contrast with the original data that is noise-free. Furthermore, they employed the Laplace mechanism to introduce noise to the original data in order to accomplish the purpose of disturbing data. This strategy, however, is less successful if there isn't enough training data to develop a machine learning model.
Another research paper written on the topic of a customer data-sharing platform with the help of blockchain was helpful for our research. In this paper, the main purpose was to make a platform where the customers and the sellers can connect with each other. This platform would have solutions for these main problems: private payment, ensuring privacy and user control, and incentives for sharing [13]. This was mainly done to make sure that the interaction of the customer and the seller is always honest and transparent to solve any sort of dispute claims. The use of smart contracts helps to make sure that the data and transactions of the customer are always checked and to honor them with a reward system as well. Moreover, this system tracks the dealings and interactions of the sellers and customers with detailed information too. The private blockchain Multichain securely stores these streams of data and efficiently and with much ease processes it through other nodes where required [1]. There is data sharing in the private blockchain but not the public one for a better data sharing process. There is the use of Ethereum blockchain and the different functionalities of similar systems used [8][7][9]. There is some very important knowledge we get from this research but mainly how to handle a system that secures the data of the customers and sellers and stores other information of their interaction with the help of smart contracts and blockchain. In their research paper, A.Basak et al.[16] have studied the supply chain management system centered around Bangladesh. They stated that the supply chain management system is very beneficial because of Supershops. The authors also mentioned that the supply chain is also very important in the market as well. The customers of the Supershops are generally the ones who are the final consumers and the different medium and

contributors who are responsible for bringing the product to the customers pay key roles in the supply chain. The supply chain management system mainly starts from the supplier who supplies the raw products to the manufacturer. Then the manufacturer uses the supplied product to make the desired product and gives it to the distributor. The product afterwards is distributed to the retailers and finally to the customers. This finally completes the supply chain system. The management of this supply chain helps to bring out any sort of flaws or errors in the chain. The customers buying from the Supershops can also see the chain to know how and where they are getting their products. In our research paper the basics of the supply chain system is very important. To understand how a supply chain works in case of a Supershop system this research paper of A.Basak et al. will help. The use of each contributor in our supply chain can be clearly realized with the help of this research paper.

In another research paper, we can see that Patrick Sylim et al.[4] has planned out a system where the FDA is placed in the role of a supervisor, whereas the customers can simultaneously view the distribution history of the drug by scanning a given printed code in their proof of purchase slips. Within this scenario, two occurrences are created. The first instance is generated in Ethereum along with a Proof of Work (PoW), meanwhile another instance is created in the hyperledger fabric platform. They use a Distributed File System (DFS), Swarm, which contains the Distributed Application (DA), smart contract as well as the blockchain. 5 starting points, namely the manufacturer, wholesaler, retailer, FDA, and lastly the consumer are all involved in this system. Moreover, another function of this system is to detect 5 different inconsistencies that include oddities in timestamps, omission of nodes, and dissimilarities in the data points. This is helpful for our research since our supershop needs to manage a smart contract with its organizations as well as being able to simultaneously view the distribution histories tied to them while at the same time trying to make sure there are no discrepancies in either the timestamps, the nodes and the data points. In addition, we have to make sure that the first instance does not need to be created in our research paper.

Moving onto the next research paper by Jennifer Cristina Molina et al.[6], the idea of a system is brought forth which ensures minimum privilege using the hyperledger technology of blockchain. The actors who are involved in the pharmaceutical supply chain are provided with a ledger to record the process parameters. Furthermore, each transaction that is being done only includes two actors, which later on including both the details of the transactions as well as the actors are then uploaded to a web portal. This is done so that transparency is maintained across all the transactions. This will help our research since, at the end of the day, our supershop has to maintain the transaction transparency across all the organizations it is connected with.

In [14], the authors explained and compared some existing works on implementing Hyperledger in hospital information systems. They first compared the different architectures of Hyperledger like Fabric, Sawtook, Iroha, Indy, Besu then described the motivation behind using Hyperledger for Hospital Information System which are the integrity of medical information, data privacy, data control, interoperability, real-time traceability of data etc. Then, they explored the existing medical systems based on Hyperledger, summarized those works and defined four research directions including medicine traceability, medical records (EMR and PHR), medical images and other medical fields like financial incomes, medical insurance, system

performances. We also will use one of the architectures of Hyperledger which is Hyperledger fabric to use in our Supershop for securing transaction records as well as product information and maintaining a secure supply chain management system.

Chapter 3

Background Analysis

3.1 Blockchain

A blockchain is a secured technology for storing data in a pattern of chains. Data in a blockchain is unchangeable due to its encryption nature. According to [2], in 2008 A person or a team named "Satoshi Nakamoto" published the paper named "Bitcoin: A peer-to-peer Electronic Cash System" which specified a method of sending cash directly from person to person without any involvement of a 3rd party medium. It was first applied in cryptocurrency.
Blockchains are used for any sort of transaction whether tangible or intangible. This is a technology which stores data and information in an immutable ledger. This ledger cannot be accessed by anyone other than authorized members. Blockchain technology is used very often to keep information and data safe and secure. The data are kept in a decentralized system which makes it more secure than any other system. The data are stocked in different blocks which are connected with each other which resembles a chain like system. This chain helps to keep connection with the previous block. This connection is kept throughout the block system and creates a blockchain. These blocks are very well built to make sure that the data stored are encrypted with some sort of hash function. An example of this is the hash algorithm used for Bitcoin called SHA256. This connection among the blocks are shown in the diagram below:

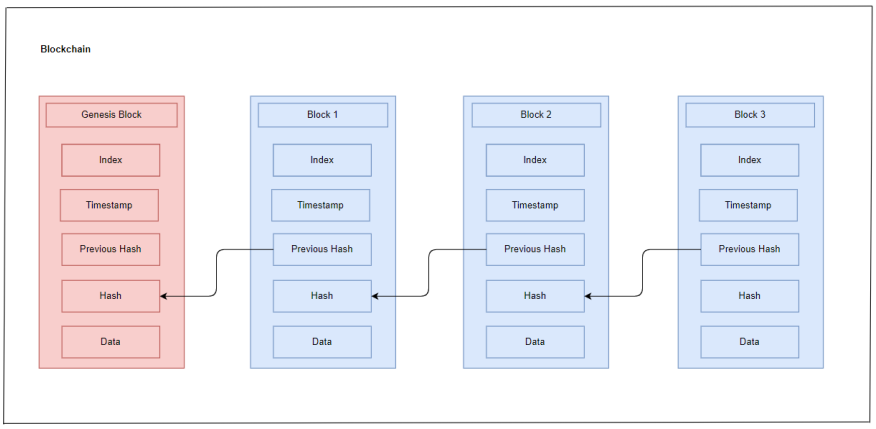


Figure 3.1: Blockchain Technology Diagram

From this diagram we can see that there is a block called Genesis Block. This block is the block 0 as it is the first block of the blockchain. Every chain contains an index, timestamp, previous hash, hash, and the data for storage. The hash is the main encryption function to secure the block. A block stores a hash of itself and also the hash of the previous block. This makes sure that the previous block is unchanged. If the hash of block 2 and the hash stored by block 3 of block 2 matches then it means that block 2 is unchanged. If the block does not match, that means the block is changed. The Genesis block is different from the other blocks because there is no previous block for storing in this case. That is why it is called a Genesis Block as it is the starting block in a blockchain.
There are some key features that the blockchain technology contains. Firstly, this technology is decentralized. This means that if a blockchain network is built then that network will not have any center point. The main intention of this decentralized method is to make sure that if one part of the network is compromised all the others will not be affected. The diagram below shows how a centralized network and a decentralized network are different:

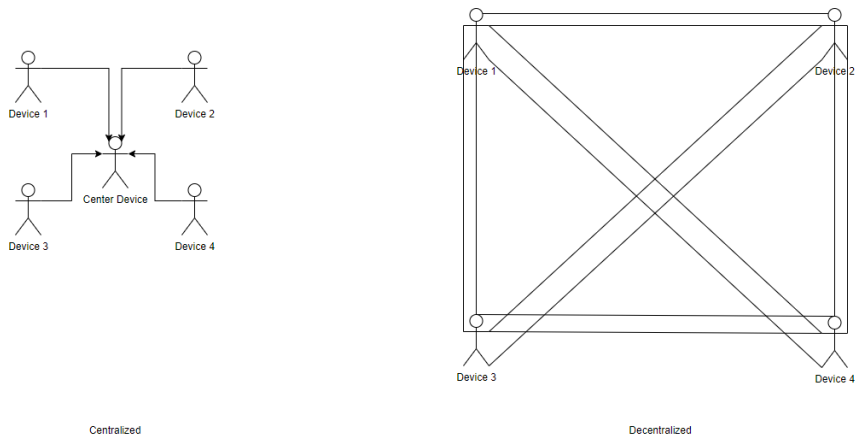


Figure 3.2: Decentralized Network

As shown in the figure 3.2 we can see the difference between centralized and decentralized networks. The centralized network has devices which are connected to a central device. The decentralized network is not connected to a central device rather it is connected individually with the other devices. The network of the blockchain has some organizations or participants which can get access to certain data of the network. The data are distributed in the blockchain as a ledger among these organizations. The authority given to the organizations decides their permission in the block chain. The part of the ledger that the organization has the permission to can be accessed by them. Any changes made to the blockchain will be recorded which makes the blockchain immutable. This helps to create a peer-to-peer relationship in the blockchain network. Since the blockchain is immutable and decentralized, there can be no 3rd party involvement. The transaction of the information is completed among the organizations itself as shown in figure 3.3.
A very important feature of the blockchain is Smart Contract. A smart contract is basically a program made to work as a contract to fulfill the agreement between

two or more parties when a certain threshold or condition is met. An example of a smart contract may be the distribution of data among the participants in a supply chain when the product has reached the store. This is done to make sure that no extra party is involved. The figure below shows how a smart contract work:

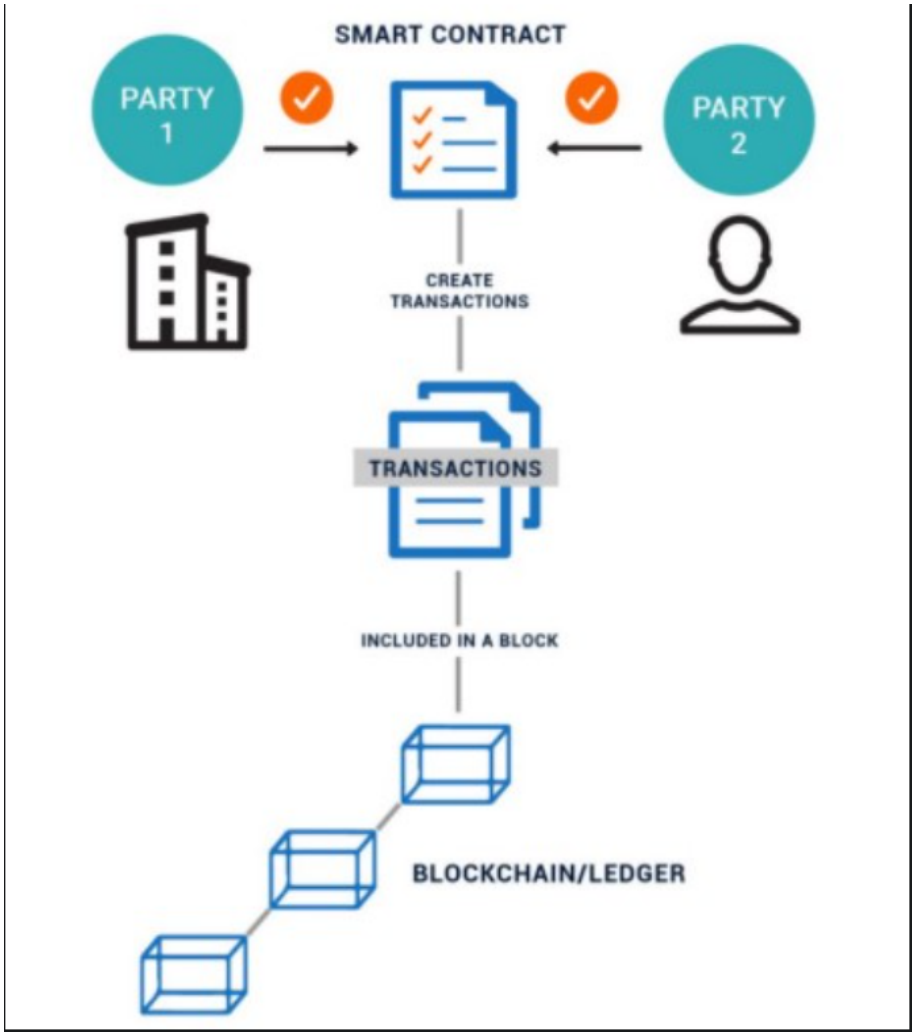


Figure 3.3: Smart contract

As shown in the figure 3.3, we can see that the smart contract works between two parties or more. When the conditions are met the smart contract is automatically invoked and a transaction is created. The transaction depends on the smart contract between the parties. This transaction is also registered in the blockchain.
In order to secure the blockchain, a hash function is normally used. However, the hash encryption can be broken with high computation power. That is why two new methods of consensus mechanism were introduced. The proof of work and proof of state are the new mechanisms. The proof of work is a system of protecting the blockchain where the next block maker is chosen. The opportunity is given to all the participants who are called miners. The first miner to solve the encryption problem gets the opportunity. This requires more power and electricity consumption. A problem in the proof of work is the possibility of being centralized because of the mining pool made by the combined effort of miners. The flowchart shows how the

proof of work algorithm works:

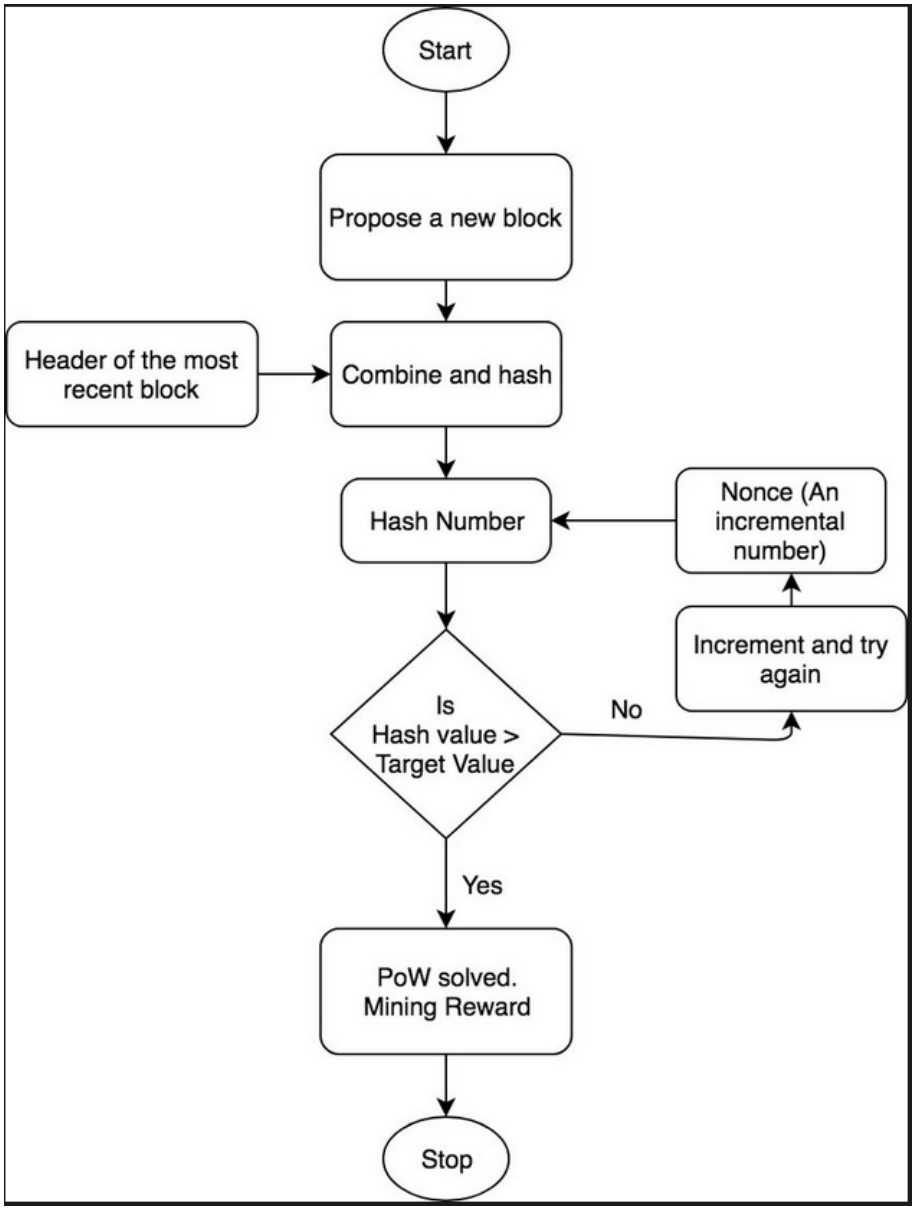


Figure 3.4: Proof of Work

The proof of state is a mechanism which solves this problem. The proof of state chooses a random validator who gets to mint or forge the next block. In order to make sure that the chance of being chosen is greater, the validator needs to deposit a substantial number of coins. If there is any sort of deceit detected in the validators action then a part of the deposit is taken away depending on the deposited amount. The flowchart below explains how the system generally works:

These are the algorithms used for securing the Blockchain system but the PoW and the PoS algorithms are not fully secure. The centralization issue of the PoW is a big issue for this algorithm. The chance of rich validators getting more opportunities makes the algorithm an issue as well. That is why both the algorithms are not perfect and with flaws.

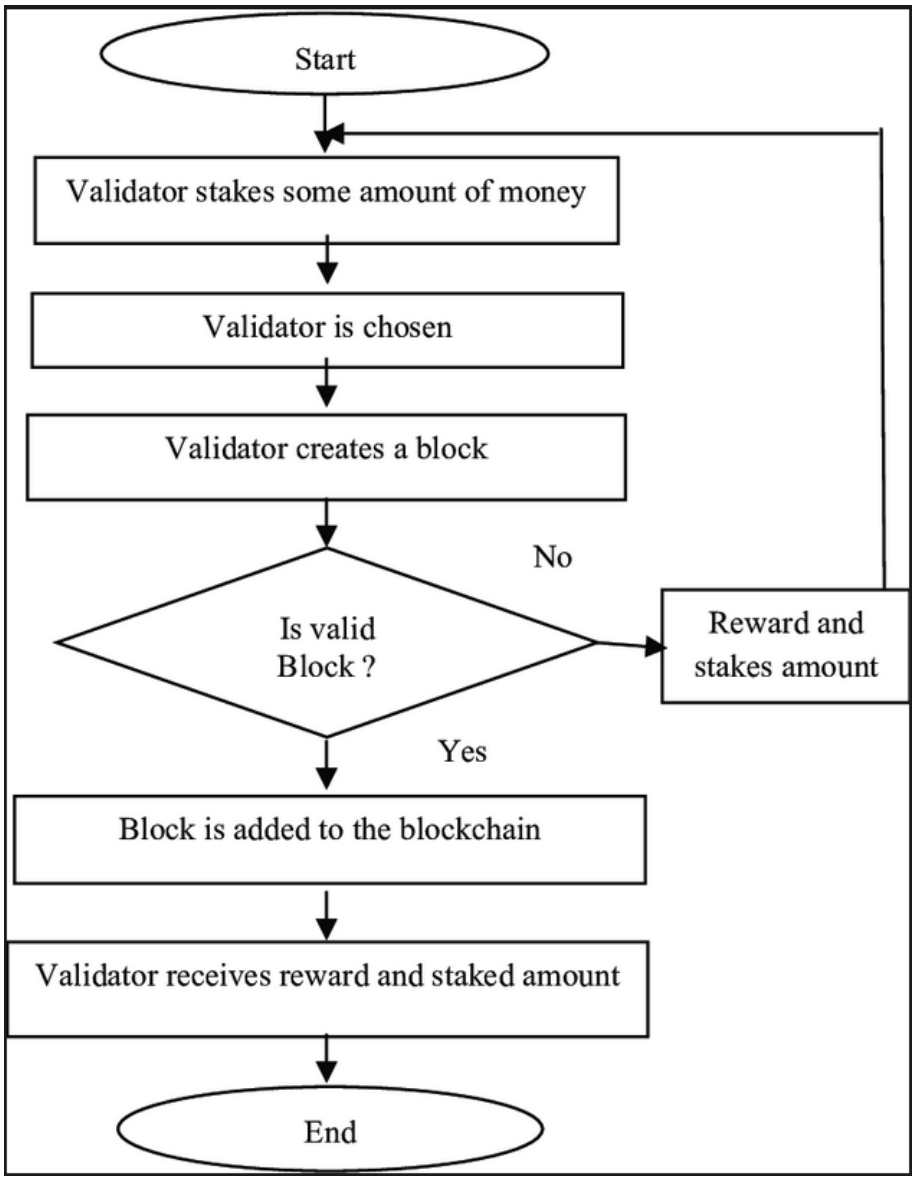


Figure 3.5: Proof of State

3.2 Hyperledger Fabric

Hyperledger Fabric is an open-source development project by Hyperledger Blockchains company. As the name indicates it has ledgers. Moreover, it includes smart contracts as well as protocols that aid people to engage in the transaction process. Furthermore, not any user can get access to the Blockchain network since the Fabric architecture is permissioned and private. Only by being a member of the Blockchain network can a user get access to the system. In a PoW algorithm, the consensus is reached by mining yet this does not happen in a Hyperledger. Adding further into a Fabric architecture being permissioned and private, to conduct private and sensitive dealings, a private subnet, Channel, is created between two or more blockchain nodes of a particular network where all the users are validated, and only then can

they perform transactions [17]. For this section, we will list below all the key components of a Hyperledger Fabric:

3.2.1 Key Components Of Hyperledger Fabric

Shared Ledger

As stated in [15], a ledger actually holds a list of transactions for a company, where the list represents the present condition of a business. Through the use of a Hyperledger, we satisfy two main concerns of our supershop:

- To highlight the list of our ledger states along with their current values.
- To record the sequence of transactions that led to these states.

In other words, the facts about the current states as well as the records are stored in a ledger. Thus, even if the facts of those current states are altered, the same does not hold true for the records themselves since, in a ledger, a record can be added yet it cannot be changed. (Figure 3.6)

In a Hyperledger Fabric, the ledger itself is divided into two sub-components which comprise a set of facts towards a range of business objects. The two subcomponents are described below as such:

1. **World State:** As stated in [15], a ledger's status at any particular time can be easily defined by the world state. In simpler terms, the world state basically acts as a ledger's database. Since ledger states are recorded as key-value pairs, with the help of the world state, we can easily determine the present value of a state rather than searching through an entire backlog of transactions.
2. **Blockchain:** A blockchain, otherwise known as a transaction log, is an archive that holds all the changes in transactions that were made in the current value of the world state. These transactions are stored inside blocks that are attached to the blockchain which in turn helps us recognize the past changes that have occurred in the present world state. This is because the blockchain data structure is comparatively different compared to the world state since the value a blockchain holds is not possible to be modified.

Usually, it is thought that a Hyperledger Fabric network consists of one logical ledger. However, in practice, the network manages several instances of a ledger. Through the process called Consensus, those instances are kept constant.

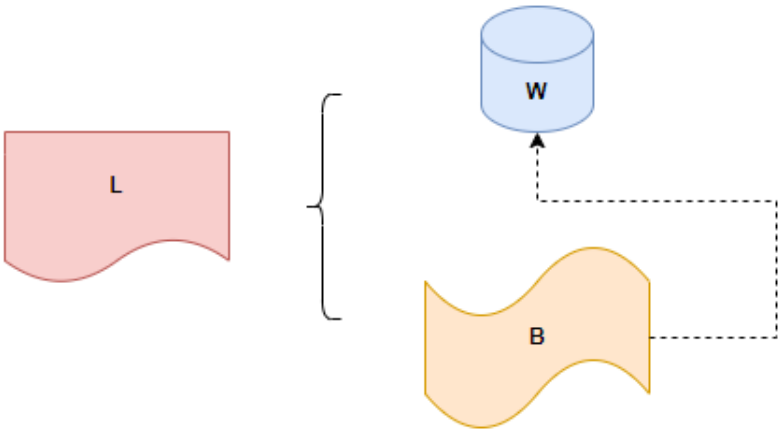


Table	
L	Ledger
W	World State
B	Blockchain
W, L, B	L comprises B and W
W, B	B determines W

Figure 3.6: Shared ledger

Consensus

Transactions may occur between a separate set of participants in a network, nonetheless, they must be recorded in the ledger in the sequence they first transpired.

However, in order to implement this action, the previously mentioned sequence of transactions must be put in place, along with a procedure to deny the faulty entries that have been placed in the ledger, either by mistake or through deliberate intentions. Hyperledger Fabric was designed in such a way to allow network creators to select a consensus mechanism that best portrayed the relationship between the participants involved. One way to achieve the consensus is through the use of PBFT (Practical Byzantine Fault Tolerance), where file clones within a network can share information with each other in order to maintain their coherence in the event of tampering being found.

Peers, Applications, Channels, and Organizations

Peers play a huge role in a blockchain network since they host both the ledgers as well as the smart contracts within that network. Since both smart contracts and ledgers hold the shared processes as well as the information within a network, we can understand why peers hold such a critical role.
Whenever applications need to access ledgers and chain codes in a network, they always require a connection to the peers in that network. This is because applications, through the use of peer connections, can run chain codes to search or update a ledger.
Channels are what allow multiple peers within a network to communicate with each other or with applications. In other words, a channel acts as a means for a group of blockchain network components to communicate and operate transactions in private.
Rather than being managed by a single organization, blockchain networks are administered by a group of organizations. Peers play a crucial role in the construction of a distributed network since they are maintained by these organizations and serve as their network connection points. Figure 3.7 shows how peers, applications, channels and organizations work together in a blockchain:

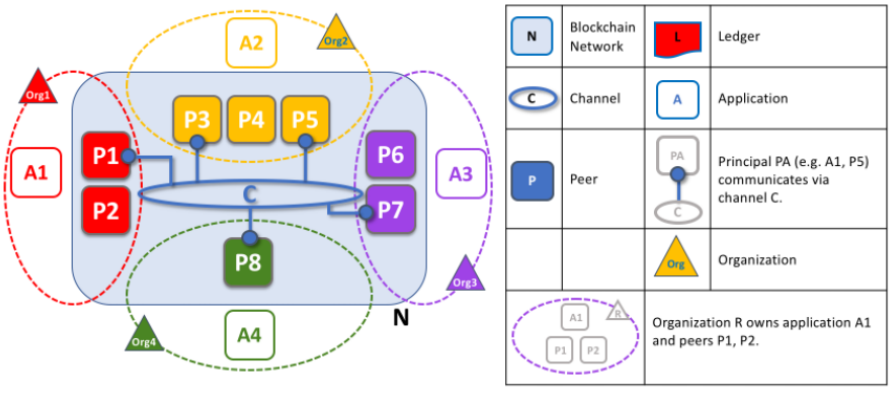


Figure 3.7: Peers - Applications - Channels - Organizations Structure

In the diagram, we can see 8 peers (P1, P2, P3, P4, P5, P6, P7, P8) being under their respective organizations (Org1, Org2, Org3, Org4). P1, P3, P5, P7, and P8 are connected to a channel, C. Applications (A1, A2, A3, A4) produced by their respective organizations will link to the peers of that organization as well as to those

of other organizations. For transparency, no order node has been included in this structure.

Smart Contracts

In Hyperledger Fabric, we often use the term smart contract and chaincode in tandem. In summary, a smart contract is used to describe the transaction logic which manages the longevity of an object in a business in the world state. The transaction logic is then compiled into a chain code, later on being distributed to a blockchain. Thus it is possible for several smart contracts to be defined by the same chaincode.

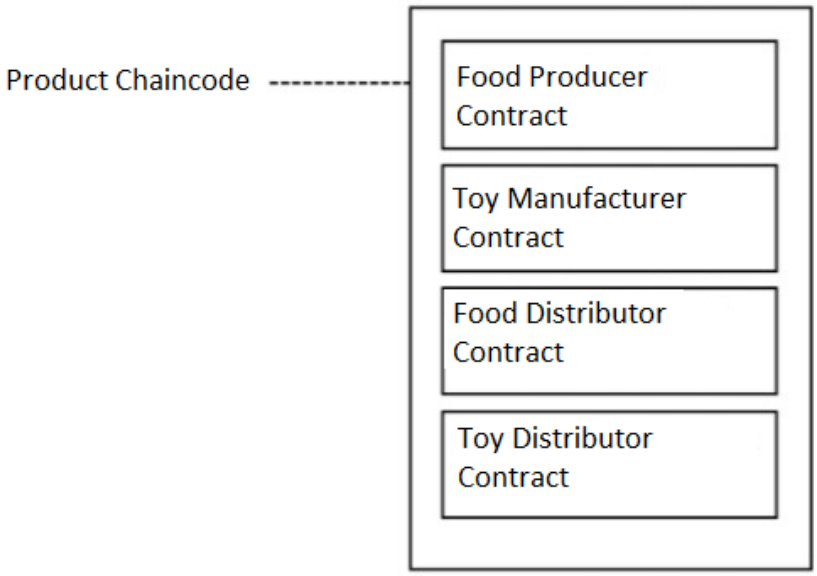


Figure 3.8: Smart Contract and Chaincode

In Figure 3.8, we can see our product chaincode connected with four smart contracts. From the diagram, we can observe that a smart contract is a program tied to a specific domain, whereas a chaincode acts as technical storage for a collection of connected smart contracts.

Public Key Infrastructure

Public Key Interface, otherwise known as PKI, allows for the creation of secure connections in a blockchain network. One example of the use of PKI is when we are accessing any content from a web page, we are probably encrypting it with a PKI to ensure that the content is from a trusted source since the PKI is responsible for the S in HTTPS. Figure 3.9 shows how a PKI workflow is created using its 4 components:

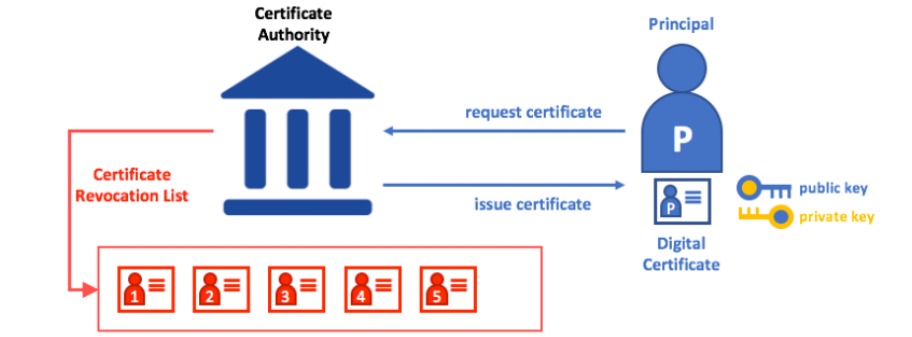


Figure 3.9: PKI Workflow

As previously mentioned, PKI is made up of 4 main components:

1. Digital Certificates - It is a document that contains information about a certificate's holder.

2. Public and Private Keys - For secure communication, a private key is used to create a signature on a message that can only be verified by the corresponding public key along the same message. As shown in Figure 3.10, we can observe that Mary is using her own private key to create her signature on the message that can only be verified by someone who has access to her public key.

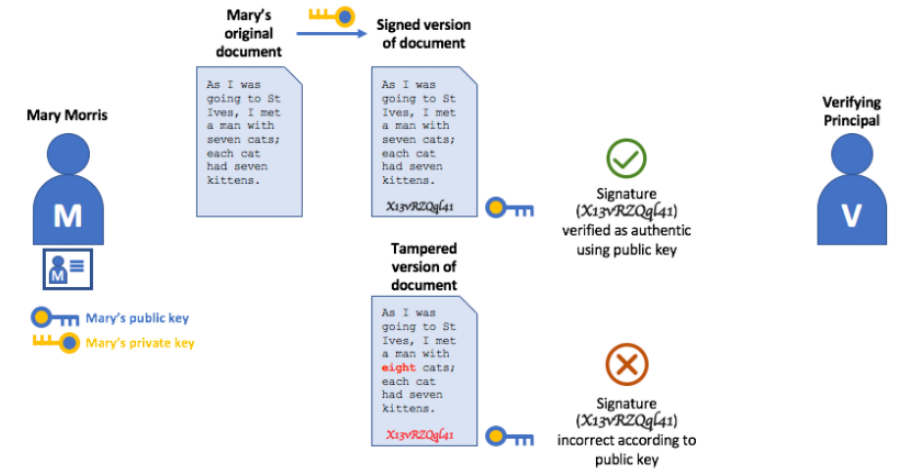


Figure 3.10: Implementation of Public and Private Key

3. Certificate Authorities - Digital certificates are issued by a Certificate Authority to several actors. These certificates help link the actor to the actor's public key. Figure 3.11 better explains this concept:

Figure 3.11: CA issuing Digital Certificates

4. Certificate Revocation Lists - Also known as CRL, it is a list of pointers to certificates that a CA is aware have been canceled for various reasons. Put simply, it checks whether a certificate is invalid or not.

Identity and Membership Service Provider

Identity in a blockchain network helps establish the various actors involved that include peers, orderers, client applications, administrators, etc. The identities are digital in nature and are represented in a digital certificate. These identities are crucial because they decide which participants in a blockchain network have specific privileges over resources and availability of information. Usually, in a blockchain network, we require a trustworthy authority to verify and authenticate member identities and roles which is easily achievable through Hyperledger Fabric since that trusted authority is in the form of a component known as the Membership Service Provider (MSP) [41]. MSP outlines the principles that regulate an organization's valid identities.

3.2.2 Core Features of Hyperledger Fabric Model

In this section, we will be noting down the core features that led us to choose hyperledger fabric in addition to utilizing the key components mentioned above, to optimize and manage the authenticity and transparency of our supershop's supply chain.

Endorsement

An endorsement policy is attributed to each chaincode and applies to all of the smart contracts established therein. An endorsement policy is critical because it specifies which entities in a blockchain network must sign a transaction generated by a certain smart contract before it can be deemed valid. Organizations can further classify identities into clients, admins, peers, and orderers using the typical MSP implementation.

- Client - An identity that transacts on a blockchain network.
- Admin - An identity that lets a peer join a channel or sign a channel configuration update transaction.

- Peer - An identity that endorses or commits transactions.
- Orderer - An identity that acts as an ordering node.

Fabric Architecture

Having been designed with a modular structure in mind, Hyperledger Fabric can fulfill a wide range of organizational use case needs [15]. Some of Fabric's modular components that were helpful in our research are listed down as follows:

- A configurable ordering service generates transaction order consensus before forwarding blocks to peers.
- Programmable MSP is able to issue cryptographic identifiers to network entities.
- Various forms of databases can be set up by a ledger.
- Smart contracts can now be programmed using the standard programming languages.
- Availability of a peer-to-peer gossip service that forwards the blocks generated by the ordering service to other peers [15].

Fabric Permissions

Hyperledger allows the use of permissioned blockchains where the blockchain network operates among a small number of well-known, traceable, and regularly verified participants who are all constrained by a governance architecture that provides a high level of trust [15]. Through recognizing an actor's identity, a permissioned user can implement the standard Crash Fault Tolerant (CFT) or Byzantine Fault Tolerant (BFT) consensus algorithms without requiring excessive mining costs.

Execute, Order, and Validate

A new featured architecture introduced to Fabric. By splitting the transaction flow into three phases, it overcomes the order-execute model's robustness, flexibility, scalability, performance, and secrecy issues [15]:

- Execute a transaction order to verify its accuracy before endorsing it. Moreover, eliminates any non-determinism as inconsistencies can be cleared out before placing an order.
- Arrange transactions with the use of a (programmable) consensus protocol.
- Before admitting transactions to the ledger, review them against an application-specific endorsement policy.

Fabric Confidentiality

Hyperledger Fabric introduces the idea of channels, which enables private transactions between peers or members of the same network. Only nodes in that specific network may see the transactions that are taking place when a channel is used. Participants of a channel can create collections utilizing private data, which offers similar privacy to channels without the effort of establishing and managing a separate channel. As a way of either reinforcing a connection or creating new sales, some consumers may be offered preferred pricing in a network of supply-chain partners. If every member in that particular blockchain network can see each and every contract as well as transaction, it becomes impossible to maintain such commercial relationships in a totally transparent network because everyone wants the best costs. Hyperledger Fabric assists us in avoiding issues like this, allowing us to retain the openness and authenticity of our superstore.

Hyperledger Performance

Transaction processing and transaction ordering are managed differently in the system so that they can both operate separately at the same time. Additionally, since transactions are required to be completed before placing an order, all peer nodes can process transactions simultaneously. The hyperledger fabric network is substantially more reliable than other blockchain systems due to its scalability (somewhere between 2000 transactions per second). This is advantageous since we can now extend and add more actors to our blockchain without compromising its performance.

3.3 Supply Chain Management

3.3.1 Supply Chain and Supply Chain Management(SCM)

A supply chain is a network between a company, its suppliers and the end-users which consists of all the steps from taking the raw materials from the suppliers to the final delivery of the product or service to the customers. It outlines all elements of the production process including the actors and actions performed and information exchanged between them at each stage or between different stages as well as the necessary materials for the final product or service that are converted from natural resources. There are two kinds of supply chains, internal and external.

- **Internal Supply Chain**

  Internal supply chain is the part of the supply chain that happens inside an organization. Many international companies are multidivisional, each division playing different roles. In these scenarios, the divisions see each other the same way they would see the external organizations in a supply chain.

- **External Supply Chain**

  External supply chain is a traditional supply chain that usually everyone knows of. Here, the actors in the supply chain are not of the same organizations.

A supply chain management or SCM, on the other hand, is a management of the materials, products, data and finances related to a product or service throughout the supply chain. It is vital for maintaining an optimized supply chain to ensure faster production, preventing supply shortage and control and reduction of the costs.

3.3.2 Supply Chain Models

There are different types of models a company can choose to fit their supply chain into depending on the structure and needs of the company and the business model it is following. However, every model has the same focus which is improving the workflow, efficiency, and responsiveness.

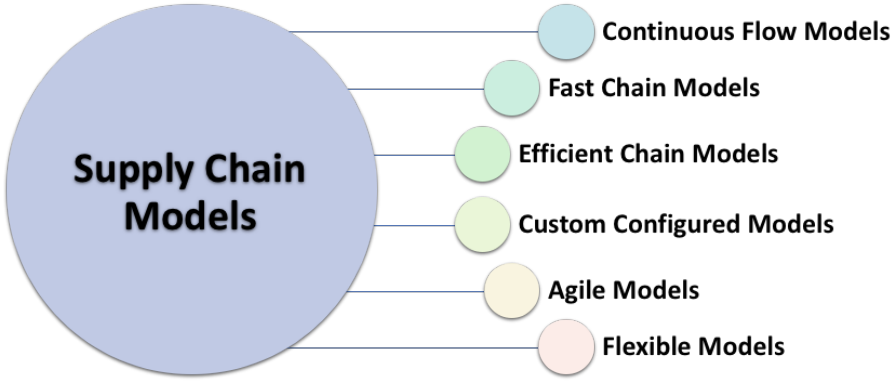


Figure 3.12: Supply Chain Models

- **Continuous Flow Model:** This is a traditional model which works best for mature companies that produce and sell the same products. The brands have to be popular and little to no redesigning should be required for the products. Managers in this model need to refill the raw materials regularly to prevent production bottlenecks.
- **Fast Chain Model:** As the name indicates, this model works best for markets that are fast paced. Companies that offer products that have a short lifecycle and are based on trends like fashion products more likely use this model. They take advantage of the current trend with this model by getting their items to market as soon as possible and speeds up the process from concept of the product to prototyping to manufacturing to final product to the consumer.
- **Efficient Chain Model:** This model is ideal for highly competitive markets where end to end efficiency in delivery logistics is required and price is a major factor. This model emphasizes on how well the manufacturing equipment and people have been used and how efficient the inventory management is. This model also relies heavily on commodity and prices of raw materials.
- **Flexible Model:** The flexible model is frequently used by industries who produce seasonal or holiday items that have a high degree of consistency and a few predicted demand peaks. This model allows them to quickly ramp up production and efficiently shut down when demand drops. So basically, this model tries to be the best of both worlds. It can respond to high volume demands during peak season as well as handle the times of low or no demand.

- **Agile Model:** This model works best for companies that face unpredictable demands which need to be met quickly and efficiently. The main focuses of this model are speed, cost efficiency, being responsive to the consumer demand and the market, maintaining flexibility and high productivity. The four components that a supply chain must have to be deemed agile are-
  - Virtual integration,
  - Process alignment,
  - Network base and
  - Market sensitivity.

  Virtual integration necessitates real-time monitoring of market demand changes, process alignment means sharing supply chain responsibilities across the organization, network-based means every actor contributing equally in the supply chain and the market sensitivity modifies the rate of production instantaneously whenever any changes come in demand.

- **Customized Configured Model:** A custom-configured model which is a combination of agile and continuous flow models, is useful in instances where there are options for consumer customization or multiple configurations of a product is required especially during assembly and production.

3.3.3 Process of Supply Chain Management

The whole supply chain consists of many actors, components/phases and drivers which are key parts of SCM. Without these, a supply chain cannot be complete and effective. The basic components, drivers and the whole process are explained below.

Core Components and Drivers of Supply Chain Management

For SCM to work steadily and flawlessly, the whole supply chain needs to be divided into several roles and processes which can be considered as the components and are followed respectively. There are five basic components of a traditional supply chain-

1. **Planning:** This is one of the most important stages as It is essential to plan and manage the resources necessary to meet a company's demand for a product or service. In this step, the demands, costs, profit and manpower need to be checked and predicted by the supply chain leaders. Without proper planning, it is highly impossible to maintain effective and long-term profit in a business. So, all the pros and cons of the other next steps depend on how efficiently it was planned.

2. **Sourcing:** Following planning, is the sourcing in which the required resources need to be purchased. This is the step which plays a vital role in saving costs. Choosing the right vendors or suppliers to supply the suitable raw materials on time and within the budget while ensuring the profit is the main focus here. This step is also called the Procurement stage.

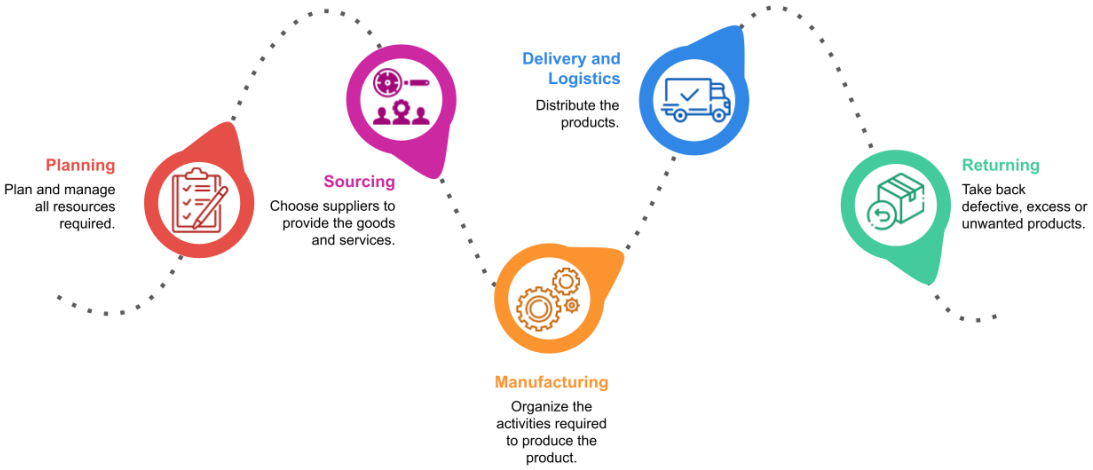


Figure 3.13: Supply Chain Management Phases

3. **Manufacturing:** In this stage, the production of the final product is done. This stage focuses on productivity and efficiency. Here, necessary activities are organized for taking the raw materials from suppliers, creating the product, conducting quality tests, packaging it for shipping and scheduling the delivery to maintain the integrity of the product. Supply managers need to make sure all of these steps are taken carefully.

4. **Delivery and Logistics:** After manufacturing or production, comes the delivery and logistics where basically the final product is delivered to the customers. In this stage, the companies focus on maintaining the brand image by ensuring positive customer experience and the timeliness and seamlessness of the delivery. This stage is also necessary when delivering the product to other actors like supplier > manufacturer > distributor > retailers. The essential activities that need to be checked in this phase are-

    - Customer orders are coordinated
    - Deliveries are scheduled
    - Loads are dispatched
    - Customers are invoiced and
    - Payments are received

5. **Returning:** Under many circumstances, on post-delivery, the return of items may be expected as part of a robust business process. No matter how strong the quality control procedures are, there might be cases of unforeseen lapses. In such scenarios of errors, which are most of the times followed by the complaints of customers, a company must recall the product(s), apologize, take back damaged, excess, or unwanted products and return or exchange the right product(s). It is also known as reverse logistics. This stage focuses on customer support and is crucial for maintaining positive relationships with customers.

Apart from these basic components there are also some other components which have been referred to as the drivers of supply chain management on [12]. These are production, information, inventory, location, transportation.. Some of these overlap with the core components of SCM explained above like production means manufacturing, transportation means delivery and distribution of the product. The other elements are explained below-

- **Information:** To be successful, it is crucial for a business to be aware of the current knowledge on the various aspects of its production. When information is shared throughout the organization properly and in time, the supply and demand for a product and its materials are better understood.
- **Inventory:** The list of raw materials and other elements required for a product or service is referred to as an inventory. This needs to be updated regularly to separate the available and required stock when necessary. An inaccurate track of the materials and elements can hamper both the production and supply of products and services.
- **Location:** In any business, location plays an important role in defining the profit of that business. It is important for a company's success that a suitable location is chosen which is well connected and from where it is easy to access the production resources. Warehouse is a part of this driver, which is used as a storage for where all the raw materials or finished products.

Depending on changing business requirements, each of the drivers can be managed to maintain responsiveness or efficiency or both which are two of the most important things that are needed to be considered in a supply chain. Depending on the situations, a mix of responsiveness and efficiency is adjusted continually by companies and supply chains.

Workflow of Supply Chain Management

Now that the components and drivers have been explained, it is time to go into the whole process and flow of the supply chain. In every supply chain, there are some entities that play the roles of different actors in different stages of the whole network. Common actors that are seen in traditional supply chains are-

- Supplier
- Manufacturer
- Distributor
- Wholesaler
- Retailer
- Customer/Consumer

Suppliers take natural resources and make the necessary raw materials for the products. Manufacturer takes those raw materials, produces the final product, and completes packaging. Distributors distribute the products from manufacturer to wholesalers, retailers or even the end-users as it is seen on Figure 3.14. The wholesalers

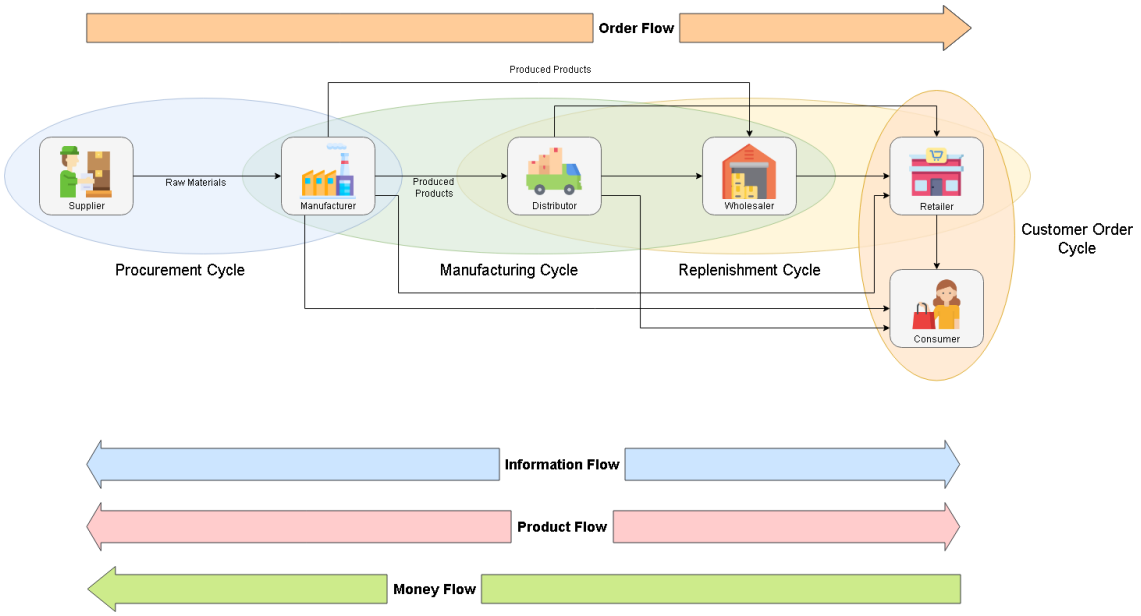


Figure 3.14: Supply Chain Management Process

however take the products in bulk from the distributors or manufacturers, usually store them in a warehouse and only sell the products to retailers. And, finally, the retailers buy the products from wholesalers or take the products from manufacturers or distributors and sell them to the customers. So, the key difference between the wholesalers and the retailers here is that wholesalers are B2B(business-to-business) and retailers are B2C(business-to-customers). Wholesaler and distributor are different from each other as the distributors are usually involved actively in the supply chain to maintain the company's image and often act as a sales representative of the manufacturer and the wholesalers, on the other hand, are only concerned about the orders of the retailers and don't care about the overall sale of the product as long as the retailers are satisfied. Although it is not this simple as the role and position of some actors change in many cases. There are also many supply chains in which the wholesalers act as a distributor or vice-versa. Moreover, there are companies in which the final product gets directly delivered from the manufacturer to the customer. Apple is a major example of that. This is possible when a company manufactures its own products so in a time where e-commerce and online shopping has become a regular thing, when a customer orders a product online, the product gets shipped to them directly from the manufacturing organizations.

As seen on Figure 3.14, during the whole process of supply chain, only order and money or finances are flowing in one direction. Information flow is bidirectional because to complete a flow of order successfully and efficiently, the consistent flow of information between the stages is mandatory. Product flow being bidirectional indicates the execution of the order of the product from supplier to customer and return of the product from customer to manufacturer or supplier on the returning stage when a product is damaged or unwanted. Even though the money or finances flow is shown unidirectional, it is not always that case. There are times when refunds are required so in those cases, the money flow can be bidirectional.

The actors and processes in a supply chain can also be divided in a series of cycles between two or three consequent stages. This is called cycle view which includes-

- Customer Order Cycle (performed between retailers and customers)
- Replenishment Cycle (performed between distributors, wholesalers and retailers)
- Manufacturing Cycle (performed between manufacturers, distributors and wholesalers)
- Procurement Cycle (performed between suppliers and manufacturers)

The processes in a supply chain can also be divided into two categories: push or pull. This is called push/pull view. When orders are executed in response to a customer's order, that is the pull process. Push process is when the orders are initiated in anticipation of the customers' order.

Handfield Nichols (1998) on [3] presented some important information and technology applications for supply chain management which are needed to keep the flow of information and production consistent. These are-

- **EDI:** EDI or Electronic Data Exchange refers to document exchange between computer to computer. This is necessary for communicating information between organizations. It is a very important tool in a supply chain as information is a major factor for a successful supply chain management.
- **Barcoding and Scanning:** It is a very popular technology nowadays seen everywhere especially at supershops. As thousands of products get moved from place to place in a supply chain, this helps identifying and verifying the products. Every barcode holds information about that product so keeping the consistency, authenticity and integrity in the products becomes easier when scanned.
- **Data Warehouse:** Just a warehouse for storing the products is not enough for and effective supply chain. Information also flows with the products like product information, transaction data etc. This information needs to be stored somewhere which is why at every stage, every organization needs a database system for keeping, tracking and managing these data.
- **Internet:** In this age of information and the internet, having an internet is a must for every organization for keeping track of the internal and external information all around the world as well as for communication.
- **Decision Support System:** Not always can humans make decisions on every situation. There might be times when an urgent and efficient decision needs to be made based on the past calculations and predictable outcomes. During these situations, an automated and easy to implement decision system is needed to keep the supply chain intact.

3.3.4 Supply Chain Management Metrics

Organizations can use metrics or key performance indicators (KPIs) to help them focus on the most critical operations and enhance existing processes. Regulatory

compliance, safety, and contractual duties are all supported by critical metrics. Other metrics track and enhance efficiency, customer service, and profit margins. To have an optimized supply chain, 5 common metrics should be tracked are, which are-

1. **Perfect Order Index**

   Perfect order index measures the percentage of error-free orders in the entire supply chain operation. It multiplies the perfect orders from each stage to indicate an overall performance.

2. **Cash-to-Cash Cycle Time**

   The cash-to-cash (C2C) cycle or in general sense, cash conversion, tracks how long it takes a company to pay the suppliers for the raw materials and get paid by the customers for the final product. It is made up of 3 measurements of a supply chain: inventory days, payables days, and receivables days.

3. **Order Cycle Time**

   Order cycle time, also referred as supply chain cycle time, measures the time from order receipt to product delivery. A shorter cycle indicates that the process is more adaptable, agile, and responsive to the environmental changes.

4. **Fill Rate**

   The fill rate, which is also known as the demand satisfaction rate, is the percentage of customer orders delivered as ordered on the first shipment without backorders or lost sales. It depicts the sales that can be serviced better if inventory performance is improved.

5. **Inventory Turnover**

   Inventory turnover calculates the number of times the entire inventory is sold in a certain time period. It's a conclusive metric that provides a full and accurate picture of how efficient the supply chain is. A low inventory turnover generally implies that a company has excess inventory because of weak sales.

3.3.5 Key Features of Effective and Thinking Supply Chain Management

As the technology is evolving and the competition in business is rising everyday, depending heavily on a human managed supply chain will not be effective in near future or even not effective today. So to be at the same pace with the world, the industries must have a thinking supply chain which does not necessarily replace but helps and speeds up the whole management of supply chains.

IDC's Simon Ellis in one of his paper [11] defined five Cs that IBM on [18] identified as the key features of an effective and thinking supply chain-

1. **Connected:** Access unknown and unlearned data like unstructured data from social media, structured data from the traditional B2B and ERP integration tools.

2. **Collaborative:** Using cloud-based commerce networks for improving collaboration and engagement with suppliers.

3. **Cyberaware:** Having the ability to secure the systems and databases from hacks and other cyberintrusions.

4. **Cognitively Enabled:** AI platform controlling the supply chain by being the decision maker and leading the next best actions in the chain in an automated, self-learning and timely way.

5. **Comprehensive:** Being fast, comprehensive, having no latency and the analytical capabilities scaling with data in real time.

Chapter 4

Methodology

The primary purpose of this supershop supply-chain model is to ensure the authenticity of products from the production to consumer level. In order to carry out this operation, we are going to use blockchain to keep the production information, and each information in the blockchain is going to be followed by a hash value which will ensure the immutability of the information. The Fabric uses SHA-256 hashing algorithm to generate the hash values. Furthermore, the operational support team of the supply-chain management uses a private database to keep the integrity of the manufactured and delivered products. When the product is authenticated by the operational support team and ready for transfer to the distributors its information is added to the blockchain which is not modifiable and if someone tries to do so, the operational support team will be notified. Finally, the production phase is completed by getting the product blocks of food and toys digitally signed and storing the information to the ledger of operational support database. The figure 4.1 shows the flow of the above mentioned process.

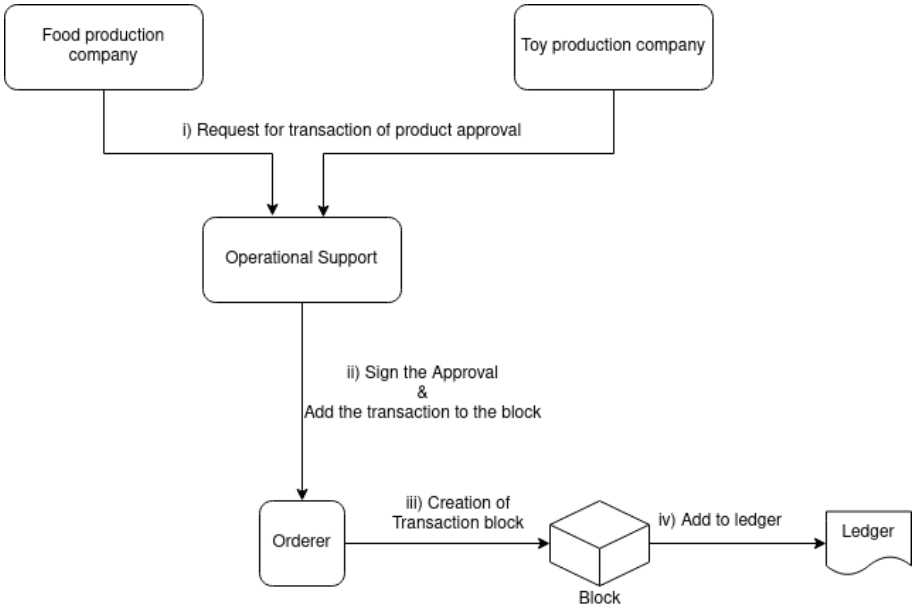


Figure 4.1: Production stage workflow

When the production and approval stage is complete, a smart contract is deployed by the Operational support to safely pass the product through different stages. When the product is being transferred from the food production or toy production company to the food or toy distributors, a smart contract will be deployed between them through the chain code to ensure the integrity of the product by using digital signatures between the production company and the delivery company and update the location. Again, the delivery company completes the transaction with the distribution companies and make sure the location and ownership of the product is transferred to distributor companies. When the products reach the distributors of food and toys, they transfer it to the delivery companies. The smart contract will take place between the distributors and delivery companies. Again, the batch of the products will be digitally signed by both parties and the location of the product will be set to delivery companies but the ownership will remain to the distributors. Finally, the delivery company will deliver the product by using another smart contract between them and the supershop and the location and ownership will be updated again. Using the smart contract, the owner of the products will be able to pinpoint the previous location and owners of the product and authenticate the whole transferal process. Along with the supershop, the customer can also verify if they are buying an authentic product by using product codes on the packaging to the authenticity of the products they are buying. (Figure 4.2)

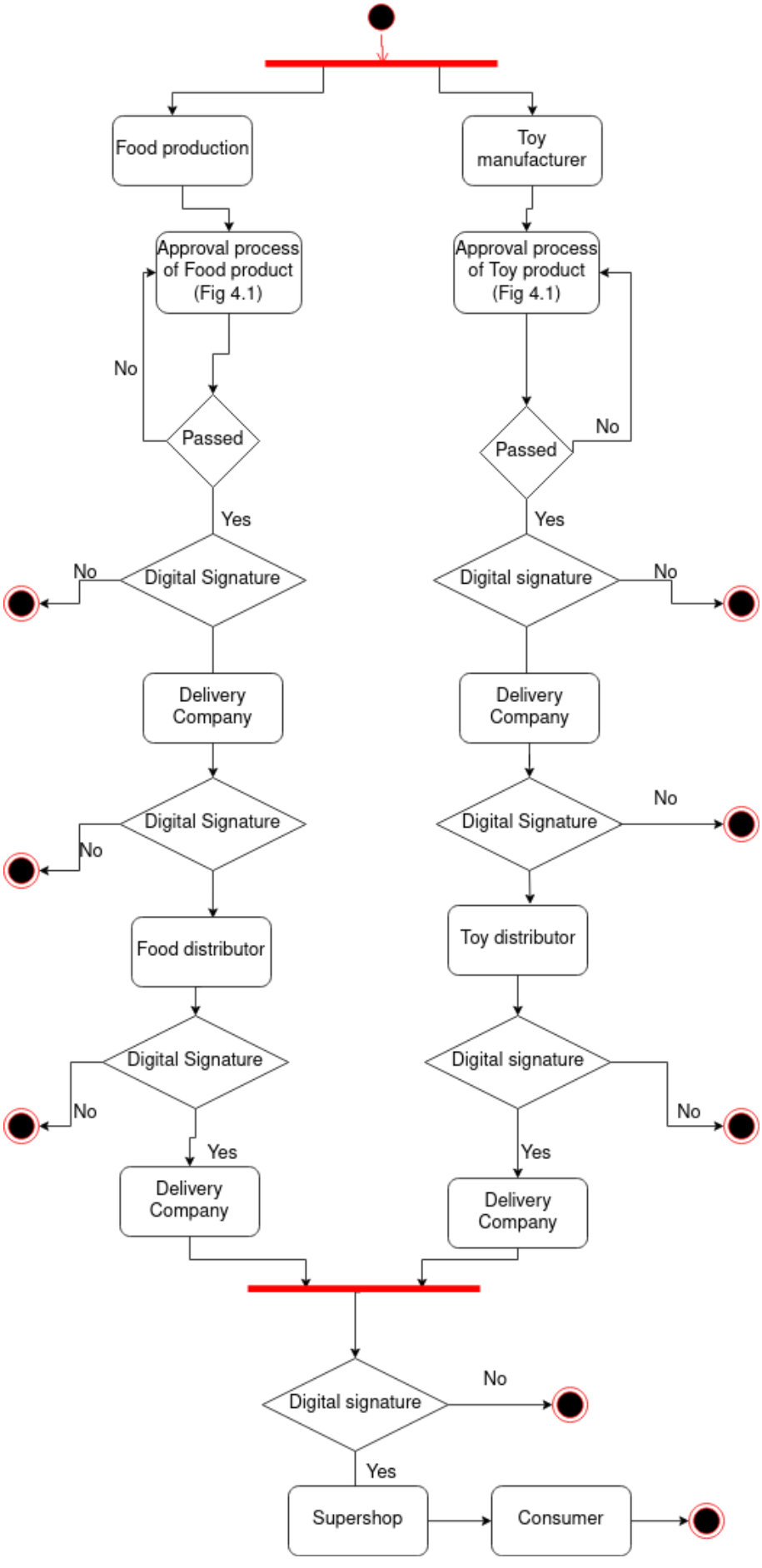


Figure 4.2: Flow of product supply

4.1 Hyperledger Fabric Network Model Of Supershop Supply chain

To build our supershop system, at first, we are going to create a network that will ensure quality control and trace the products from different kinds of suppliers for the supershop system. Hence, several organizations will together form a consortium to build our network. The consortium comes to an agreement upon a set of policies that will authorize the permissions for the participating organizations. There will also be smart contracts that will be used to create transactions and be responsible for sending the transaction information to the communicating nodes connected through a particular channel. The network will have the following organizations -

- Food Producer
- Toy Manufacturer
- Operational Support
- Food Distributor
- Toy Distributor
- Supershop
- Delivery Company

The mentioned organizations will store information in the blockchain. It will keep track of the transfer products from other organizations to the Supershop, the transactions made by each organization, and the location of the product.

4.1.1 Components of the network model

Applications

For the proposed network model, we are going to have seven applications through which the participating members are going to interact with the network. In our network model, the applications of Food producer, Toy manufacturer, Operational support, Food distributor, Toy distributor, Supershop and delivery company are identified as A1, A2, A3, A4, A5, A6, A7 respectively. Each application consists of peers and a ledger. There are four types of peers that are required in a functioning network for the proposed model. These are -

1. **Orderer peer:** The ordering service or peer is the initiator of this network. The primary purpose of this peer is to receive the endorsed transactions from the endorsing peers and turn them into blocks in accordance with the configuration file of the network. After creating the blocks, the Orderer peer sends the blocks to the other peers in the network in order for them to check the integrity of the block and add it to the ledger. While committing peers and endorsing peers only keep record of valid transactions, the Orderer peer keeps track of all transactions regardless of their validity. Therefore, due to its administrative nature, the operational support team controls it in our network.

2. **Anchor peers:** Anchor peers of the organizations shall be used for communication between the peers using the gossip protocol. Only the anchor peer connected through the same channel can utilize this feature.

3. **Committing peers:** Committing peers are ones responsible for saving the transactions once they receive it from the order service. The transaction is saved in the ledger of the peer and shared with the other peers using the gossip protocol.

4. **Endorsing peers:** Endorsing peers will primarily be used for endorsing the transacting which will require endorsing peers of the participating organizations to come to an agreement to the proposal of chaincode invocation. Each endorsing peers in the network will separately implement a chaincode utilizing the transaction proposal made by a particular endorsing peer and generate a transaction proposal response. For example - in the production phase the producers make a transaction proposal to the Operational support their peers must make sure the production records and the production assortment is compatible.

Channels and Ledgers

The network will have three channels named channel 1, channel 2, and channel 3. Channel 1 shall be used only for privatized transactions between Food producer and Operational support as transparency of production data might hamper the production companies. Similarly, channel 2 shall be used for transactions between toy manufacturers and the operational support. Lastly, previously mentioned and the remaining organizations will make transactions using channel 3 which is completely transparent. As the transactions of channel 1 and 2 are private they will maintain separate ledgers named L1 and L2. However, for the public channel named channel 3, another ledger named L3 shall be maintained by it.
L1, L2 will contain the Manufacturing or production information of the products manufactured by the Food producer and Toy manufacturer respectively.
L3 will contain product information and location of Food and Toys in the supply chain.

Membership Service Provider

As Hyperledger fabric is a permissioned network unlike regular blockchain, the participating members need to prove their eligibility in order to make transactions on the network. This is where the certificate authority will play the key role. The certificate authority will provide unique identities to the organizations by creating a public and a private key so that they can prove their identity. When a peer uses a private key for endorsing or digitally signing a transaction the MSP uses the peers' public key to verify the signature attached to the transaction. The Operational support will play the administrative role in proving the membership to the network. Combining all the components discussed above the network model is shown in figure 4.3.

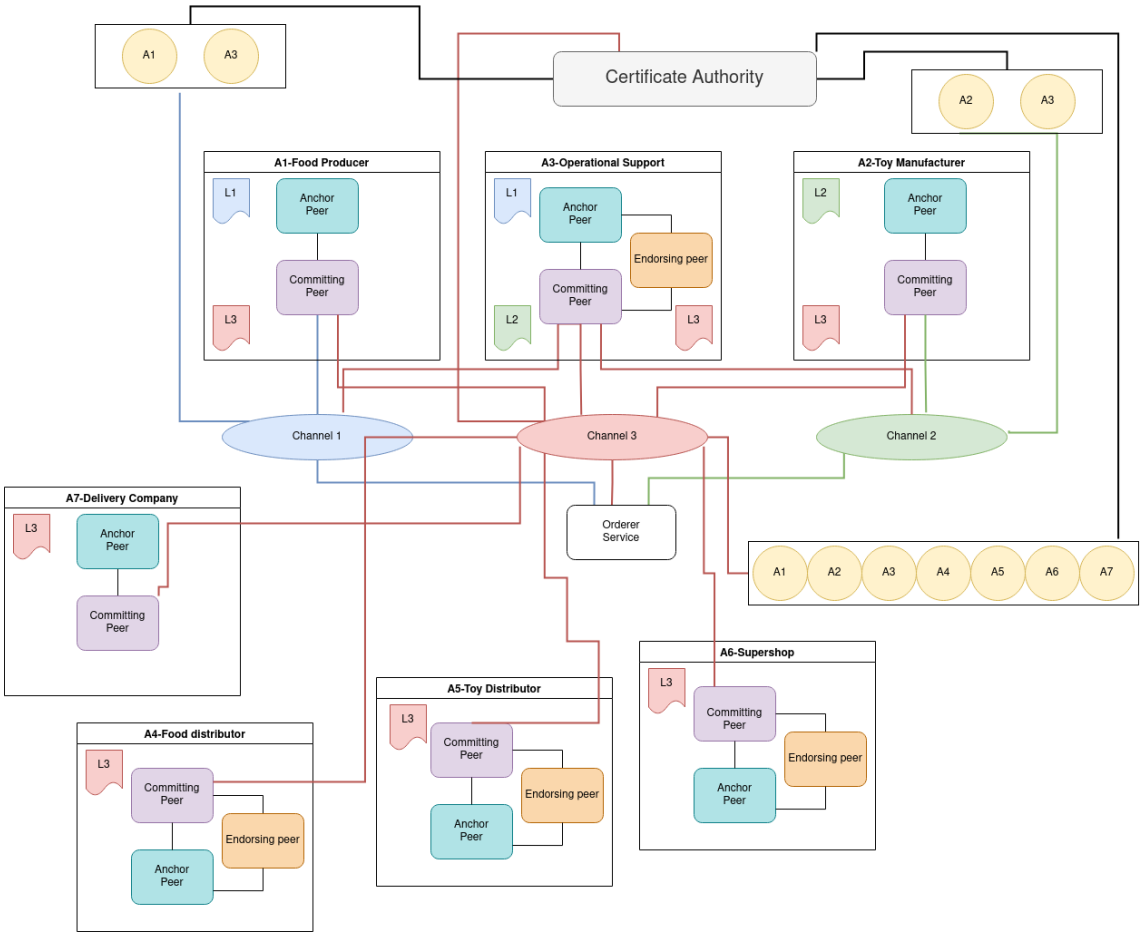


Figure 4.3: Network Model of the Supershop Supply-chain

4.2 Working Procedures

The functioning procedures of the network can be divided into two stages and they are -

1. The Production approval procedure
2. The Supply chain procedure

4.2.1 The Production approval procedure

The transactions in the production part of the network occur as a three stage procedure.

Proposal Stage

The particular phase involves communication between the peers and some other endorsing peers. First, the food producer or Toy manufacturer makes a batch of

products and they generate a record of production and hash that record using the SHA256 algorithm. After hashing the record, the application of the Toy Manufacturer and Food producer creates a transaction proposal and sends it to the operational support for endorsing the transaction. Then the endorsing peer, which in this case is the operational support, executes the chaincode using the received transaction proposal to create a response to the transaction proposal. Instead of updating the ledger in this step, the endorsing peer simply signs the transaction and sends it back to the application of the Food distributor for further processing and completes the first phase of the procedure. Here we can see from the figure 4.4 that A1 of the Food producer is creating transactions T1, T2, T3, T4 and sending transaction proposals using channel 1 to the endorsing peer of A3 or Operational support. Again, another application A2 is creating transactions T5, T6, T7 and sending the proposal to the endorsing peer of A3 through channel 2. The endorsement peer of A3 is sending the approved transaction T2, T3 along with endorsement back to A1 and endorsed T5 back to A2.

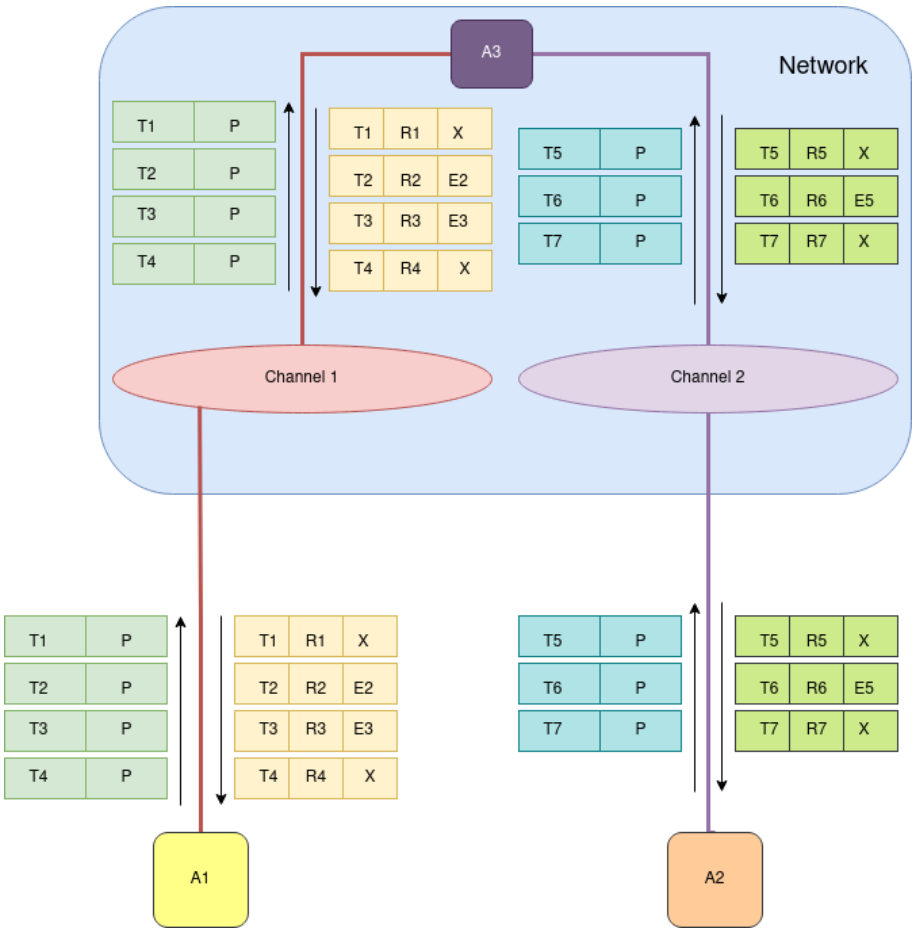


Figure 4.4: Proposal and Transaction endorsement

Creating Blocks

After the completion of proposal stage the client applications A1, A2 have received some endorsed transaction proposal response from the endorsing peer of A3. Now, in this stage of the production approval procedure A1, A2 sends the endorsed transaction proposal responses to the Orderer node. The orderer node will create blocks named "L1 BLOCK-1" and "L2 BLOCK-2" using the responses and ultimately distribute those to every peer on that channel for validation. The number of transactions per block and the max elapsed duration of blocks are set using the "BatchSize" and "BatchTimeout" parameters. Then the block "L1 BLOCK-1" and "L2 BLOCK-2" are saved in the ledger of the orderer peer and later distributed to L1 and L2 ledgers based on the channels they are part of. (Figure 4.5)

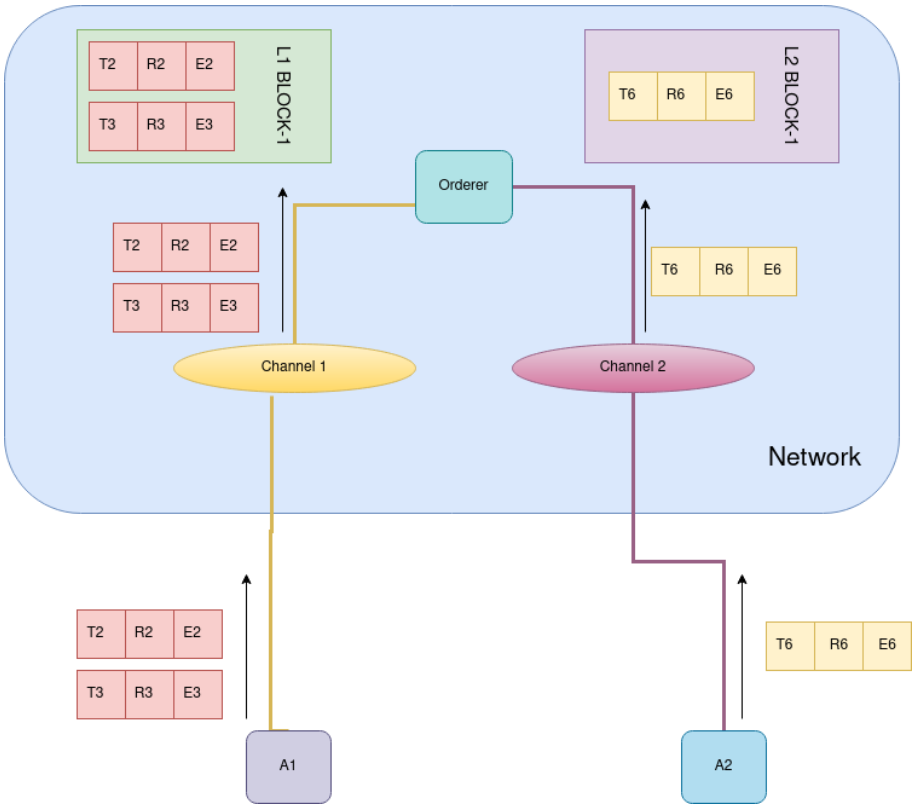


Figure 4.5: Orderer peer creating blocks

Validation and Committing the Blocks

During the final phase of this transaction workflow, distribution and validation of the blocks occur from the order peer to the related peers and the ledgers get updated. Besides packaging blocks, the Orderer peer distributes the blocks to peers. Here in figure 4.6, the Orderer peer distributes the blocks "L1 BLOCK-1" and "L2 BLOCK-2" created in the previous phase to the peers. "L1 BLOCK-1" is updated in ledger L1 of peer A1 and A3, while "L2 BLOCK-2" is updated in the ledger L2 of peer A2 and A3. Finally, after the the block creation is complete, the change becomes permanent and product quality is ensured.

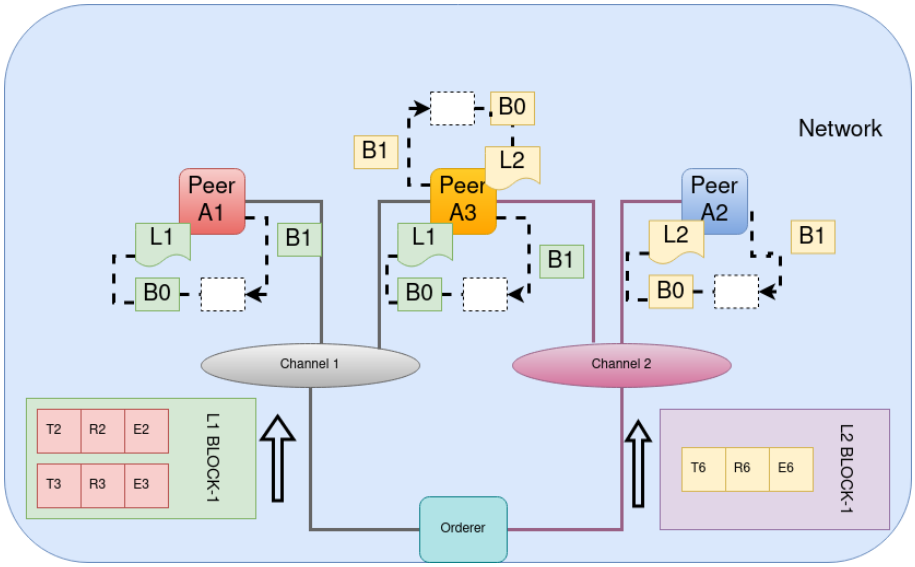


Figure 4.6: Validation and committing the blocks

4.2.2 Supply Chain Workflow

Interaction in the supply chain part of the network is done utilizing channel 2. As the primary goal of this research is to ensure product authenticity through transparency, all the concerned peers are connected in channel 3, or in other words, channel 3 is made public. In the supply chain phase, the participating members will sign every transaction and get endorsed by our central authority or Operational support. The organizations will be the one storing in the blockchain. They will store the states and location through which the product is passing and update ownership. A preview of these steps is shown in figure 4.7.

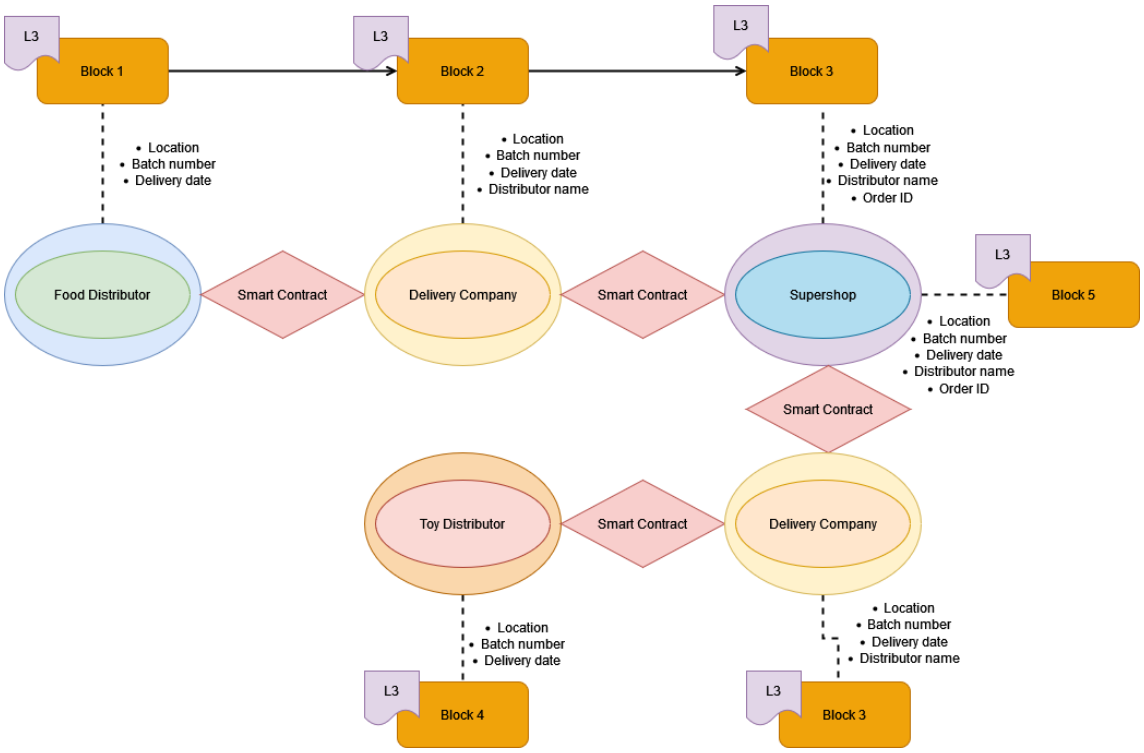


Figure 4.7: Supply chain phase workflow

Chapter 5

Baseline Implementations and Results

Previously we have shown the basic model and architecture of our model which gives a concrete idea about how blockchain and our model works. We have seen how different organizations share their infrastructure to integrate with blockchain networks using Hyperledger Fabric. In this section, we talk about the actual implementation of our network which includes the creation of smart contracts as well.

5.1 Prerequisite and hardware requirements

For the real-life implementation of this model, there are some compulsory minimum requirements. Without these requirements, it is not possible to run this proposed model. The required software and hardware specifications are listed below:

Software Requirements:

- Ubuntu Linux 14.04 / 16.04 LTS (both 64-bit) or higher, or Mac OS 10.12 or higher
- Docker Engine: Version 17.03 or higher
- Docker-Compose: Version 1.8 or higher
- Git: 2.9.x or higher
- Python: 2.7.x
- Node.js 8.9 or higher / Java Oracle JDK 8.0
- A code editor.e.g VSCode

Hardware Requirements:

- Memory: A memory of 4GB at least
- HDD: A storage of 20GB HDD or more
- Processor: Any CPU with at least 1.0Ghz clock speed is recommended

5.2 Implementation of Code

After we have installed the necessary dependencies of Hyperledger Fabric 1.4, we are going to define our network structure discussed in the previous section of this research. The network has been using Fabric 1.4 using all the necessary actors along with a rest API to interact with the network.
Now we are going to discuss the fundamental components of the network.

5.2.1 Defining the organizations

First, we are going to define the organizations in our network. There are a total of 7 organizations in this network and we are going to define them in our first step.

```
OrdererOrgs:
  - Name: Orderer
    Domain: example.com
    EnableNodeOUs: true
    Specs:
      - Hostname: orderer

PeerOrgs:
  - Name: FoodProducer
    Domain: foodproducer.example.com
    EnableNodeOUs: true
    Template:
      Count: 1
    Users:
      Count: 1
  - Name: ToyManufacturer
    Domain: toymanufacturer.example.com
    EnableNodeOUs: true
    Template:
      Count: 1
    Users:
      Count: 1
  - Name: OpSupport
    Domain: opsupport.example.com
    EnableNodeOUs: true
    Template:
      Count: 1
    Users:
      Count: 1
```

Figure 5.1: Defining Network Organizations (Production phase)

In figure 5.1 we can see an instance of the peer organizations that are going to be part of the production phase of the network. There are more organizations in the network who are part of the supply-chain phase shown in figure 5.2.

```
30    - Name: Deliverer
31      Domain: deliverer.example.com
32      EnableNodeOUs: true
33      Template:
34        Count: 1
35      Users:
36        Count: 1
37    - Name: FoodDistributor
38      Domain: fooddistributor.example.com
39      EnableNodeOUs: true
40      Template:
41        Count: 1
42      Users:
43        Count: 1
44    - Name: ToyDistributor
45      Domain: toydistributor.example.com
46      EnableNodeOUs: true
47      Template:
48        Count: 1
49      Users:
50        Count: 1
51    - Name: Supershop
52      Domain: supershop.example.com
53      EnableNodeOUs: true
54      Template:
55        Count: 1
56      Users:
57        Count: 1
```

Figure 5.2: Defining Network Organizations (Supply-Chain phase)

Next, we have included an entry for every organization in the network with their own MSP indicator so that we can detect it as a part of our network and grant the ability to belong to a channel and make transactions. In addition to that, the consortiums are defined so that the organizations can be part of the channel.
In Figure 5.3 a sample of configuring MSP for the Super shop organization is shown. We have set policies which include readers, writers, and admins. An ID for the organization in the network is also defined here. "Signature" value is set for users who have to sign for the policy to be satisfied.

```
    - &SuperShop
      Name: SuperShopMSP
      ID: SuperShopMSP
      MSPDir: crypto-config/peerOrganizations/SuperShop.example.com/msp
      Policies:
          Readers:
              Type: Signature
              Rule: "OR('SuperShop.admin', 'SuperShop.peer', 'SuperShop.client')"
          Writers:
              Type: Signature
              Rule: "OR('SuperShop.admin', 'SuperShop.client')"
          Admins:
              Type: Signature
              Rule: "OR('SuperShop.admin')"
```

Figure 5.3: Including MSP for an Organization

Again, in figure 5.4, the consortium is defined in order for the organizations to communicate through a particular channel. For example, here the Food producer and the Operational support are building a consortium to make transactions using Channel 1.

```
Profiles:

    SupplyOrdererGenesis:
        <<: *ChannelDefaults
        Orderer:
            <<: *OrdererDefaults
            Organizations:
                - *OrdererOrg
            Capabilities:
                <<: *OrdererCapabilities
        Consortiums:
            Consortium1:
                Organizations:
                    - *FoodProducer
                    - *OpSupport
    Channel1:
        Consortium: Consortium1
        <<: *ChannelDefaults
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *FoodProducer
                - *OpSupport
            Capabilities:
                <<: *ApplicationCapabilities
```

Figure 5.4: Defining consortiums for channel 1

5.2.2 Chaincode

Using the chain code, the required methods for interacting with the blockchain network shall be defined. The methods are -

- **addAsset:** This method will insert the object sent by parameter of our ledgers. we are able to input 3 types of assets, Manufacturer name, product name and batch number. (Figure 5.5)

```
class SypplyChain extends Contract {


  async addAsset(ctx, asset) {
    console.info('============= START : Add asset ===========');
    await ctx.stub.putState(JSON.parse(asset).id.toString(), Buffer.from(asset));
    console.info('============= END : Add asset ===========');
    return ctx.stub.getTxID()
  }
```

Figure 5.5: addAsset

- **queryAsset:** This method allows the functionality to check the status of the asset in our blockchain network. The method does not access the ledger and instead it only shows the current status of the added asset by reading world state.

```
async queryAsset(ctx, assetId) {
  console.info('============= START : Query asset ===========');
  const assetAsBytes = await ctx.stub.getState(assetId);
  if (!assetAsBytes || assetAsBytes.length === 0) {
    throw new Error(`${assetId} does not exist`);
  }
  console.log(assetAsBytes.toString());
  console.info('============= END : Query asset ===========');
  return assetAsBytes.toString();
}
```

Figure 5.6: queryAsset

- **setPosition:** This method of our chaincode will ensure the position or the exact coordinates of the products by using the id and the product coordinates. By fetching the asset from the ledger it will update the new location after the product has been passed down in our supply chain.

```
async setPosition(ctx, id, latitude, longitude) {
  console.info('============= START : Set position ===========');
  const keyAsBytes = await ctx.stub.getState(id);
  if (!keyAsBytes || keyAsBytes.length === 0) {
    throw new Error(`${id} does not exist`);
  }
  let key = JSON.parse(keyAsBytes.toString());
  key.latitude = latitude;
  key.longitude = longitude;
  await ctx.stub.putState(id, Buffer.from(JSON.stringify(key)));
  console.info('============= END : Set position ===========');
  return ctx.stub.getTxID();
}
```

Figure 5.7: setPosition

- **getHistory:** The method will return the history of the assets, including the changes that were made in order to pass the product in the supply chain.

```
async getHistory(ctx, id) {
  console.info('============= START : Query History ==========');
  let iterator = await ctx.stub.getHistoryForKey(id);
  let result = [];
  let res = await iterator.next();
  while (!res.done) {
    if (res.value) {
      console.info(`found state update with value: ${res.value.value.toString('utf8')}`);
      const obj = JSON.parse(res.value.value.toString('utf8'));
      result.push(obj);
    }
    res = await iterator.next();
  }
  await iterator.close();
  console.info('============= END : Query History ==========');
  return result;
}
```

Figure 5.8: getHistory

5.2.3 Creating Network Users

For the users to be able to interact with our network, we have assigned the operational support as admin and they will give other users access to the channels and ledgers. For enrolling the admin, the following steps were used in figure 5.9-

- Calling js using the organization name of operational support using “node adminEnrollment.js opSupport”
- Retrieving the connection of opSupport
- Creating a certificate authority client through the package we have used
- Creating a wallet to store the admin certificate of opSupport
- Identity creation and storing it in the wallet using a username and password

```
'use strict';

const FabricCAServices = require('fabric-ca-client');
const { FileSystemWallet, X509WalletMixin } = require('fabric-network');
const fs = require('fs');
const path = require('path');

async function main() {
    try {
        const args = process.argv.slice(2);
        const org = args[0];
        const ccpPath = path.resolve(__dirname, '..', 'connections', `connection-${org}.json`);
        const ccpJSON = fs.readFileSync(ccpPath, 'utf8');
        const ccp = JSON.parse(ccpJSON);

        // Create a new CA client for interacting with the CA.
        const caInfo = ccp.certificateAuthorities[`ca.${org}.example.com`];
        const caTLSCACerts = caInfo.tlsCACerts.pem;
        const ca = new FabricCAServices(caInfo.url, { trustedRoots: caTLSCACerts, verify: false }, caInfo.caName);

        // Create a new file system based wallet for managing identities.
        const walletPath = path.join(process.cwd(), `wallet/wallet-${org}`);
        const wallet = new FileSystemWallet(walletPath);
        console.log(`Wallet path: ${walletPath}`);

        // Check to see if we've already enrolled the admin user.
        const adminExists = await wallet.exists('admin');
        if (adminExists) {
            console.log('An identity for the admin user "admin" already exists in the wallet');
            return;
        }

        // Enroll the admin user, and import the new identity into the wallet.
        const upper = org.replace(/^\w/, c => c.toUpperCase());
        const enrollment = await ca.enroll({ enrollmentID: 'admin', enrollmentSecret: 'adminpw' });
        const identity = X509WalletMixin.createIdentity(`${upper}MSP`, enrollment.certificate, enrollment.key.toBytes());
        await wallet.import('admin', identity);
        console.log('Successfully enrolled admin user "admin" and imported it into the wallet');

    } catch (error) {
        console.error(`Failed to enroll admin user "admin": ${error}`);
        process.exit(1);
    }
}

main();
```

Figure 5.9: Registering the admin

After registering the admin we used the same identity to generate user using the following steps in figure 5.10-

- Calling js using the organization name of operational support using “node adminEnrollment.js opSupport”
- Retrieving the connection of opSupport

- Creating a certificate authority client through the package we have used
- Identity creation and storing it in the wallet using a username and password

```
'use strict';

const { FileSystemWallet, Gateway, X509WalletMixin } = require('fabric-network');
const path = require('path');

async function main() {
    try {

        const args = process.argv.slice(2);
        const org = args[0];
        const ccpPath = path.resolve(__dirname, '..', 'connections', `connection-${org}.json`);
        // Create a new file system based wallet for managing identities.
        const walletPath = path.join(process.cwd(), `wallet/wallet-${org}`);
        const wallet = new FileSystemWallet(walletPath);
        console.log(`Wallet path: ${walletPath}`);

        // Check to see if we've already enrolled the user.
        const userExists = await wallet.exists('user1');
        if (userExists) {
            console.log('An identity for the user "user1" already exists in the wallet');
            return;
        }

        // Check to see if we've already enrolled the admin user.
        const adminExists = await wallet.exists('admin');
        if (!adminExists) {
            console.log('An identity for the admin user "admin" does not exist in the wallet');
            console.log('Run the enrollAdmin.js application before retrying');
            return;
        }
        // Create a new gateway for connecting to our peer node.
        const gateway = new Gateway();
        await gateway.connect(ccpPath, { wallet, identity: 'admin', discovery: { enabled: true, asLocalhost: true } });

        // Get the CA client object from the gateway for interacting with the CA.
        const ca = gateway.getClient().getCertificateAuthority();
        const adminIdentity = gateway.getCurrentIdentity();

        const upper = org.replace(/^\w/, c => c.toUpperCase());
        // Register the user, enroll the user, and import the new identity into the wallet.
        const secret = await ca.register({enrollmentID: 'user1', role: 'client' }, adminIdentity);
        const enrollment = await ca.enroll({ enrollmentID: 'user1', enrollmentSecret: secret });
        const userIdentity = X509WalletMixin.createIdentity(`${upper}MSP`, enrollment.certificate, enrollment.key.toBytes());
        await wallet.import('user1', userIdentity);
        console.log('Successfully registered and enrolled admin user "user1" and imported it into the wallet');

    } catch (error) {
        console.error(`Failed to register user "user1": ${error}`);
        process.exit(1);
    }
}

main();
```

Figure 5.10: Registering the Users

5.2.4 Creation of API

In this part, a method is created which receives the previously created connection and the wallet as a parameter. It will establish a connection to the network and return our chaincode to those we want to send a transaction to. Also, we have created routes that are accessible to the API and call the chaincode methods that we are using. Furthermore, get () methods in our API will return the data negotiated for the previous identifier. They will not generate a transaction in our network, instead they will only check the status of the identifier we are sending. Whereas, the post method will save the transactions in the ledgers of our network and return the generated transaction id . The post method will redirect transactions to the network and be endorsed before storing in the ledger (Figure 5.12). Finally, if we initiate the server the API will listen and enable us to interact with the network.

```
app.get('/api/getFood/:id', async function (req, res) {
  try {
    const contract = await fabricNetwork.connectNetwork('connection-SuperShop.json', 'wallet/wallet-SuperShop');
    const result = await contract.evaluateTransaction('queryAsset', req.params.id.toString());
    let response = JSON.parse(result.toString());
    res.json({result:response});
  } catch (error) {
    console.error(`Failed to evaluate transaction: ${error}`);
    res.status(500).json({
      error: error
    });
  }
})
```

Figure 5.11: "get ()" Method of API

```
app.post('/api/addFood', async function (req, res) {

  try {
    const contract = await fabricNetwork.connectNetwork('connection-foodProducer.json', 'wallet/wallet-foodProducer');
    let Food = {
      id: req.body.id,
      name: req.body.name,
      latitude: req.body.latitude,
      longitude: req.body.longitude,
      length: req.body.length,
      weight: req.body.weight
    }
    let tx = await contract.submitTransaction('addAsset', JSON.stringify(tuna));
    res.json({
      status: 'OK - Transaction has been submitted',
      txid: tx.toString()
    });
  } catch (error) {
    console.error(`Failed to evaluate transaction: ${error}`);
    res.status(500).json({
      error: error
    });
  }

});
```

Figure 5.12: "post ()" Method of API

5.3 Results

Transaction

Here in Figure 5.13 we have created a sample call to our network to add new food item to our network using sample values-

```
solaiman@hossain-18101202:~$ curl --request POST \
> --url http://localhost:3000/api/addFood \
> --header 'content-type: application/json' \
> --data '{\
> "Name":Coca Cola,\
> "id":10001,\
> "latitude":"43.3623",\
> "longitude":"8.4115",\
> "length":34,\
> "weight":50\
>  }'
```

Figure 5.13: Sample call

After the product addition is successful we get the response shown in figure 5.14 in the command shell. The result returns a transaction confirmation message and a transaction id.

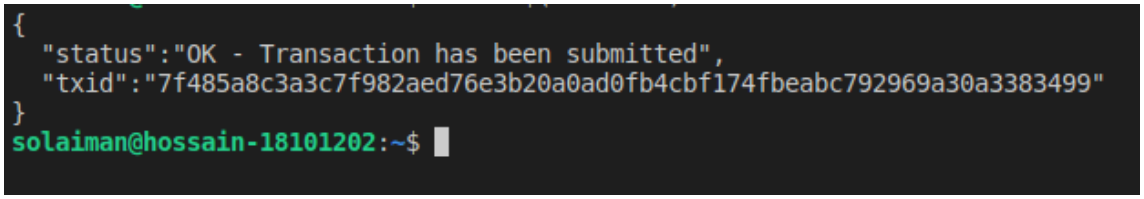


Figure 5.14: Transaction confirmation

Now when we make another sample call using the product id we will get the Food product information that we added in the blockchain using a sample call. (Figure 5.15)

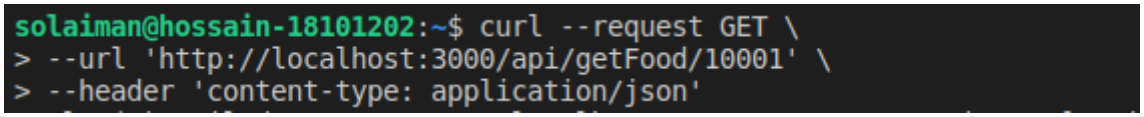


Figure 5.15: Sample call for retrieving information from ledger

After the sample call is made a successful response (figure 5.16) is received from the network which returns several parameters stored in the blockchain.

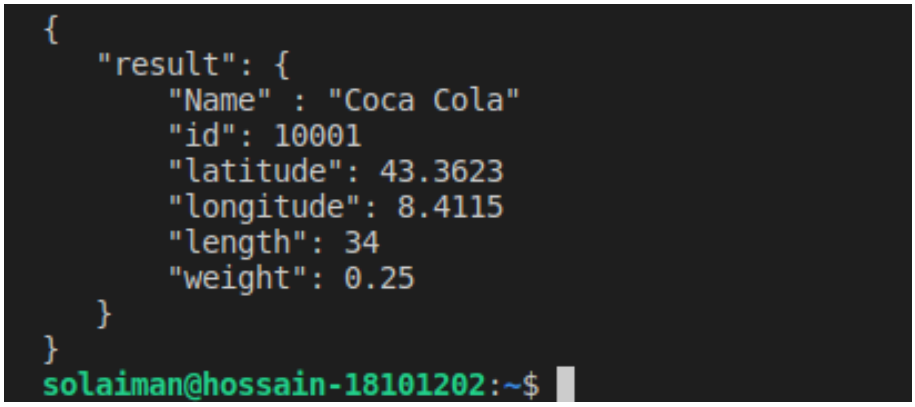


Figure 5.16: Getting information from ledger

Similarly, we can get the product history through a sample call to our network. In the product history we can see the changing location and ownership history of the products. (Figure 5.17)

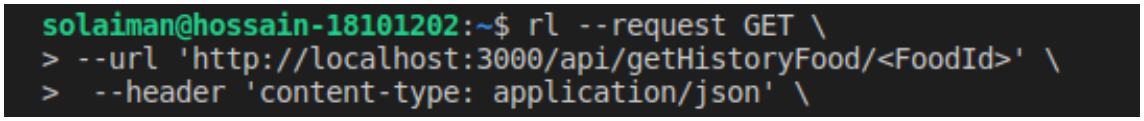


Figure 5.17: Sample Call to get product history

The call returns with the ownership, name, id, latitude, longitude, type, id values history stored in the blockchain along with the approval in production stage. (Figure 5.18)

Figure 5.18: Product History

Thus, getting the product history ensures the transparency that we have maintained in our network in order to maintain the authenticity. The customers can get the product history using a validated product code to ensure that the product they are using is not a counterfeit.

Chapter 6

Performance Analysis Metrics

Due to the fact that we have only made backend integration of our network we were not able to do performance analysis. Thus, we are going to discuss a few performance metrics of the hyperledger fabric network that we plan to use for the performance testing of our network in our future works. [5]

Read Latency:

Read latency is the delay between the submission of the read request and the response.
Therefore, Read latency = Response time - Submission time

Read Throughput:

Read throughput is the measurement of completed read operations in a particular period of time which is indicated by RPS or reads per second. Although, it is not a primary measurement of hyperledger network performance.Systems are deployed alongside blockchain to speed up the read process and queries.
Read Throughput = Read operation / Total time

Transaction Latency:

Transaction latency is the time taken for a transaction to be in effect throughout a particular network. This measurement is done using the time of result submission to the time of the availability of result in the network. It includes propagation delay and other times consumed for settlement of the consensus mechanism. The delay is measured across all nodes in the network.
Transaction latency = (Time of confirmationNetwork threshold -Submission time)

Transaction Throughput:

It is the rate of transactions which are completed by the blockchain in a particular time. It is also indicated by TPS or transaction per second throughout the network.
Transaction throughput = (Total transactions - Valid transactions)/Total time

A benchmark done in [5] focuses on evaluating a transaction through the gateway of a network. When retrieving an asset using a randomized UUID from the world state database of CouchDB, they have noted the throughput and latencies concerning the

maintenance of a constant transaction backlog for the four clients in their network. Here is the benchmark results achieved from the test -

Asset Size (Bytes)	Max Latency (s)	Avg Latency (s)	Throughput (TPS)
100	1.10	0.06	567.4
1000	1.06	0.07	558.9
2000	0.24	0.07	531.4
4000	0.25	0.08	478.0
8000	0.26	0.09	395.4
16000	0.29	0.12	306.1
32000	0.36	0.17	208.3
64000	0.75	0.35	107.0

Figure 6.1: Benchmark results of asset retrieval from database [5]

The utilization of resources is observed using a fixed transaction rate of 350 transactions per second and asset size of 8 Kilobytes.

Chapter 7

Conclusion

To conclude, Hyperledger fabric can play a vital role in the supershop supply chain for product authentication. Among all the other blockchain technology, Hyperledger fabric is the best in this scenario because of having a private network system while giving high efficiency and performance without losing any security. We have properly pointed out how the technology works in supershop and helps to authenticate the products within our network that we created initially. It has also been discussed how the ledgers are going to communicate among themselves within the network and will operate transactions. The results of using the technology are also demonstrated in CLI. Although we have used Hyperledger fabric in a sample super shop network which consists of relatively few channels and organizations, this model of ours can be further improved by adding additional ones according to needs. In the near future, there will be many encryption algorithms which can make the blockchain network much more secure. However, there are still some problems with the PoW and PoS algorithms which can be manipulated which is why algorithms like Ouroboros and Casper are being developed. This will offer better security to the blockchain and make the network much more authentic. Soon we can use the latest encryption algorithms to improve our model further.

Bibliography

[1] G. Greenspan, "MultiChain private blockchain - white paper," pp. 1–17, 2013.

[2] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Applied Innovation*, vol. 2, 2016.

[3] R. B. Handfield and E. L. Nichols, *Introduction to supply chain management*, 2nd ed. Upper Saddle River, NJ: Pearson, Jul. 2017.

[4] F. P. Sylim, A. Liu, and P. Marcelo, "Blockchain technology for detecting falsified and substandard drugs in distribution: Pharmaceutical supply chain intervention"," *JMIR Res Protoc*, vol. 7, no. 9, pp. 1–12, 2018.

[5] N. Lincoln, "Hyperledger fabric 1.4. 0 performance information report," *Retrieved*, vol. 14, 2019.

[6] J. C. Molina, D. T. Delgado, and G. Tarazona, "Using blockchain for traceability in the drug supply chain"," *Communications in Computer and Information Science*, vol. 1027, pp. 536–548, 2019.

[7] A. K. Shrestha and J. Vassileva, "User data sharing frameworks : A Blockchain-Based incentive solution," in *The 10th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEEE IEMCON 2019)*, 2019.

[8] A. K. Shrestha, R. Deters, and J. Vassileva, "User-controlled privacy-preserving user profile data sharing based on blockchain," Sep. 2019. arXiv: 1909.05028 [cs.CR].

[9] A. K. Shrestha and J. Vassileva, "User acceptance of usable blockchain-based research data sharing system: An extended TAM based study," Dec. 2019. arXiv: 2001.00079 [cs.CY].

[10] C. Yin, B. Zhou, Z. Yin, and J. Wang, "Local privacy protection classification based on human-centric computing," en, *Hum.-centric comput. inf. sci.*, vol. 9, no. 1, Dec. 2019.

[11] S. Ellis and J. Santagate, "The path to a thinking supply chain," *IDC Technology Spotlight*, 2020.

[12] mhugos, *Five supply chain drivers*, en, https://www.scmglobe.com/five-supply-chain-drivers/, Accessed: 2022-5-20, Mar. 2020.

[13] A. K. Shrestha, S. Joshi, and J. Vassileva, "Customer data sharing platform: A blockchain-based shopping cart," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Toronto, ON, Canada: IEEE, May 2020.

[14] Z. Leng, Z. Tan, and K. Wang, "Application of hyperledger in the hospital information systems: A survey," *IEEE Access*, vol. 9, pp. 128 965–128 987, 2021.

[15] *A blockchain platform for the enterprise — hyperledger-fabricdocs main documentation*, en, https://hyperledger-fabric.readthedocs.io/en/release-2.2/, Accessed: 2022-5-20.

[16] A. Basak, M. M. Israfil, and S. Seddiqe, *Supply chain management of super shops in perspective of bangladesh*, https://globaljournals.org/GJMBR_Volume14/6-Supply-Chain-Management-of-Super.pdf, Accessed: 2022-5-20.

[17] *Channels — hyperledger-fabricdocs main documentation*, en, https://hyperledger-fabric.readthedocs.io/en/latest/channels.html, Accessed: 2022-5-20.

[18] *What is supply chain management?* en, https://www.ibm.com/topics/supply-chain-management/, Accessed: 2022-5-20.