TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ENBO CHEN

AN APPROACH FOR IMPROVING TRANSPARENCY AND TRACEABILITY OF INDUSTRIAL SUPPLY CHAIN WITH BLOCKCHAIN TECHNOLOGY

Master of Science Thesis

Examiner: Professor José L. Martínez Lastra and Dr. Andrei Lobov
Examiner and topic approved by the Faculty Council of the Faculty of Engineering Sciences on 5th October 2016

ABSTRACT

ENBO CHEN: An Approach for Improving Transparency and Traceability of Industrial Supply Chain with Blockchain Technology
Tampere University of technology
Master of Science Thesis, 74 pages
November 2017
Master's Degree Program in Automation Engineering
Major: Factory Automation and Industrial Informatics
Examiner: Professor José L. Martínez Lastra and Dr. Andrei Lobov

Keywords: blockchain, supply chain, tracking, smart contract, distributed network

Nowadays, the modern supply chain is facing the new threats and opportunities due to quality, safety, ethics, environmental impact and other serious problems aroused by the opacity of supply chain. On the contrary, a transparent and traceable supply chain can help suppliers minimize fraud and errors, enhance inventory management, reduce courier costs, lower waste and delay. Consequently, transparency and traceability are essential to the sustainable development of industrial supply chain in the future.

Driven by growing demand for transparency and traceability from consumers, companies, and governments, some fundamental labeling technologies (e.g. RFID, QR code, NFC tag, etc.) have been already combined with web technology and applied in logistics system to identify a product with origin information. Even though, those traditional technologies fail to provide a trusted and cost-efficient system to share information and record provenance.

This thesis aims to find an approach to improve transparency and traceability of supply chain in a secure and cost-efficient way. To achieve this goal, an approach with an emerging technology is presented in this thesis: blockchain, a shared, distributed and permissioned ledger that records every transaction information associated with asset through supply chain, which is synchronized and verified in real time with all entities in the supply chain but can be accessed only by authorized participants.

The result of this research work not only provides in-depth research of blockchain technology but also proposes a concrete solution with an implementation of blockchain technology to shape a transparent and traceable supply chain network. This solution can track provenance and trajectory of an asset through the complex supply chain in real time, at the same time, provides unprecedented visibility and confidentiality. Finally, this thesis also shows a possibility to integrate the blockchain system with other web service and traditional enterprise resource planning system.

PREFACE

“I went to the woods because I wished to live deliberately,
to front only the essential facts of life,
and see if I could not learn what it had to teach,
and not, when I came to die, discover that I had not lived.
I did not wish to live what was not life, living is so dear;
nor did I wish to practice resignation,
unless it was quite necessary.
I wanted to live deep and suck out all the marrow of life,
to live so sturdily and Spartan-like as to put to rout all that was not life,
to cut a broad swath and shave close,
to drive life into a corner,
and reduce it to its lowest terms.”

— Henry David Thoreau, Walden

This page is probably the most irrational one in the whole thesis.

Firstly, I would like to thank my father and mother for all the financial and emotional support(首先，感谢我的父母给予在物质上和精神上的全力支持). I fully appreciate all the help and support from my girlfriend July. I thank my bunny Hesse for accompaniment for the long night of writing thesis.

Next, I thank startup company Wone.io (Daren Tuzi, Otto Liuhtonen, Timo Siukkola) to bring me to the blockchain world. I can never forget the exciting “aha” moment when I first heard about blockchain technology.

I sincerely thank to all the great help from Professor José L. Martínez Lastra, Wael Mohammed and Borja Ramis Ferrer and all the discussion of MSc FAST Club.

I greatly appreciate Futurice Master's Thesis Bootcamp™ sponsored by Futurice Oy. Thanks to Mike Arvela and Emilia Kyllönen to make this happened for me. I appreciate all the guidance from my teacher Vilma Lehtinen, mentor Juho Vähä-Herttua and all comrades in Bootcamp.

I feel grateful to free and high-quality education from Tampere University of Technology and this 100-year-old amazing country – Finland.

Last but not the least, I would specially thank to my thesis supervisor Andrei Lobov who always say, “Let’s make world to a better place!”, “Sky is the limit!”, “Life is beautiful/crazy!”. He has been always challenging and inspiring me during this thesis and all my master studies.

Tampere, 22.11.2017

Enbo Chen

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ACL	Access Control Language
API	Application Programming Interface
BFT	Byzantine Fault Tolerance
B2B	Business to Business
CA	Certificate Authority
CLI	Command Line Interface
CP	Committing Peer
ECDSA	Elliptic Curve Digital Signature Algorithm
EP	Endorsing Peer
EPC	Electronic Product Code
ERP	Enterprise Resource Planning
HTTPS	Hypertext Transfer Protocol Secure
IT	Information Technology
JSON	JavaScript Object Notation
MSP	Membership Service Provider
NFC	Near-field communication
OP	Ordering Peer
P2P	Peer to Peer
PBFT	Practical Byzantine Fault Tolerance
PO	Purchase Order
PoW	Proof of Work
QR	Quick Response
REST	Representational state transfer
RFID	Radio-frequency identification
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SWOT	Strength Weakness Opportunity Threat
TLS	Transport Layer Security
UI	User Interface

Hn	Horizontal layer
Vn	Vertical layer
→	Physical flow
⇢	Information flow

1. INTRODUCTION

The first chapter of this thesis introduces an overview of the research context and problem statement. Motivation and objective will be also included in this section to help audience understand the impulsion and determination of solving the problem. Finally, research limitation and structure of thesis will bring a general outline and scope of this thesis.

1.1 Motivation

Today, Transparency and traceability are pivotal and imperative to sustainable development of industrial supply chain because of a convergence of issues: conflict resource, deforestation, sweatshop labor (including health and safety), anti-corruption, product safety, product defect, risk management and other serious issues [1]. All these problems force companies to explore an appropriate approach to trace and track the origin, history, distribution, current status of products, parts or materials in the complex environment. Tracking trajectory of product through the chain not only enhance the relationship with downstream and customers, but also gain company's control and expects of its upstream suppliers.[2]

Driven by growing calls for transparency, some technologies, such as RFID(Radio-frequency identification), QR code, Electronic Product Code(EPC)[3][4], pave a practical way to track and trace goods in supply chain, but there are still some key issues not solved effectively and completely. First of all, how to preserve and share the sensitive and valuable data in a trusted and secure way has been the most challenging issues with traditional practices. Moreover, centralized database and traditional system cannot prevent fraud and errors of origin information flowing from upstream to downstream, which make customers and downstream firms extremely hard to access and verify the provenance of assets.

This thesis works on an approach to improve transparency and traceability of industrial supply chain as well as keep confidentiality and interoperability and by applying blockchain technology. Blockchain is a shared, immutable, trusted ledger for storing the history of transactions and sharing the data with authorized participants. By decentralized network and smart contract, trusted and tamper-proof information of product is securely and immutably recorded from the beginning to the end of the supply chain. Meanwhile, the cost of human involvement required to create, execute and enforce a contract can be reduced dramatically by smart contract, which perform a predetermined action (transaction etc.) once certain conditions are met and all those action are traceable in the future.[5]

By this approach, blockchain can help to enable unprecedented, secure transparency across global supply chain, which will eliminate fraud and errors, improve logistics and supply chain management, minimize the cost, reduce waste and delay. According to estimates from IBM, blockchain is able to raise global GDP by almost 5% as well as total trade volume by 15%[6].

1.2 Problem Statement

Traceability and transparency is one of the most problematic issues in modern supply chain. Thus, tracking the provenance of goods on the blockchain reduces risk and increases quality in production and distribution.

However, there are some critical problems following:

- Origin information is not immutable because of centralized data system
- The provenance of product is not visible and traceable through supply chain because of the cost and complexity
- Mistrust between organizations, including fear that information might be passed on to a competitor has stopped organizations from sharing data
- There is not secure system to share and pass origin information across supply chain
- Integration with existing ERP system and information platform is missing

This thesis helps to address these problems by tracking the trajectory of assets in a cost-efficient, trusted, distributed blockchain network. It can provide unprecedented visibility into where things are in real time, but also traceability, showing where things have been before. At the same time, confidentiality and interoperability are ensured to work in the complex business world.

1.3 Objectives

The purpose of this thesis is to suggest an approach to shape a transparent and traceable supply chain by applying blockchain technology. The approach should be as comprehensive, feasible, realistic as possible for real industrial supply chain. Use case and validation scenarios will be designed and implemented to verify the feasibility of solution.

To fulfil the objectives, this thesis mainly focuses on below:

- Transparency: Purpose a shared and distributed ledger that is synchronized with immutable data and verified with all the authorized participants across supply chain in real time
- Traceability: Enable permissioned visibility of activities and reveals the real-time location, ownership history and condition of any assets
- Confidentiality: Data records on the blockchain can be accessed only by authorized participants with permission. It can be widely shared and protected at the same time
- Interoperability: integration with the existing systems and web service

1.4 Research Limitations

This thesis mainly focuses on the achievement of enhancing transparency and traceability of industrial supply chain powered by blockchain technology.

Even through one of the key advantages of blockchain is that it should be much more secure than traditional IT solutions[7]. However, the reliability and stability are most challenges in today's complex industrial supply chain. This thesis will cover the basics of the safety mechanism of blockchain technology and platform used, but no more in-depth studies on cybersecurity.

Another potential use of blockchain and smart contract is reducing the cost of doing business in global supply chains by issuing the automatic transactions between companies and suppliers[8]. Although, this thesis focuses more on information flow instead of payment through supply chain.

Also, the authenticity of the record entry is not assured but need to be supervised because the immutable record in blockchain does not indicate that its corresponding counterpart material or product in the physical world has not been forged [9].

In the future, more research and trials need to be done comprehensively before blockchain reaching the enterprise adoption and becoming irreplaceable mainstream.

1.5 Structure of Thesis

This thesis has five main parts. All the description of each chapter is following:

Chapter 1 mainly defines motivation, problem statement, objectives of this thesis as well as the limitation of research.

Chapter 2 describes the state of the art of the current status and challenge of modern supply chain. Besides, it introduces blockchain technology and its implementations.

Chapter 3 dive into methodology of creating a supply chain model on the blockchain network.

Chapter 4 dive into implementation, which is consist of use cases, validation scenarios.

Chapter 5 analyze the results of validation scenarios and inspect the influence blockchain have in four aspects: transparency, traceability, confidentiality and interoperability.

Chapter 6 discusses the significance and credibility of the founding of the result with the validated solution.

Chapter 7 concludes with the summary of the research and recommended areas for future work.

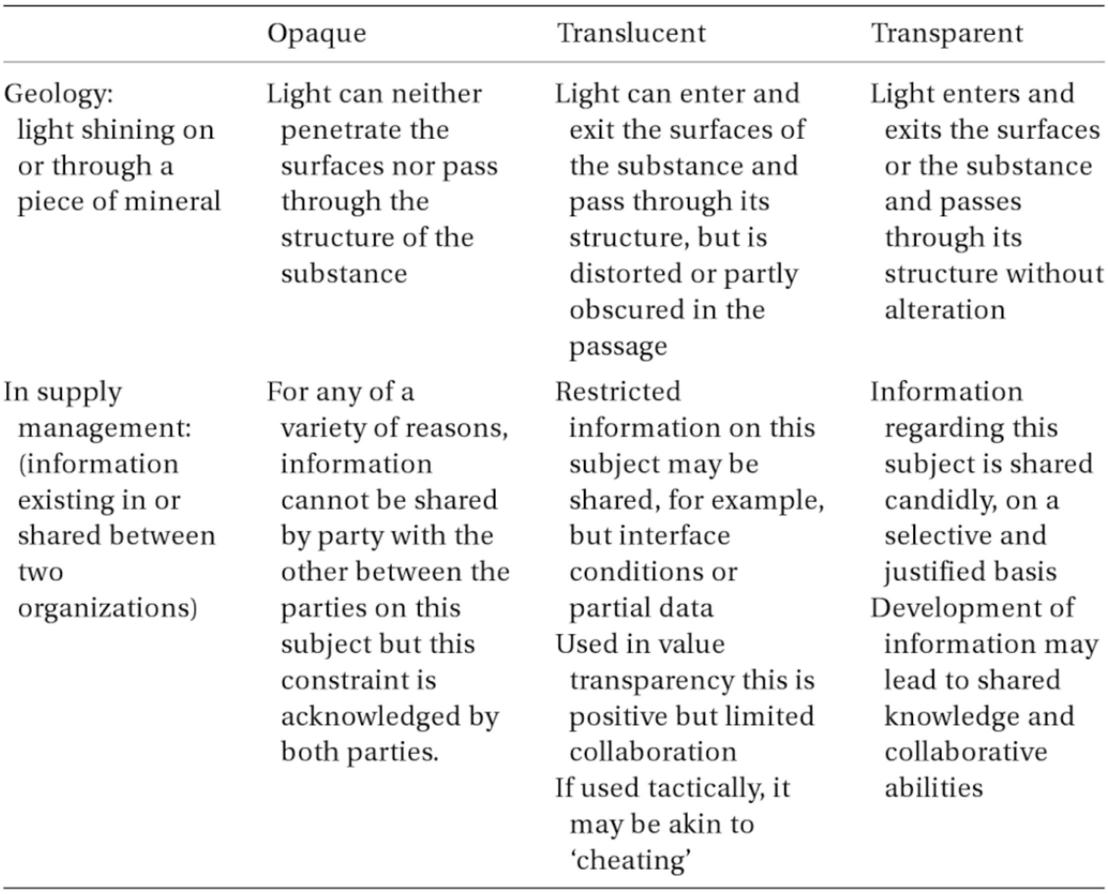
2. STATE OF THE ART

This chapter describes the most recent stages of supply chain in terms of transparency and traceability. Then, it illustrates possible technologies already applied in supply chain and disadvantages of them. Network topology is mentioned to compare the difference between traditional and modern business network. Last but not least, it introduces the cutting-edge blockchain technology in general.

2.1 Transparency and Traceability of Supply Chain

Table 1 shows transparency definition in supply chain management as derived from a geological metaphor in [10]. Transparency is defended as "Information regarding this subject is shared candidly, on a selective and justified basis. Development of information may lead to shared knowledge and collaborative abilities".

Table 1. *Transparency definition in geology and supply management[10]*



	Opaque	Translucent	Transparent
Geology: light shining on or through a piece of mineral	Light can neither penetrate the surfaces nor pass through the structure of the substance	Light can enter and exit the surfaces of the substance and pass through its structure, but is distorted or partly obscured in the passage	Light enters and exits the surfaces or the substance and passes through its structure without alteration
In supply management: (information existing in or shared between two organizations)	For any of a variety of reasons, information cannot be shared by party with the other between the parties on this subject but this constraint is acknowledged by both parties.	Restricted information on this subject may be shared, for example, but interface conditions or partial data Used in value transparency this is positive but limited collaboration If used tactically, it may be akin to 'cheating'	Information regarding this subject is shared candidly, on a selective and justified basis Development of information may lead to shared knowledge and collaborative abilities

The original definition of traceability in ISO[1](international Organization for Standardization) is "the ability to identify and trace the history, distribution, location, and application of products, parts, and materials"[11]. Under EU law[12], traceability means "the ability to track any food, feed, food-producing animal or substance that will be used for consumption, through all stages of production, processing and distribution." In 2012, international non-profit organization GS1[2] , who is dedicating to standardizing barcoding widely in the world, stated "traceability data" in Global Traceability Standard: "what is it? (i.e., the traceable item), who has been involved? (i.e., the traceability partner(s)), where did it happen? (i.e., location), when did it happen? (i.e., date / time, period of time) what happened? (i.e., process or event)"[13].

2.2 Network Topology

Network topology is the virtual shape or structure of a network presetting physical or logical arrangement of the elements (links, nodes, etc.) of a communication network[14][15]. This section will discuss about the typical topology of business network in real world.

2.2.1 Topology of Internet

In 1964, Paul Baran began rethinking about the optimization of structure of the Internet. As shown in Figure 1, Paul proposed three possible topology of Internet: centralized(or star), decentralized(mixture star and mesh), and distributed(or grid or mesh).[16]

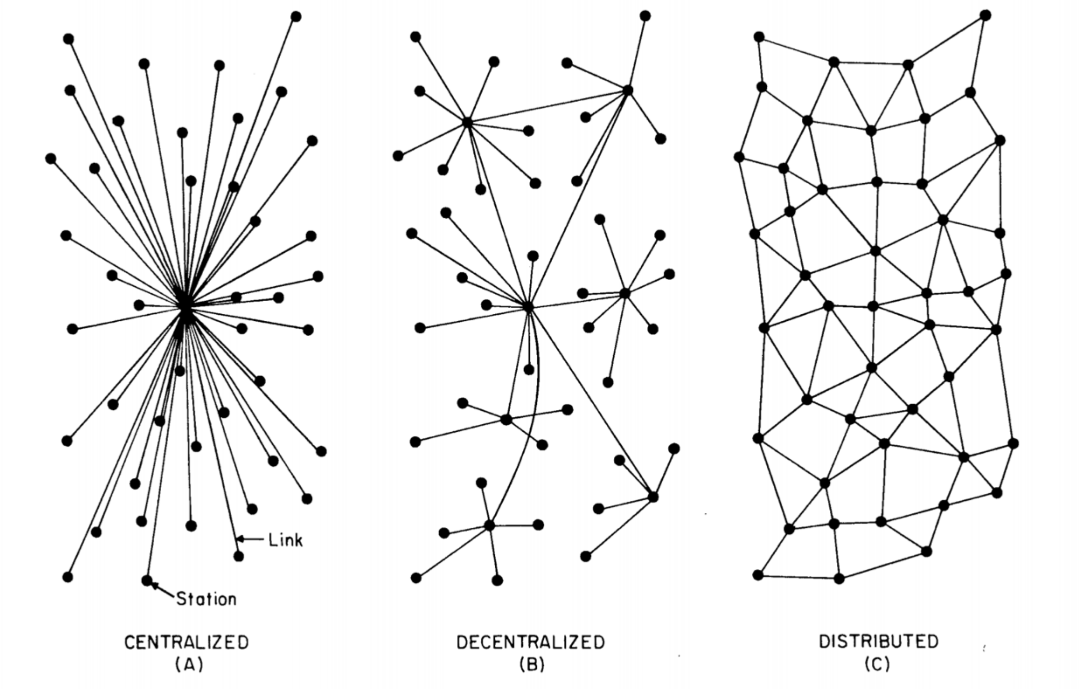


Figure 1. *Centralized, decentralized and distributed networks by [16]*

[1] https://www.iso.org/home.html
[2] https://www.gs1.org/

He proposed a distributed network of unmanned nodes, acting as switches and routing information from one node to another until to the final destinations. While he asserted the centralized and decentralized networks were vulnerable to attack, the third distributed structure would be more resilient. So he suggested the Internet's structure should be like a distributed network.[17]

2.2.2 Topology of Business Network

Even there is no such a big change of business network since business records have been developed and kept. Members transact with each other in the network and maintain their own ledger separately. Centralized authority trusted by all participants takes over the exchange for all business, which make process slow and expensive. Even though there are strong needs of transparency, traceability and trust, today's approach is not possible to share information and process with efficiency and trust.

Transition of business network in the future might be similar to Paul's expectation of Internet, shifting from centralized to decentralized. See Figure 2:

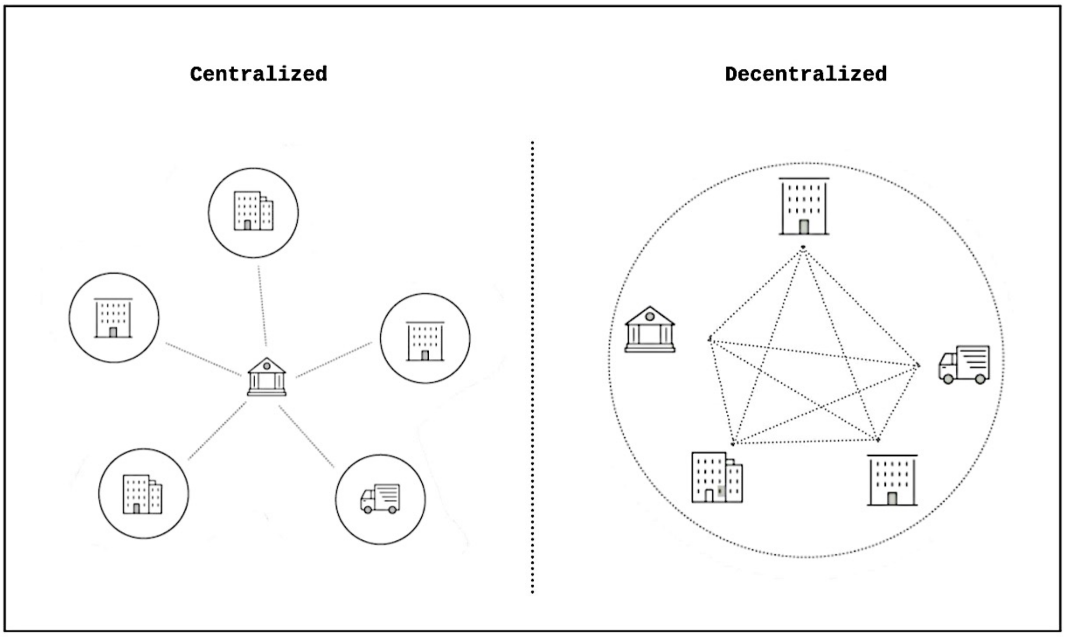


Figure 2. *Centralized and decentralized business network [18]*

In future's decentralized network, all participants can transact with each other directly and freely without 3rd party involved. There will be a unified system for managing the identity of all participants and processing the transactions in the network. Provenance of goods is track and trace for future's use. Security is ensured and privacy is protected. Contracts can be signed and executed automatically by machine instead of manually by human. [19]

2.3 Blockchain Technology

Blockchain technology is a peer-to-peer distributed ledger by combining cryptography, consensus mechanism, smart contracts and other assistive technologies[20]. This section will introduce this technology from five aspects: network, data structure, cryptography, consensus mechanism and smart contract.

2.3.1 Data Structure

The essence of chain in blockchain is about the characteristic of data structure. In white paper of Bitcoin [21], Satoshi Nakamoto reclaimed bitcoin as a ordered, back-linked list of block, encapsulating hash of previous block, nonce, timestamp and Merkle root of all transaction. The data structure of bitcoin was succeeded by other implementations and foundation of blockchain technology. See Figure 3:

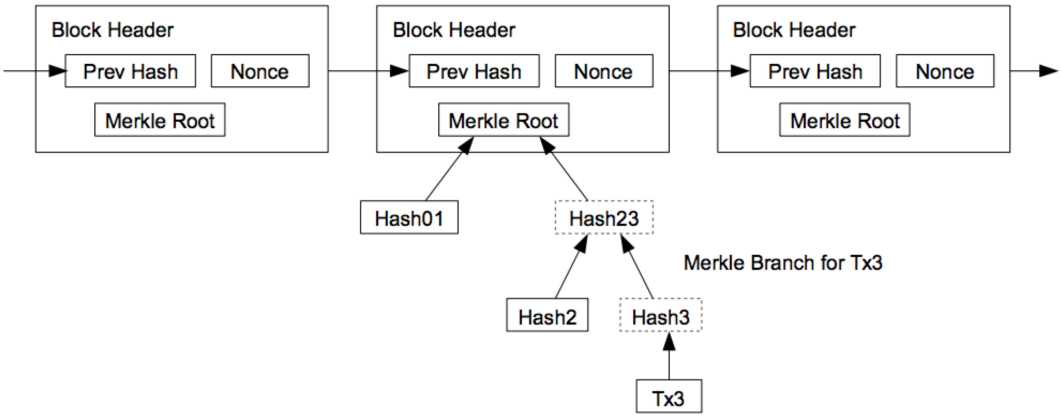


Figure 3. *Longest Proof-of-Work Chain from Bitcoin by [21]*

In bitcoin, each block is identified by a hash(SHA256) on the header and a reference to the hash of previous block in the chain. Nonce and difficulty target are the core elements for proof-of-work algorithm. Moreover It uses Merkle tree to summarize and verify the integrity of all the transactions in the block[22].

2.3.2 Distributed P2P Network

Blockchain is built on the distributed Peer-to-Peer(P2P) network used to propagate transactions and broadcast to all nodes in the network[23]. In Figure 4, P2P network has equivalent privileges, capabilities and responsibilities, whereas server/client network has one centralized and powerful server serves resources or services to clients.

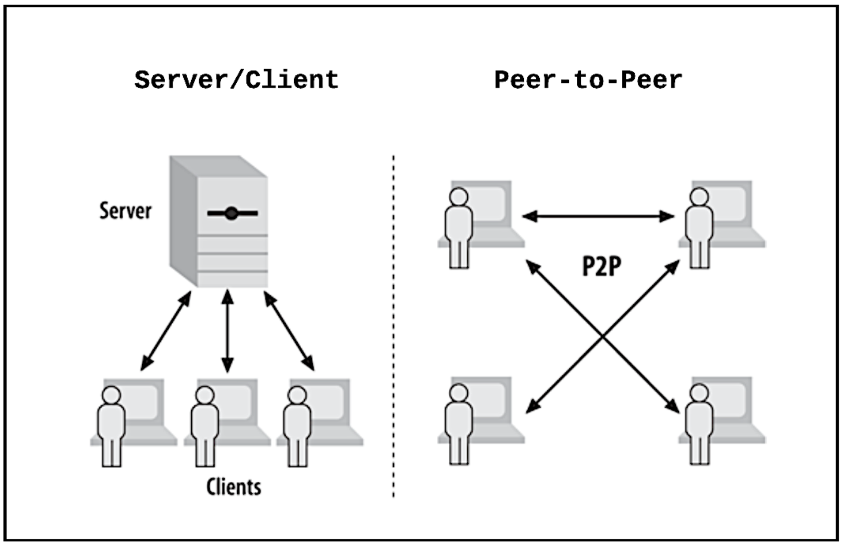


Figure 4. *Server/Client network and peer-to-peer network [24]*

Compared to Server/Client network, P2P network is more reliable and resilient as each peer can has own copy of the common data. However, since peers are allowed to take control access of data in the network, security and privacy would be problematic without right configuration.

2.3.3 Cryptography

To ensure the security and integrity of the information stored in the blockchain, a large number of modern cryptographic techniques including cryptographic hash function and elliptic curve public-key cryptography are used in the definition and construction of blockchain. Meanwhile, these cryptographic techniques are also used to design consensus algorithms based on proof of work as well as identification of users.[25]

The following cryptographic techniques are widely used in most blockchain implementation (especially bitcoin):

1. Hash Algorithm: Bitcoin mainly use SHA256 and RIPEMD160 to hash everything
2. Merkle Tree : The tree in the blockchain is a binary tree that stores the transaction information and performs its integrity verification process
3. Public Key Cryptography: Use key pair to control access to blockchain, which contains a private key and a unique public key, which the former is derived from the latter. Private key is used to issue digital signature and allow spending, while public key verifies signature and represents the address of wallet[21]

However, in [20], some compromise of underlying cryptographic algorithm(SHA256, RIPEMD160, ECDSA) still have possible threats and impact on security of blockchain.

2.3.4 Consensus Mechanism

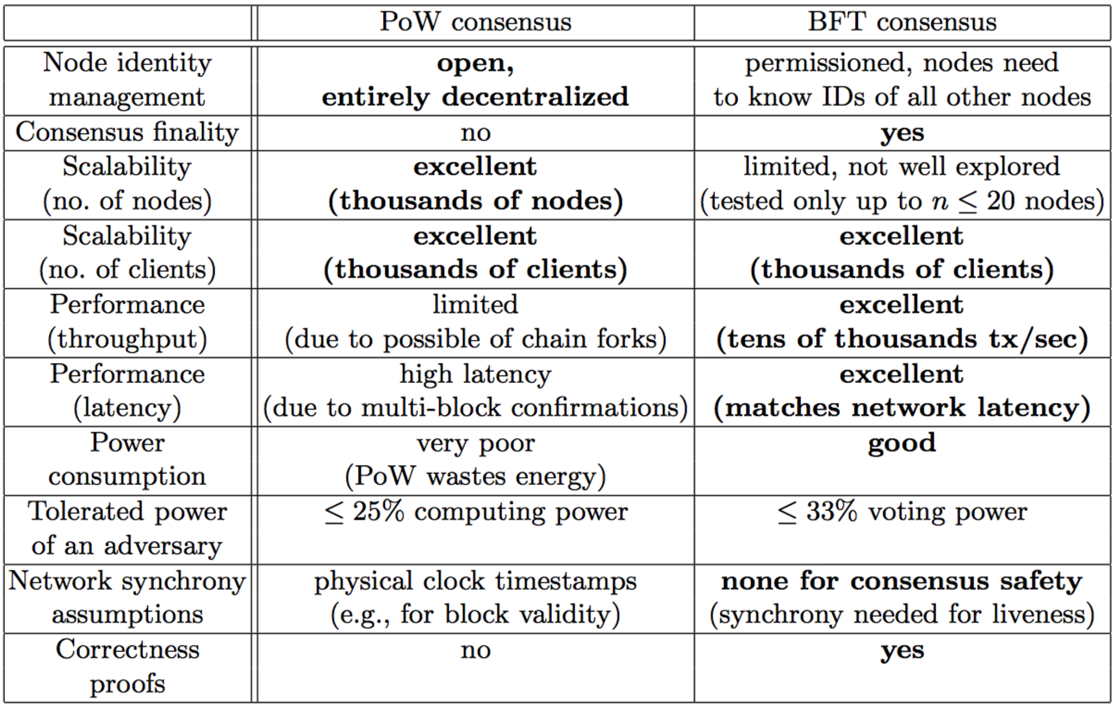
Consensus mechanism is used to authenticate and validate transactions on blockchain without a central authority[26]. It plays a crucial role in blockchain technology to keep distributed ledger synchronized securely through a collaborative process[18]. There are two most used consensus protocols are: PoW (Proof of Work) and BFT (Byzantine Fault Tolerance).

PoW is the consensus algorithm used in bitcoin. The main idea is to hashing power competition to solve the difficult mathematical problem (similar to HashCash). The first one that resolve the math problem can create the next block and get a certain amount of rewards. This mechanism PoW is energy-consuming because of the workload taken as the safeguard. If adversary try to generate an alternate chain faster than the honest chain, it will take more than 50% of the world's hashing power to tempering it, from which the gains from forging can be much greater than the cost. Thus, PoW can ensure safety of the blockchain effectively but not efficiently.[27]

BFT (Byzantine Fault Tolerance), known as the Byzantine Generals Problem, designed for fault tolerant to achieve consensus even a small portion of malicious nodes are existing. PBFT (Practical Byzantine Fault Tolerance) is one of the implemented algorithm introduced by Miguel Castro and Barbara Liskov in 1999[28]. This algorithm is initially devised for solving practical problems with low-latency and efficiency. There are three phases in the whole process in this algorithm: pre-prepared, prepare and commit. A node has to received votes from over 2/3 of all nodes to enter next phase to reach the final consensus. PBFT could work when malicious nodes is less than 1/3 of the total nodes in the network.[29]

A high-level comparison between PoW consensus and BFT consensus in given in Table 2. There are a handful of important blockchain properties: node identity management, consensus finality (the possibility of changing past transactions due to temporary forks in the blockchain), scalability with regard to number of consensus nodes and clients, performance in terms of latency, throughput and power consumption, tolerated power of adversary, network synchrony assumption, correctness proofs of protocols. [30]

Table 2. *High-level comparison between PoW and BFT consensus families by [30]*



	PoW consensus	BFT consensus
Node identity management	**open, entirely decentralized**	permissioned, nodes need to know IDs of all other nodes
Consensus finality	no	**yes**
Scalability (no. of nodes)	**excellent (thousands of nodes)**	limited, not well explored (tested only up to $n \leq 20$ nodes)
Scalability (no. of clients)	**excellent (thousands of clients)**	**excellent (thousands of clients)**
Performance (throughput)	limited (due to possible of chain forks)	**excellent (tens of thousands tx/sec)**
Performance (latency)	high latency (due to multi-block confirmations)	**excellent (matches network latency)**
Power consumption	very poor (PoW wastes energy)	**good**
Tolerated power of an adversary	$\leq 25\%$ computing power	$\leq 33\%$ voting power
Network synchrony assumptions	physical clock timestamps (e.g., for block validity)	**none for consensus safety** (synchrony needed for liveness)
Correctness proofs	no	**yes**

In summary, PoW and BFT are two opposite blockchain consensus technologies of today in terms of node identity and scalability. PoW consensus algorithm provides excellent node scalability with limited performance, while BFT consensus algorithm has good performance but limited scalability[30]. Thus, the selection of BFT consensus should depend on the requirement and context of business world.

2.3.5 Smart Contract

A smart contract is “a set of promises, specified in digital form, including protocols within which the parties perform on these promises”, which were first proposed by Nick Szabo in 1996 [31]. In other words, smart contract is a digital form of contractual clauses embedded as code in software to mediate actions (e.g. release of payments) by rules-based operations[32]. Once the smart contract is initiated and the condition is met, the progress of action is automated and irrevocable as predefined in the logic[33].

Figure 5 shows a simple progress of how smart contract works to provide services. Participant 1 and participant 2 agree the cost of shipment depending on the time the goods arrive. The rules agreed by both parties and recorded into blockchain, then the appropriate payment will issue to participant 2 automatically when participant 1 receive the goods. [18]

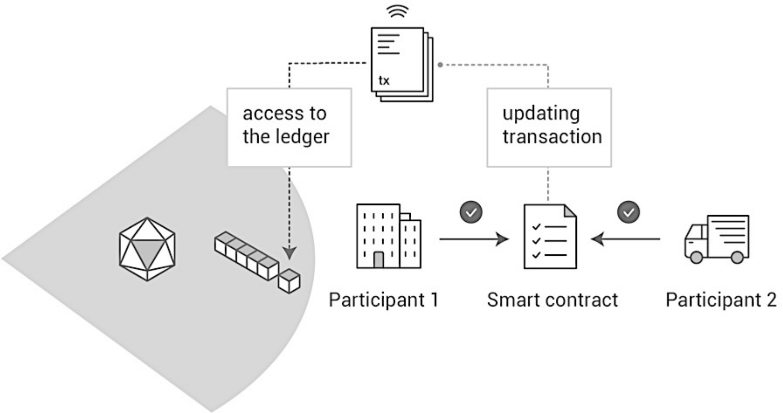


Figure 5. *Smart contract [18]*

Smart contract has unprecedented potentials to solve problems such a dispute resolution, autonomous organization[34], energy auctions[35], legal testament[36] and so on. However, smart contract would face some practical problems like legally binding contractual effect, legal enforceability, jurisdictional variations[32].

2.4 Blockchain Implementations

In 2008, the first and the most successful blockchain implementation born with Satoshi Nakamoto, an anonymous person or group, publishing a white paper called Bitcoin: A Peer to Peer Electronic Cash System. He purposed a digital currency to serve as public ledger for transactions on peer-to-peer network. After inspiration of bitcoin[3], different implementations are being developed for different purpose. This section will present the three main kinds of blockchain implementations and corresponding representatives.

2.4.1 Types of blockchain implementations

There are three main categories of blockchain differed by openness of network: Public, private and permissioned blockchain.

Table 1 shows the high level comparison between public, private and permissioned blockchain , which is conclusion from [29], [37], and [38]. The table shows the main difference from different perspectives: network, definition, benefits, challenges, cost efficiency, performance, failure points, which helps to make the right decision when designing blockchain application based on different context.

[3] https://bitcoin.org/en/

Table 1. Comparison between public, private and permissioned Blockchain

	Public	Private	Permissioned
Network	Decentralized	Partially decentralized	Partially decentralized - hybrid between public and private blockchains
What is it?	Anyone anywhere in the world can read and write on the network. Data is validated by every participant ("node") in the network, thus making it very secure	Permissions to read and write data onto blockchain are controlled by single "highly trusted" organization - the owner of the blockchain	Permissions to verify, read and write on the blockchain controlled by a few predetermined nodes. The choice of predetermined nodes can be different for every entity on the blockchain
Benefits	- Secure as the entire network verifies transaction - Transparent as all transactions are made public with Individual anonymity	- Efficient as verification is done by just owner of the blockchain - Private as the owner can control who has access to read or write on the blockchain	- Efficient as relatively lesser nodes verify transactions - Private as read and write access can be controlled by the predetermined nodes. - No consolidation of controlling power
Challenges	Inefficient as all nodes need to verify the transaction	- Controlling power is consolidated to a single organization. - Difficult to align many organization to use the same blockchain	
Cost Efficiency	Low	High	Medium
Performance	Bad	Great	Good
Failure points	Majority (nodes, power, stake)	1	*

Generally, the more decentralized and open network of blockchain is, the less efficient and worse performance the blockchain implementation has. This is because the more resource consuming and more time to reach the final consensus of shared ledger, which highly depends on which consensus algorithm the system use.

2.4.2 Public Blockchain

In a public blockchain, every entity in the network share the equal right to read and write the data in blockchain by following the common rule they agree. Anyone can be a member of the blockchain network and validate transactions with required software and hardware. Thus, public blockchain is completely decentralized because there is no centralized entity to control and manage rules in the connected network. [39]

Public blockchain is the system for business network where transparency is specially needed and trustless to third party. Bitcoin and Ethereum[4] are the most common implementations of public blockchain. They are both platforms for making transactions directly between users without third party. The both use PoW (Proof of Work) consensus protocol to secure network and finalize transitions. However, because of the characteristic of PoW, the speed of transaction is slower than traditional database and power consumption is not eco-friendly.

2.4.3 Private Blockchain

On the opposite spectrum of public blockchain, private blockchain has a fully centralized structure. Restricted permission to access data in blockchain brings much better privacy to participants in the network compared to public blockchain. However, the single entity in control has the full privilege and power to take decisions and change rules in the blockchain network[40].

Private blockchain is suitable for cases where public readability or audit are not necessary. Besides, a high trust should be established between the single entity and other participants. Also, compared to public blockchain, private blockchain has a faster transaction speed as well as lower transaction fees. Human intervention is allowed to fix any faults and easily approved by the users.[39]

2.4.4 Permissioned Blockchain

Permissioned blockchain, also called consortium blockchain, is a hybrid manner of public blockchain and permissioned blockchain. It is partially decentralized and only few participants have the right to access and validate the transactions. The right depends on the identity of the participants and the role they are playing in the business world. Permissioned blockchain usually require smart contract functionality to perform business logic and validate identity before executing transactions. [39]

Two implementations of permissioned blockchain are Hyperledger[5] and Corda[6]. Both are open-source distributed ledger platform designed to record and shared data as well as keep privacy and security to support business transactions in the complex context. [41]

2.5 Hyperledger Fabric

In this thesis, Hyperledger Fabric is the blockchain platform used to develop business network of industrial supply chain. It is a distributed ledger with elastic and extensible architecture to help participants to manage transactions with smart contracts. Besides, it supports pluggable implementations of components like consensus mechanisms and

[4] https://www.ethereum.org/

[5] https://hyperledger.org/

[6] Corda: https://www.corda.net

data format.[42] It is one of projects currently in incubation under the Hyperledger Project.

The difference between Hyperledger Fabric and other blockchain system are private and permissioned. Compared to bitcoin or other open public blockchain, Hyperledger Fabric provide membership service to issuer and validate identity of participants in the network to secure privacy and accommodate the complexity of modern supply chain. [43]

2.5.1 Shared Ledger

There are two components composing shared ledger: the world state and the transaction log. A copy of the ledger is available to every participant in the network they belong to.

The world state component manages the current state of the ledger, which is database of the ledger, at specific point of time. The data store for world state is pluggable and it is the LevelDB key-value store database by default. On the other hand, the transaction log component stores all the transactions that are update history (before and after values of ledger) of world state.

2.5.2 Smart Contracts

Chaincode is a program written in Go language and considered as Smart Contracts in Hyperledger Fabric. It runs in isolated environment in a secured Docker container. It handles agreed business logic and enables the interaction with the world state component of the ledger.

2.5.3 Privacy

Hyperledger Fabric emphasize privacy as a key concept in the network because the information in a Business-to-Business (B2B) might be extremely confidential and sensitive. Thus, it provides multiple channels for separate ledger of transactions for different groups of participants. The copies of transactions will be only recorded in the specific ledger only accessible to participants who related to the transactions and in the same channel, not others like their competitors.

2.5.4 Consensus

Consensus is the process to finalize ordering and validation of transactions according to endorsement and consensus policies in the network. Hyperledger Fabric offers a selections of consensus mechanism (e.g. SOLO, Kafka, SBFT, etc.) to plug and use depending on relationships that exist between participants. Compared to the PoW (Proof of Work) consensus with anonymous miners, these approaches would save a cost of resources and time, which are more suitable for business blockchain networks.

2.6 Supply Chain and Blockchain

Blockchain has potential to help to create visibility and traceability to supply chain by tracking the provenance of goods. This can reduce risk and increate quality in long term. This section will reveal the current status of this technology applied in supply chain and corresponding opportunity and challenge in the future.

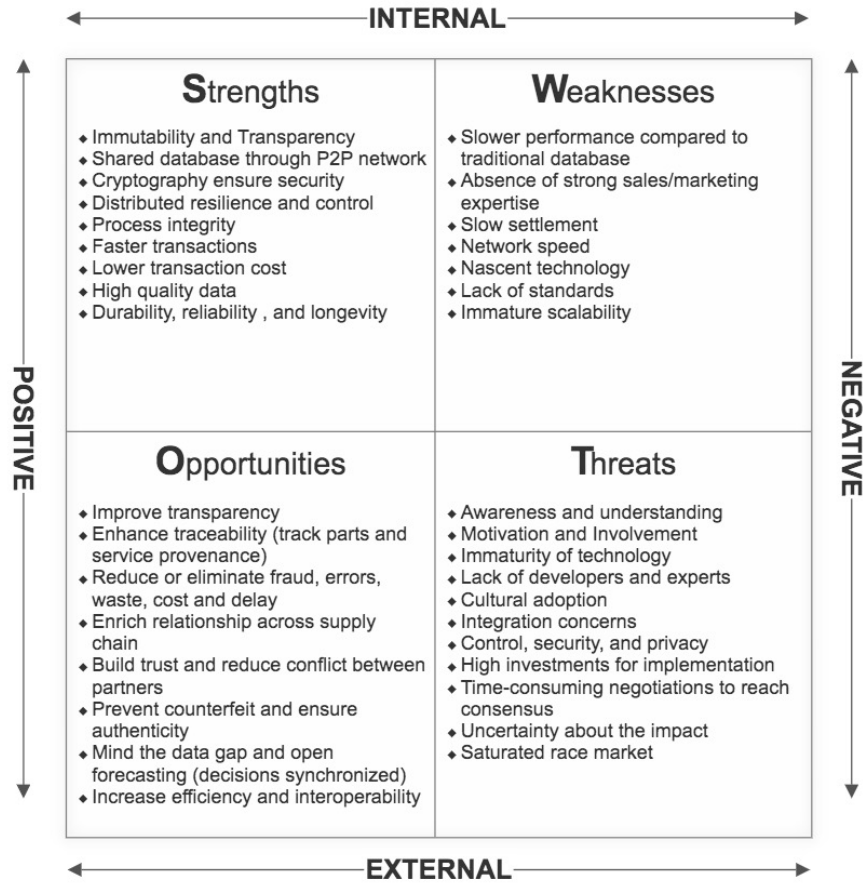
2.6.1 Current situation

There are a lot of research and application of blockchain have already been exploring in the field of supply chain. In the UK, Everledger use blockchain to track origin of diamond with detailed information such as diamond cut and quality, preventing conflict diamonds and stealing[44]. Another UK startup, Provenance certify the provenance of their products for retailers and producers in the food and drink industry, using blockchain and IoT technology[45]. A research of supply chain traceability system of food safety based on blockchain other technologies has been done in [46].

However, the widespread trial and application in practices are still lack but necessary before going to production. Moreover, more comprehensive researches have to be done for this newborn technology to face the complex business world.

2.6.2 Opportunities and Threats of applied blockchain

Figure 6 shows a SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis of blockchain applied in supply chain according to [47] and [48].This nascent technology has immense opportunities to revolutionize supply chain as well as threats to examine and confront before actual implementation.



Strengths

- Immutability and Transparency
- Shared database through P2P network
- Cryptography ensure security
- Distributed resilience and control
- Process integrity
- Faster transactions
- Lower transaction cost
- High quality data
- Durability, reliability , and longevity

Weaknesses

- Slower performance compared to traditional database
- Absence of strong sales/marketing expertise
- Slow settlement
- Network speed
- Nascent technology
- Lack of standards
- Immature scalability

Opportunities

- Improve transparency
- Enhance traceability (track parts and service provenance)
- Reduce or eliminate fraud, errors, waste, cost and delay
- Enrich relationship across supply chain
- Build trust and reduce conflict between partners
- Prevent counterfeit and ensure authenticity
- Mind the data gap and open forecasting (decisions synchronized)
- Increase efficiency and interoperability

Threats

- Awareness and understanding
- Motivation and Involvement
- Immaturity of technology
- Lack of developers and experts
- Cultural adoption
- Integration concerns
- Control, security, and privacy
- High investments for implementation
- Time-consuming negotiations to reach consensus
- Uncertainty about the impact
- Saturated race market

Figure 6. *SWOT analysis of blockchain applied in supply chain*

Blockchain create visibility to show where things are as well as traceability to track where things have been. Every event, attribute, data could be recorded in blockchain system in order to track the provenance of goods in the future. It will reduce or eliminate fraud, errors, waste, cost, and delay and increase efficiency and interoperability in the supply chain.

However, it is not a mature technology so that there are some potential barriers from technology, governance, organization and even society[33]. Lack of awareness and understanding of impact on supply chain will make blockchain not easy to adopt. Besides, the high investment for implementation and human resource would be intimidating. Last but not least, it is challenging to establish and maintain properly system for security and privacy of data in a complex business world[49].

3. METHODOLOGY OF CREATING SUPPLY CHAIN MODEL IN BLOCKCHAIN NETWORK

This chapter will present a methodology to create a supply chain model in blockchain network, from choosing a framework and implementation of blockchain to supply chain modelling.

3.1 Blockchain Selection

As illustrated in 2.4.1, there are mainly three categories of blockchain implementation depending on the openness of the network: public, private and permissioned blockchain.

Figure 7 shows a blockchain choice flow between traditional database, permissioned blockchain and public blockchain. There are a handful of the questions on this chart that will help to make the right decision based on specific requirement and situation.

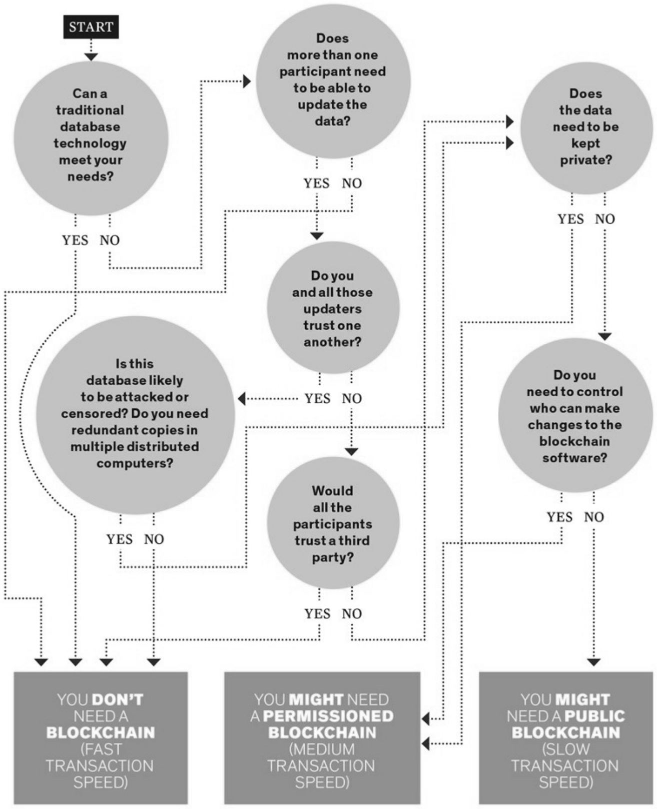

START
Can a traditional database technology meet your needs?
YES NO
Does more than one participant need to be able to update the data?
YES NO
Do you and all those updaters trust one another?
YES NO
Is this database likely to be attacked or censored? Do you need redundant copies in multiple distributed computers?
YES NO
Would all the participants trust a third party?
YES NO
Does the data need to be kept private?
YES NO
Do you need to control who can make changes to the blockchain software?
YES NO
YOU DON'T NEED A BLOCKCHAIN (FAST TRANSACTION SPEED)
YOU MIGHT NEED A PERMISSIONED BLOCKCHAIN (MEDIUM TRANSACTION SPEED)
YOU MIGHT NEED A PUBLIC BLOCKCHAIN (SLOW TRANSACTION SPEED)

***Figure** 7.Blockchain selection[50]*

To consider the situation and needs in industrial supply chain, in this thesis, permissioned blockchain is chosen because of following answers:

- The traditional database cannot meet the needs of modern supply chain
- More than one company or organization need to be able to update the data
- There is trust issue existing in the supply chain
- There is no third party can be trusted by all participants
- The data have to be kept private
- After all, the permissioned blockchain is more suitable in the research domain. However, there are different implementations of permissioned blockchain at this moment: Hyperledger, Corda[7], Multichain[8].

Compared to other implementations permissioned blockchain, Hyperledger is chosen by following reasons:

- Permissioned and private: In industrial supply chains, it is important to protect sensitive data for complying with data protection laws or regulations, which has to know who is accessing specific data
- Modularity: the modularity of Hyperledger enables enterprises to plug in their preferred encryption, consensus and other components because some multi-company networks already have identity management and some countries have their own encryption standards[51]
- Transaction Speed: transaction confirmation in seconds instead of minutes. The expected performance is 100,000 transactions per second in the standard production environment[52]
- Safety: For handling identity management and managing strong authentication, Hardware Security Module enhances protection for keys and sensitive data by providing PKCS11 for key generation[53]

3.2 Blockchain Framework and tools

In this thesis, Hyperledger Fabric[9] and Hyperledger Composer[10] are used as blockchain framework and developer tools. This section will introduce the overview of architecture and consensus flow.

3.2.1 Architecture Overview

As shown in Figure 8, there are mainly three layers in the architecture: Application, Developer Tools (Hyperledger Composer) and Blockchain Runtime (Hyperledger Fabric). Figure 8 shows the main components for each layer.

Communication between application layer and tool layer is via REST API. Application is able to call REST API generated by REST API server in Develop Tools. On the other hand, Developer Tools connects to Blockchain Runtime securely with connection pro-

[7] https://www.corda.net/
[8] https://www.multichain.com/
[9] https://github.com/hyperledger/fabric
[10] https://github.com/hyperledger/composer

files. The connection profiles contain the TCP/IP addresses and ports for the peers and cryptographic certificates[54].

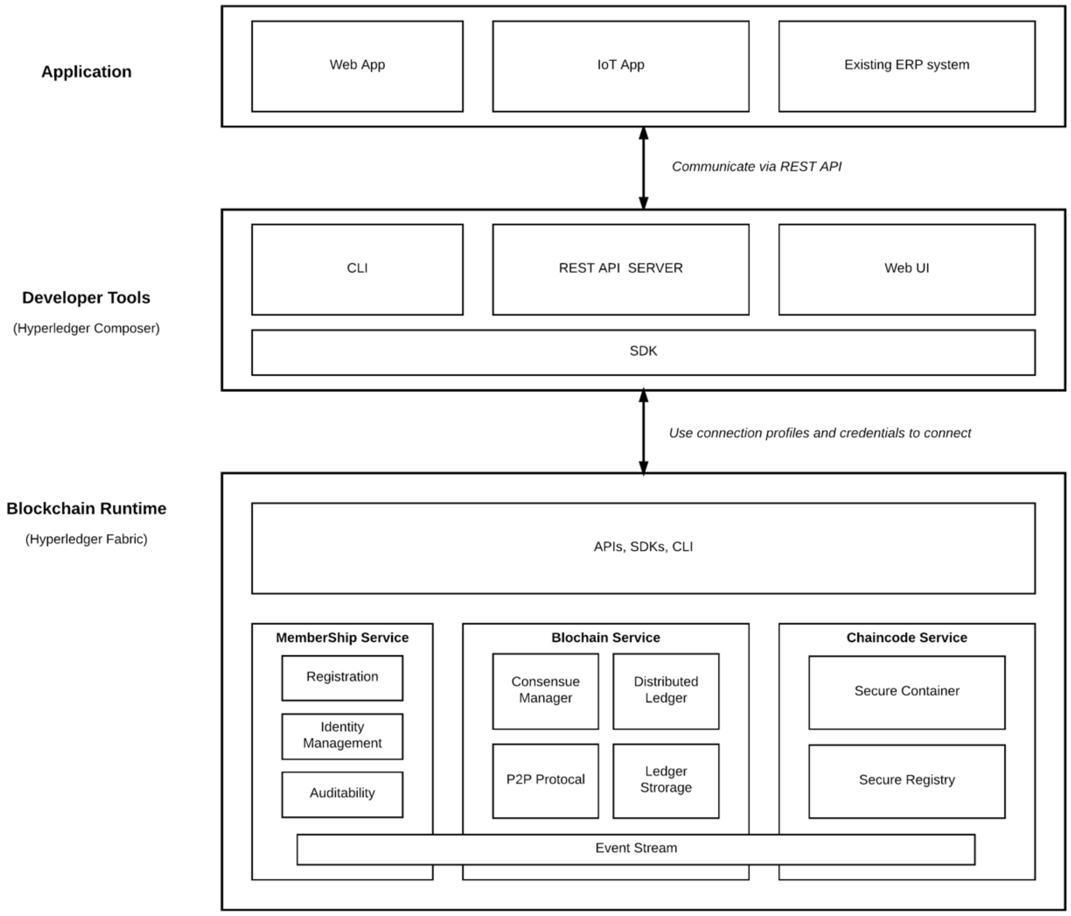


Figure 8. *Reference Architecture of system*

Application layer encompass components like web application, IoT application or existing ERP system.

Develop tools layer provides a handful of development toolset to enable rapid and easy development and testing of business logic running on blockchain runtime. This layer is composed of some high-level components: Command Line Interface (CLI), REST server, web user interface(UI) and JavaScript SDK. CLI facilitate developers and operators with efficient deployment and management of business network definitions. REST server is generated by Loopback connector to create the REST APIs for integration with application. Web UI is a playground that define and test business network quickly. SDK is set of Node.js APIs to perform operations(Create, Read, Update, Delete) of resource in blockchain distributed ledger.[55]

Blockchain Runtime is running distributed ledger where the state stores. The services in reference architecture aligned in three categories: Membership, Blockchain and Chaincode. Membership service is responsible for issuing and validating certificate and user authentication by abstracting cryptographic mechanisms and protocols[56]. Blockchain service is a core component of blockchain technology responsible for consensus, p2p network, ledger storage and distributed ledger. Chaincode service is the smart contract encapsulated in a secure and isolated sandbox, processing and validate transactions[57].

3.2.2 Consensus Process Flow

Figure 9 shows a generalized view of consensus process flow, where consensus interact with other architectural components. Distributed ledger reaches consensus by performing two separate activities:

1. Ordering of transactions
2. Validating transactions

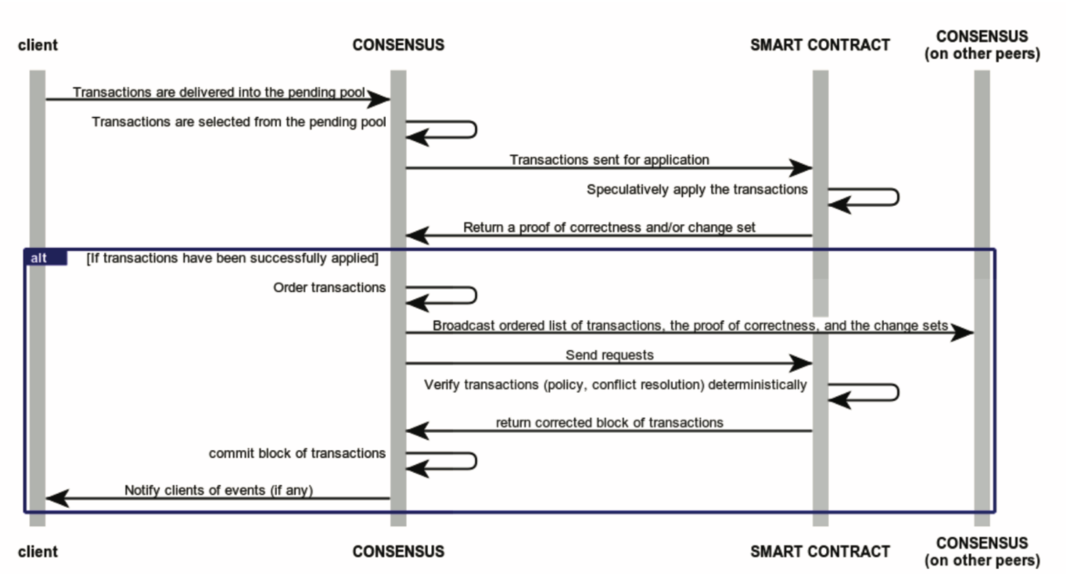

client
CONSENSUS
SMART CONTRACT
CONSENSUS (on other peers)
Transactions are delivered into the pending pool
Transactions are selected from the pending pool
Transactions sent for application
Speculatively apply the transactions
Return a proof of correctness and/or change set
alt
[If transactions have been successfully applied]
Order transactions
Broadcast ordered list of transactions, the proof of correctness, and the change sets
Send requests
Verify transactions (policy, conflict resolution) deterministically
return corrected block of transactions
commit block of transactions
Notify clients of events (if any)
client
CONSENSUS
SMART CONTRACT
CONSENSUS (on other peers)

Figure 9. *Generalized Hyperledger consensus process flow [57]*

First of all, client submits transactions to pending pool of consensus. Then ordering service in consensus select the specific number of transactions and order them based on pluggable consensus algorithm and configuration policy, then grouping them into a single block for the sake of efficiency. The content of transaction is encrypted so that ordering service is agnostic to the transactions.

Second step is validating transaction by smart contract(chaincode) to check if business logic of transaction conforms the policy and contract defined before. If transaction has been verified with proof of correctness received by consensus layer, then consensus will broadcast the ordered transaction with proof of correctness and change sets to other peers. After all, it commits the corrected block of transaction into distributed ledger.

The consensus algorithm in Hyperledger Fabric ordering service is based on permissioned voting-based, which means the leader takes ordering. Leader can be voted only when it is in-sync. The advantage of this consensus algorithm is providing crash fault tolerance and reaching consensus in seconds. [57]

3.3 Model

3.3.1 Supply Chain Modeling

There is no perfect model that can represent all supply chain due to the broad spectrum of supply chain. However, it is still necessary to have a model builder that declare the scope of supply chain model, reflecting the key real-world situation.[58]

After the analysis and synthesis of the supply chain model design in [58][59][60], Figure 10 illustrates the proposed model of supply chain in this thesis, which defines the structure vertically and horizontally. The horizontal layers refer to the amounts of different roles with different functionality through the supply chain. The length of horizontal layers depends on the tiers of supply chain, which may be long because of plentiful tiers, or short due to less tiers. The vertical layers refer to the number of different entities within each horizontal layer. In practices, one specific model structure depends on vertical and horizontal dimension of the supply chain.

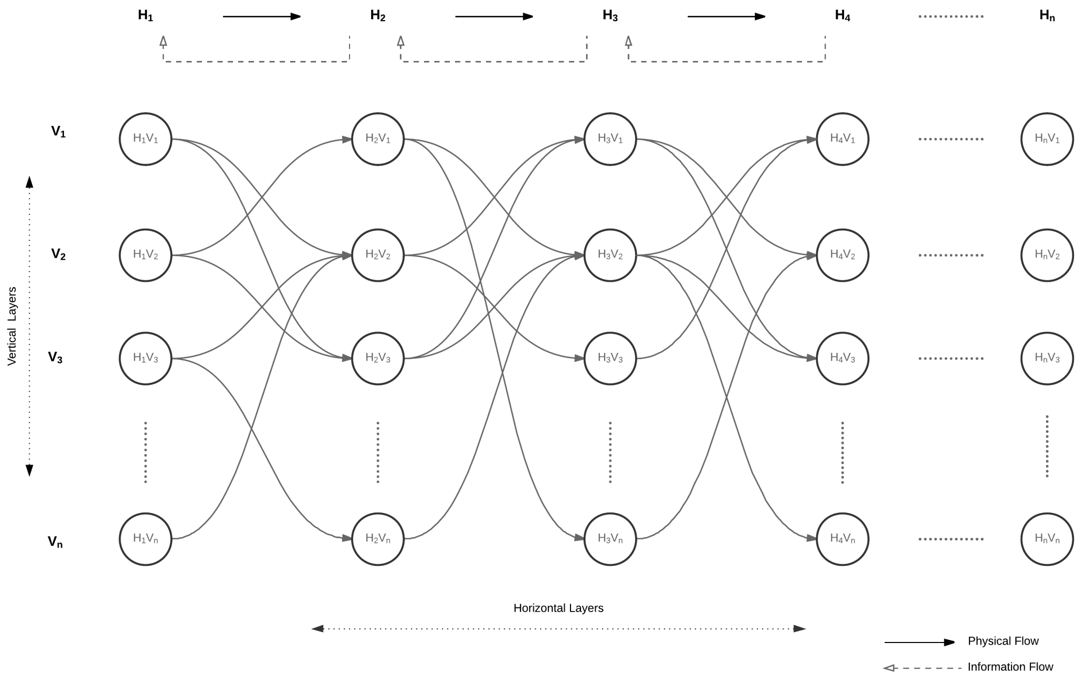


Figure 10. *Industrial supply chain modeling*

The roles in horizontal layers can be described as H_n: H is short for horizontal layers, n stands for the sequential order of position within supply chain. The larger number n is, the closer to the end of downstream the role is. Inversely, if a role has a small number of n, we can deduce its position of supply chain is in the upstream.

The entities in vertical layers can be represented as V_n: V is short for vertical layers. Compared to n in H_n, here n in V_n does not indicate the order anymore, but only show the different entities with no-repeat number.

In this two-dimensional model, any entities with different role can be described as $H_nV_{n,}$ which explicitly represents the entity with role H_n and in the vertical layer $V_{n.}$

Supply Chain Flows
In this thesis, the model contains two basic interaction flows traveling across the entire supply chain model: physical flow and information flow. There may be more flows transact between participants (e.g. financial flow, human resource flow etc.), which will not be addressed here.

The physical flow represents the flow of products from upstream of participants to downstream across the supply chain. The information flow accompanies the physical flow of products, which is perceived as an essential part of the physical products processing tasks (e.g. purchase order, product information, payment information, insurance, shipment tracking)[61].

Physical Flow
Compared to other marketplaces, the supply chain has a single direction of physical flow, from origin to endpoint, without being repeatedly traded back-and-forth between the participants and transact with other competitors vertically.

Any physical flows can be represented as following formula:

$$H_aV_b \rightarrow H_cV_d \ (a<c)$$

The sign → shows the information flows direction. Besides, the condition(a<c) is suitable for most use cases, because the assets usually transfer from upstream roles to downstream roles across the supply chain. However, some edge cases of physical flow can still break this condition. For instance, the refund assets can be inverse direction to normal physical flow.

Besides, the condition can even be narrow down more for most situation. We can assume that transferring parts in supply chain from one role to next role horizontally, without jumping to next layers or even further:

$$H_aV_b \rightarrow H_cV_d \ (a+1=c)$$

All those assumptions are depending on hypnosis of simply supply chain network. In reality, supply chain should be more complex so that those conditions could be mixed or even more condition exist.

Information Flow
The information flow is usually a backward flow in contrast to physical flow. There are usually two types of information flows defined in [62]:

1. Information flow directly linked to the physical flow: used to produce the product or service (e.g. order information, shipping information, quantity information, etc.)

2. Information flow that indirectly related to the physical flow: information about customer, market, feedback (e.g. market forecast, customer demands, etc.)

However, in this thesis, we only consider about only the first information flow which is related to physical flow.

Any information flows can be abstracted as formula below:

$$\boldsymbol{H_cV_d \rightsquigarrow H_aV_b\ (a<c)}$$

In this formula, the sign $\rightsquigarrow$ indicates the information flows direction. In the given condition, we deduct the information flow mainly comes from downstream role to upstream role, even though information exchange is bidirectional at most of time. However, the information, which is needed for producing product (e.g. purchase order), is inverse to physical flow.

Model Extension

In reality, each role can expand to more sub-tier participants. For example, in Figure 11 , H_n can break down to more layers depending on how complex the H_n are[63].

- Tier one companies($H_{n1}V_n$) supply parts or system directly to H_{n+1}, which is major suppliers of parts to H_{n+1}
- Tier two role($H_{n2}V_n$) are the key suppliers to tier one suppliers, without directly supplying a product to H_{n+1} usually, but sometime still possible to supply with higher tier role($H_{n1}V_{n}$)
- Tier m companies($H_{nm}V_n$) mainly supply to tier *m-1* Suppliers($H_{nm-1}V_{n}$)

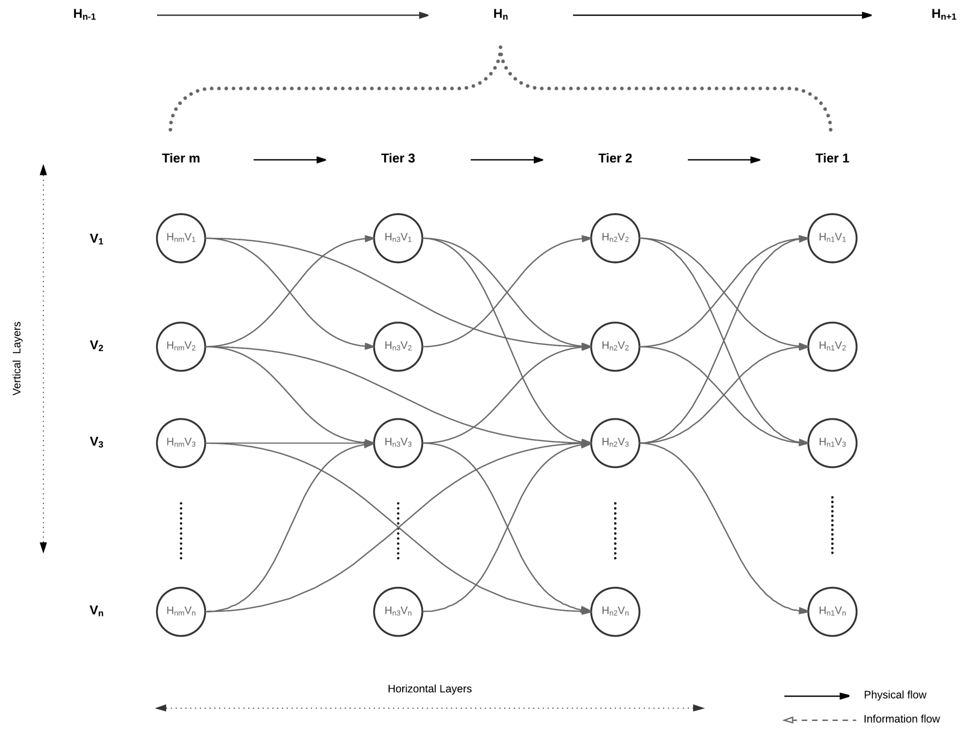


Figure 11. *Supply chain Network Structure*

Creating Supply Chain Model

Based on in-depth understanding of specific supply chain environment, creating an accurate and comprehensive model for supply chain model is crucial that to provide solid foundation for applying to blockchain model in the further chapter. Figure 12 shows a unified process of creating and updating a supply chain model mentioned before.

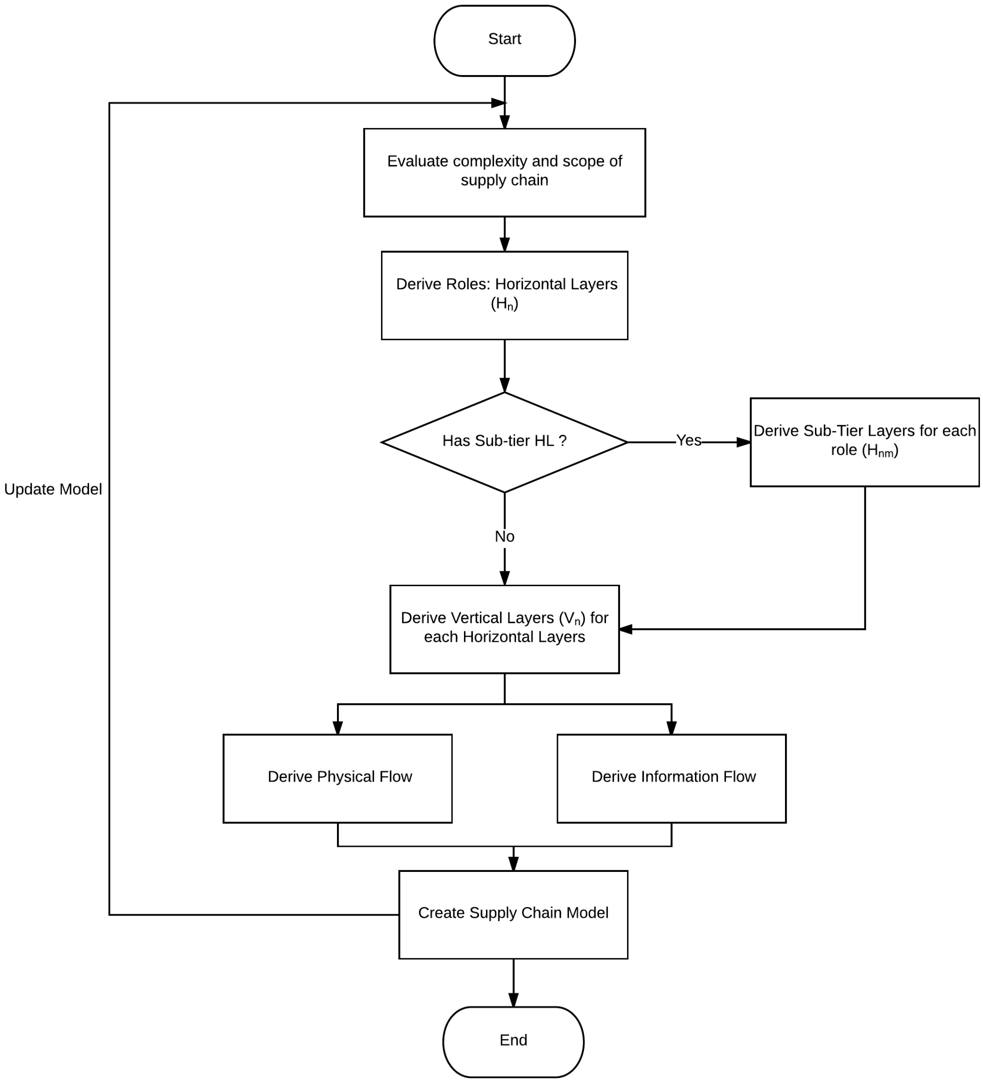

Start
Evaluate complexity and scope of supply chain
Derive Roles: Horizontal Layers (H_n)
Has Sub-tier HL ?
Yes
Derive Sub-Tier Layers for each role (H_{nm})
No
Update Model
Derive Vertical Layers (V_n) for each Horizontal Layers
Derive Physical Flow
Derive Information Flow
Create Supply Chain Model
End

Figure 12. *Flowchart of creating supply chain model*

The workflow of creating supply chain model following steps:

1. Evaluate complexity and scope: collect and analysis specific case of supply chain help to derive the important parameters of creating supply chain model later
2. Derive Roles (Horizontal Layers: H_n): figure out different roles in the horizontal layers: total amounts of layers and the order of them (from upstream to downstream): $H_{1,}$ $H_{2,}$ $H_{3,}$ …… H_n
3. Make sure if there are sub tiers for each horizontal layer
4. If the answer of step 3 is "YES", starting to derive sub tiers for each role in the horizontal layers: $H_{n1,}$ $H_{n2,}$ $H_{n3,}$ …… $H_{nm.}$ Then goes to step 6.
5. If the answer of step 3 is "NO", continue to conduct vertical layers(V_n) of all horizontal layers (from H_1 to H_n): $H_1V_{n,}$ H_2V_n ……. H_nV_n
6. Derive physical flow ($H_aV_b \rightarrow H_cV_d$) according to physical exchange between different entities through supply chain network
7. Derive information flow ($H_cV_d \dashrightarrow H_aV_b$) according to information exchange between different entities across supply chain network
8. After finishing all the seven steps above, a basic supply chain model is created
9. If any updates are possible, restart the process from step 1

3.3.2 Data model

In order to ensure traceability, there are two main categories of data in the supply chain: one is related to asset, the other is transaction data.

Figure 13 shows the designed structure of the asset data model.

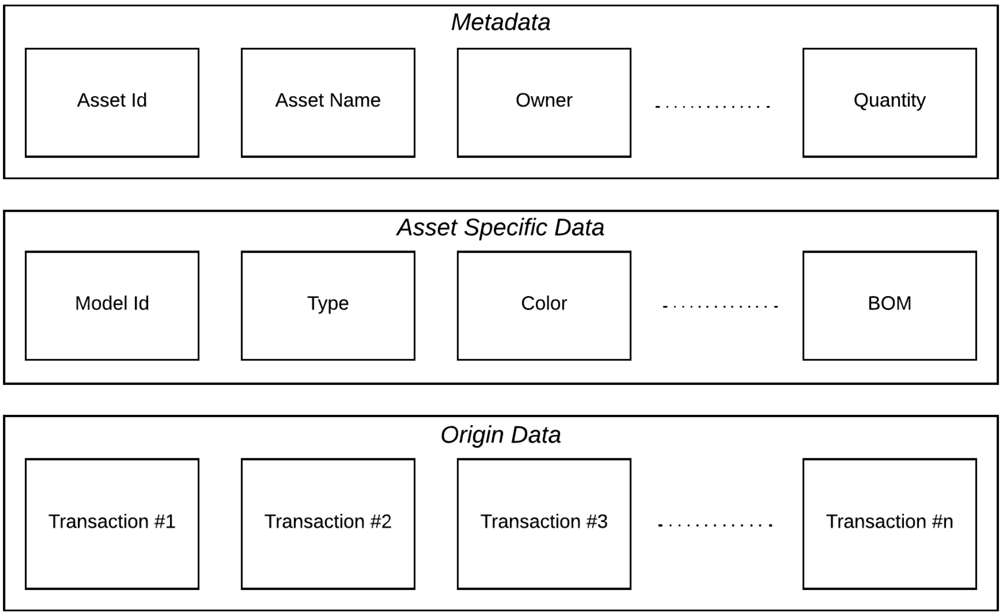


Figure 13. *Asset Data Model*

All the properties in the data model are following:

1. Metadata: The common information of item. (i.e. Asset id, Asset name, Owner, quantity, etc.)

2. Asset Specific Data: Detailed information about certain asset (i.e. Model id, type, color, BOM, etc.)
3. Origin Data(trace): all the history data related to transactions, which is the core of traceability

Figure 14 present the transaction data model in a nutshell:

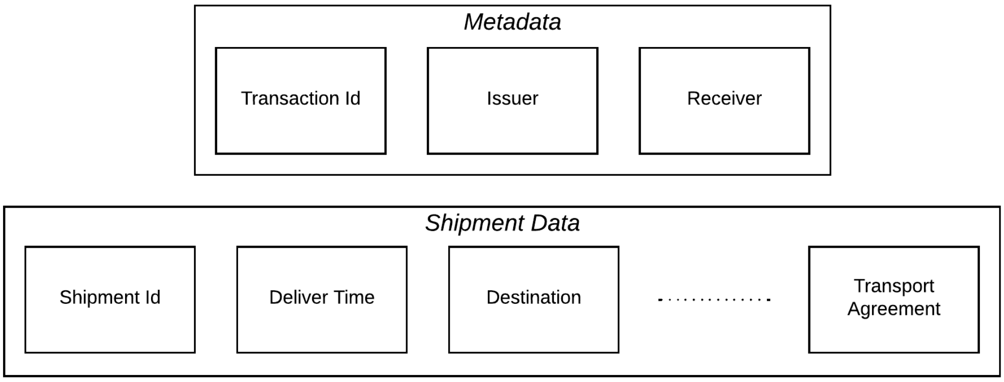


Figure 14. *Transaction Data Model*

There are two main blocks in transaction data model:

1. Metadata: The common information of transaction. (i.e. Transaction id, issuer, receiver, etc.)
2. Shipment Data: All information about shipment (i.e. Shipment id, Deliver time, destination, transport agreement, etc.)

3.3.3 Model Mapping

There are four main types of resource in business network model in Hyperledger Composer: Participant, Asset, Transaction, Event. Table 2 shows the definition and example of model in the blockchain business network:

Table 2. *Definition and Example of Model in blockchain business network*

	Definition	Example
Assets	tangible or intangible goods, services, or property	houses and listings
Participants	Actor with a certain role	buyers and homeowners
Transactions	Interaction between participants and assets	buying or selling houses, and creating and closing listings
Event	Message emitted by transaction	“Jack brought house #1 at 11.11.2017”

Participants are actors with specific role in a business network, which may own assets and submit transactions. Participant has an identifier (like ID) and other properties. The properties can be mandatory or optional depending on the requirement of the data integrity. Identity documents, such as passport, fingerprints, driving license, are required to validate before interacting with the business network.

Asset is the property (tangible goods or intangible services) that can be owned by participant. Like participant, asset also must have a unique identifier and contain defined properties. Besides, assets may be referred to other participants and assets.

Transactions are the mechanism of interaction between participants and assets. Participant can submit a transaction to change one or more properties of assets or create or delete assets.

Events are emitted by transaction processor function to indicate external system that important thing has just happened. Any applications or existing system can submit the event and receive them in the future.

Figure 15 shows how to gap between supply chain model to business network model in blockchain with mapping tool. Besides, it also shows the interaction between models in business network.

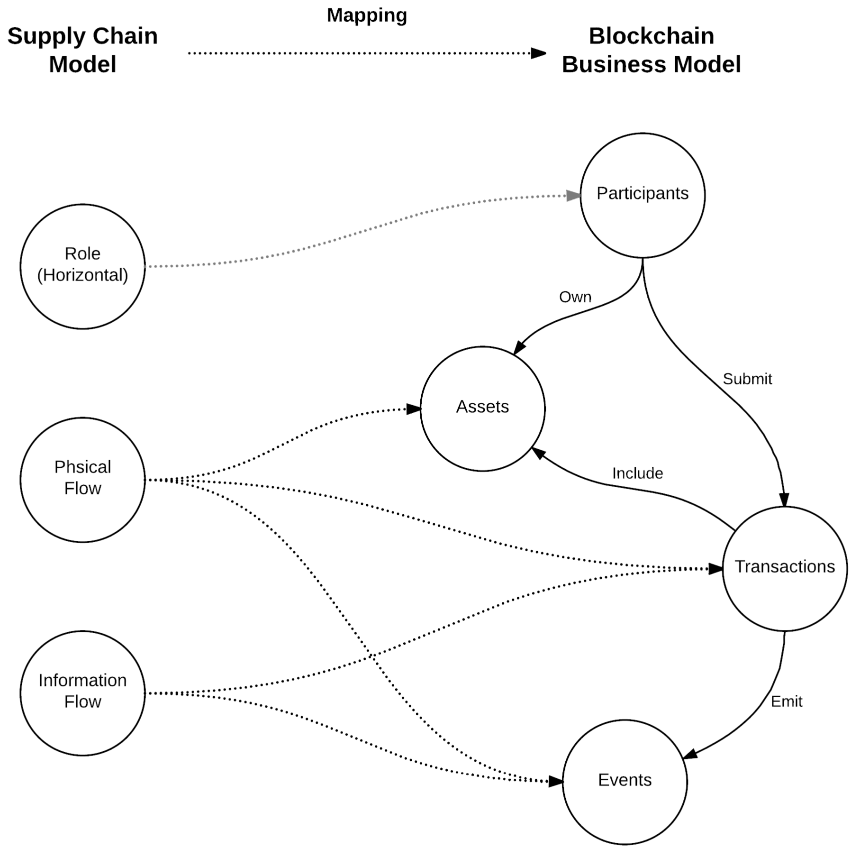


Figure 15. *Model mapping from supply chain model to blockchain network model*

From supply chain model mapping to blockchain model, horizontal role can be map to different participants with different functions in the business network. Assets, transactions and events can be deducted explicitly and implicitly from physical flow and information flow.

3.3.4 Blockchain Network Model

Figure 16 explains the network model in Hyperledger blockchain, which shows a transaction lifecycle where peers authenticate, propose, endorse, order, validate and commit transaction in the blockchain network.

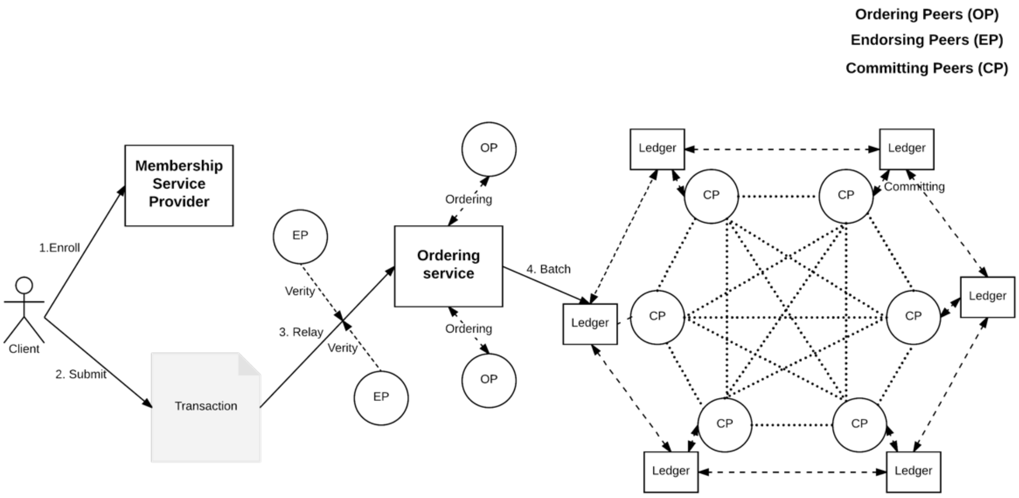


__Figure 16.__ Network model [43]

Considering the scalability and distributed computation, the network separates transaction endorsement from consensus[64]. Thus, there are three different types of peers in the network:

- Endorsing Peers: grant or deny endorsement of transaction proposal by validating with endorsement policy
- Ordering Peers: approve endorsed transactions and batch them in to one block in an order
- Committing Peers: commit endorsed and ordered transaction into common ledger

The confidential state can be only seen by endorsing peers in the network. Different identities of peers can be arranged to different participants in the network depending on policy agreed by all the participants in the blockchain network.

While blockchain network is distributed network, in most of case in the real use, network and rules might be set up by one trust party in the beginning. But since the access rule should be agreed by each node in the network and blockchain is an immutable database, if administrator or regulator try to change the rule or do something harmful, every peer in the network could see the record of each action.

3.3.5 Transaction Flow Model

Last section introduces the network model and basic interaction flow. This section will focus on the transaction flow of a lifecycle of commodity exchange. This scenario has two clients: S_n and M_n, which S_n submit a transaction to M_n for transferring a commodity ownership. Each client has its own represented peer on the network, which allow them to submit transactions and interact with the shared ledger. Figure 17 illustrates a transaction flow in the blockchain network.

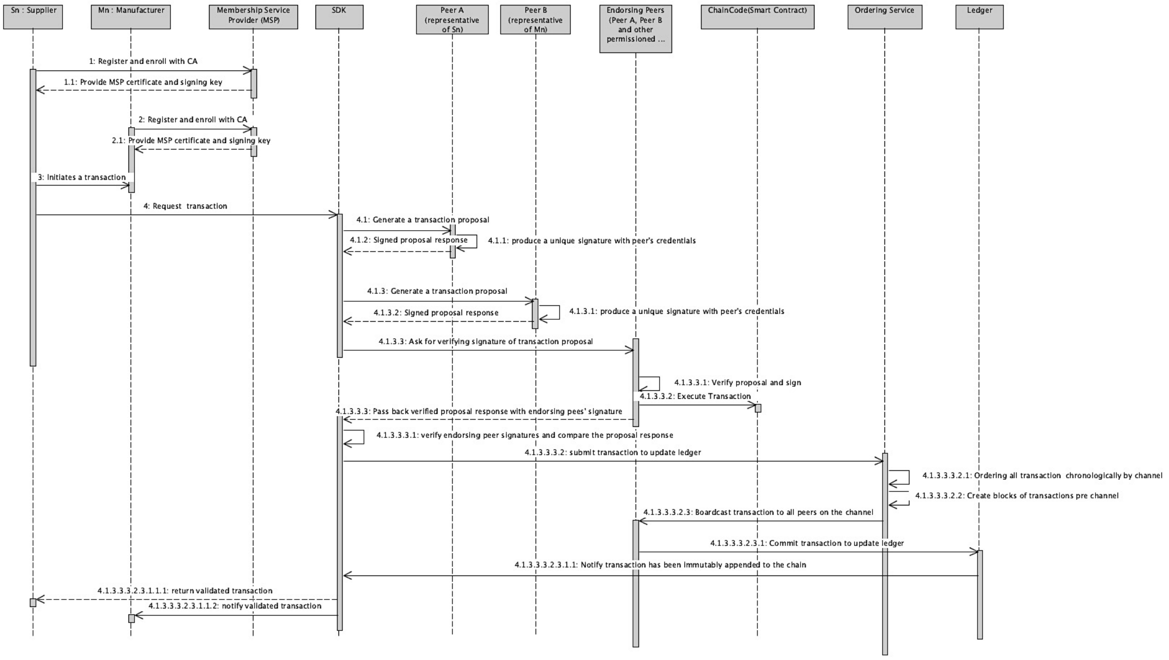

Sn : Supplier
Mn : Manufacturer
Membership Service Provider (MSP)
SDK
Peer A (representative of Sn)
Peer B (representative of Mn)
Endorsing Peers (Peer A, Peer B and other permissioned ...
ChainCode(Smart Contract)
Ordering Service
Ledger
1: Register and enroll with CA
1.1: Provide MSP certificate and signing key
2: Register and enroll with CA
2.1: Provide MSP certificate and signing key
3: Initiates a transaction
4: Request transaction
4.1: Generate a transaction proposal
4.1.2: Signed proposal response
4.1.1: produce a unique signature with peer's credentials
4.1.3: Generate a transaction proposal
4.1.3.2: Signed proposal response
4.1.3.1: produce a unique signature with peer's credentials
4.1.3.3: Ask for verifying signature of transaction proposal
4.1.3.3.1: Verify proposal and sign
4.1.3.3.2: Execute Transaction
4.1.3.3.3: Pass back verified proposal response with endorsing pees' signature
4.1.3.3.3.1: verify endorsing peer signatures and compare the proposal response
4.1.3.3.3.2: submit transaction to update ledger
4.1.3.3.3.2.1: Ordering all transaction chronologically by channel
4.1.3.3.3.2.2: Create blocks of transactions pre channel
4.1.3.3.3.2.3: Boardcast transaction to all peers on the channel
4.1.3.3.3.2.3.1: Commit transaction to update ledger
4.1.3.3.3.2.3.1.1: Notify transaction has been immutably appended to the chain
4.1.3.3.3.2.3.1.1.1: return validated transaction
4.1.3.3.3.2.3.1.1.2: notify validated transaction

Figure 17. *Sequence diagram of transaction flow*

The detailed explanation of each steps is following:

1. Client S_n and Client M_n respectively send registration and enrollment to Membership Service Provider (MSP) with their certificate authority (CA) and wait for MSP issue them confidential, which includes certificate and private key. Those confidential is used to authenticate themselves to the blockchain network in the future.

2. Client S_n initiates a transaction to Client M_n. This initiation is referred to Peer A and Peer B, who are individually representative of Client Sn and Client Mn. According to endorsement policy, transaction should be endorsed by both peers, thus this deal transfers to Peer A and Peer B. Then SDK generate a transaction proposal to each peer. Afterwards, Peer A and Peer B sign proposal with their own signature, which is derived from cryptographic credentials.

3. The signed proposal send to endorsing peers for verification. The endorsing peers will validate the transactions and execute the function, then passing back the proposal response to the SDK.

4. SDK verify the proposal response and endorsing peer signature. Then, SDK checks if the transaction fulfils endorsement policy. Afterwards, it submits the transaction to Ordering service.

5. Order Service receives all the pending transactions in the network and order them chronologically, then creating blocks of transaction.

6. Transaction encapsulated in a block pass to committing peers who will append the transaction in the blockchain. The ledger is finally updated and event is emitted to notify Client S_n and Client M_n that the transaction has been successfully proceeded.

3.3.6 3D Model

Figure 18 shows a possibility of the space extension of the supply chain model. With one more dimension, the 3D model could define different layers of model where some of the nodes are overlapped.

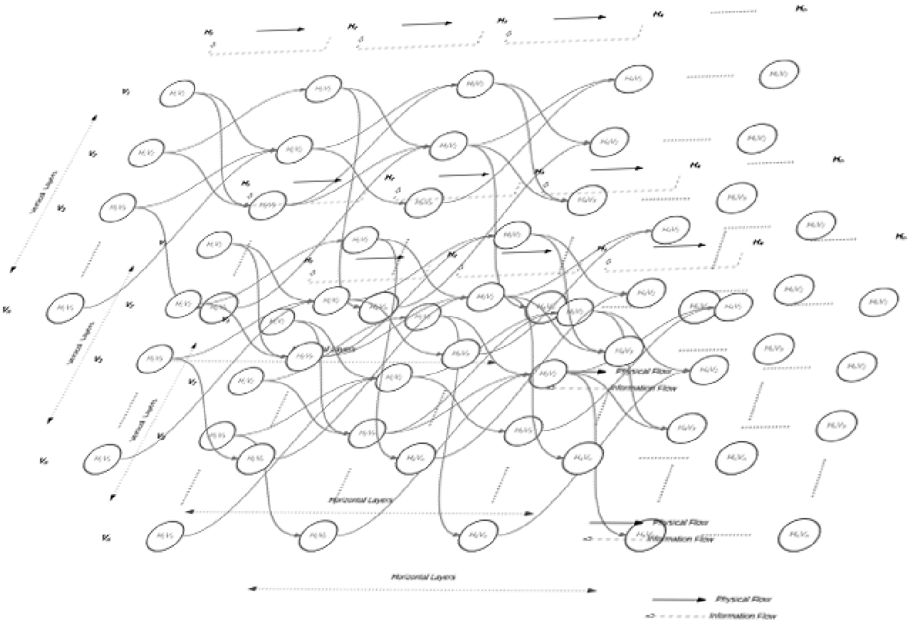


Figure 18. *3D model with 3rd dimension extension*

This 3D model is designed for emerging different supply chain to use more than one blockchain system. In reality, different supply chain may adopt different blockchain system to interact with each other. This 3D model could provide an interoperability between different blockchain systems, which allow them to communicate with each other.

4. IMPLEMENTATION

This chapter presents the implementation of the approach of creating supply chain model in blockchain which described in chapter three. Implementation includes providing a test scenario for the model.

4.1 Use Case

4.1.1 Roles

In this use case, we define mainly five different kind of roles in horizontal layers in order: supplier, manufacturer, distributor, retailer and customer. For supplier layer, it has 2 tiers of subcomponents: tier 1 and tier 2.

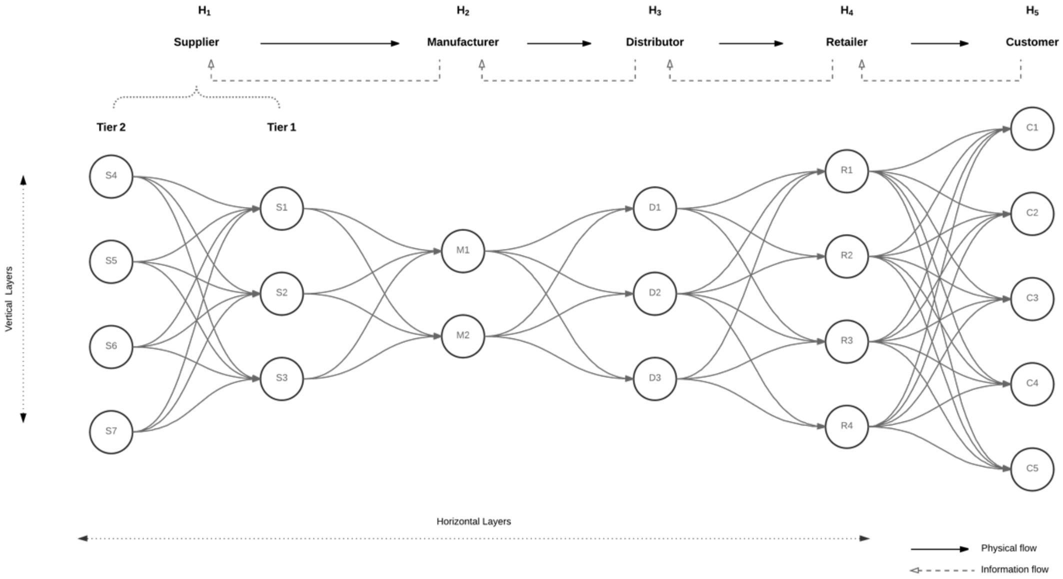


Figure 19. *A supply chain model for use case*

Different roles in the chain:

- Supplier (S1, S2, S3, S4, S5, S6, S7): Role with supplying a particular part to Manufacturer
- Manufacturer (M1, M2): Role to take over the parts supplied by suppliers, assemble them and pass away to next role: distributor
- Distributor (D1, D2, D3): Role with authority to allot or deal out or apportion, then distributing to different retailers in different physical location
- Retailer (R1, R2, R3, R4): Role to sell commodity directly to consumer

- Customer: (C1, C2, C3, C4, C5): Role of purchasing goods from retailer and the very end of supply chain

Table 3 clearly shows the conversion of notation between model mentioned in 3.3.1 and new id notation. For example: M1 is as same as H_2V_1, which indicates its role is in the 2nd place close to upstream of supply chain.

	H_1 (Supplier)		H_2 (Manufac-turer)	H_3(Distributor)	H_4(Retailer)	H_5(Customer)
	H_{1-2} (Tier 2)	H_{1-1} (Tier1)				
V_1	S4	S1	M1	D1	R1	C1
V_2	S5	S2	M2	D2	R2	C2
V_3	S6	S3		D3	R3	C3
V_4	S7				R4	C4
V_5						C5

Table 3. *Notation conversion to id*

4.1.2 Supply Chain Flows

After the roles is confirmed in the previous section, this section will discuss about interaction flow between each role. Figure 20 depicts the user activity diagram of supply chain flows. The physical line is presented with solid arrow line, while information line is draw by dotted line.

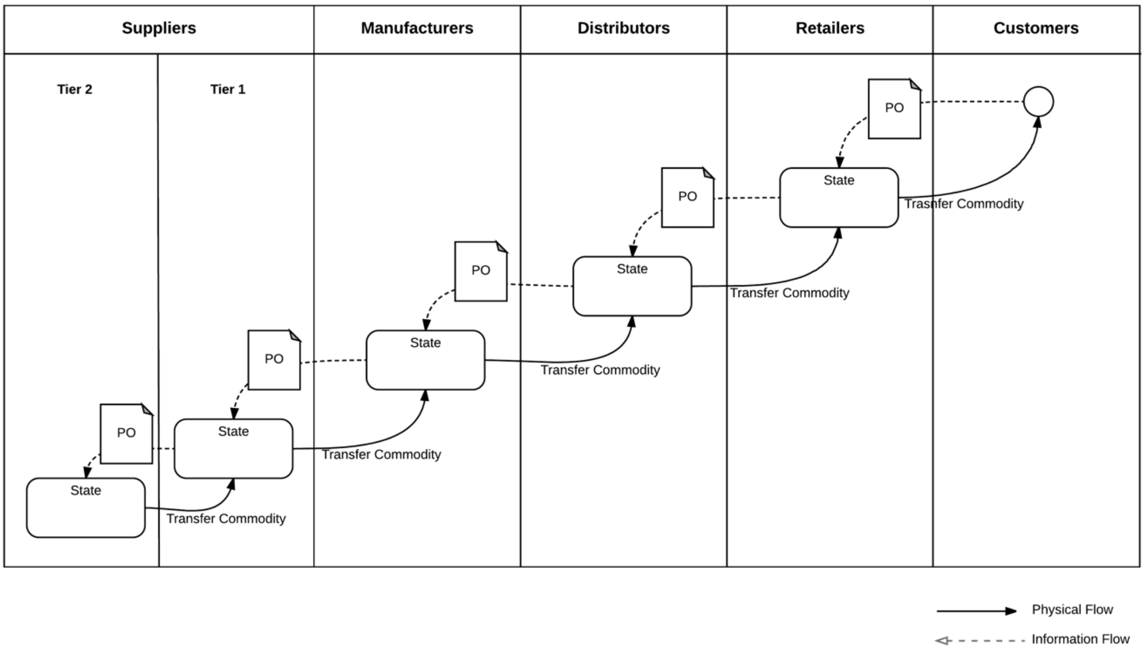


Figure 20. *Activity diagram of supply chain flows*

In real industry, supply chain flow can contain various type of physical flows and information flows. Physical flows can vary from shipment to return of goods, while information flows differ from sales orders, purchase orders(PO) to invoices. In this use case, we only illustrate one of the most typical case for each flow.

For information flows, the most frequent transaction is sending purchase order directly from upstream role to downstream participant:

$$\textbf{PO: } H_cV_d \dashrightarrow H_aV_b \ (a=c+1)$$

For this use case, customer send PO to online retailer. Retailer sends PO to distributor, distributor sends PO to Manufacturer. Manufacturer initiates PO to Tier 1 supplier. Tier 1 supplier send PO to Tier 2 supplier.

After PO is initiated, the shipment will transfer in verse direction:

$$H_aV_b \rightarrow H_cV_d \ (a+1=c)$$

For physical flows, suppliers supply components or raw materials to manufacturer, then manufacture assemble the parts to finished products. Afterwards, distributors receive finished products from the plants and ship to retailers. Finally, customers purchase products from retailers.

4.2 Validation Scenario

For validation purposes, this thesis includes a simple but practical scenario of phone industry supply chain. This scenario might not cover all of the details or specification in the real phone industry, but the process from upstream to downstream is an epitome of modern industrial supply chain. Figure 21 shows the physical flows and information flows of supply chain of phone industry.

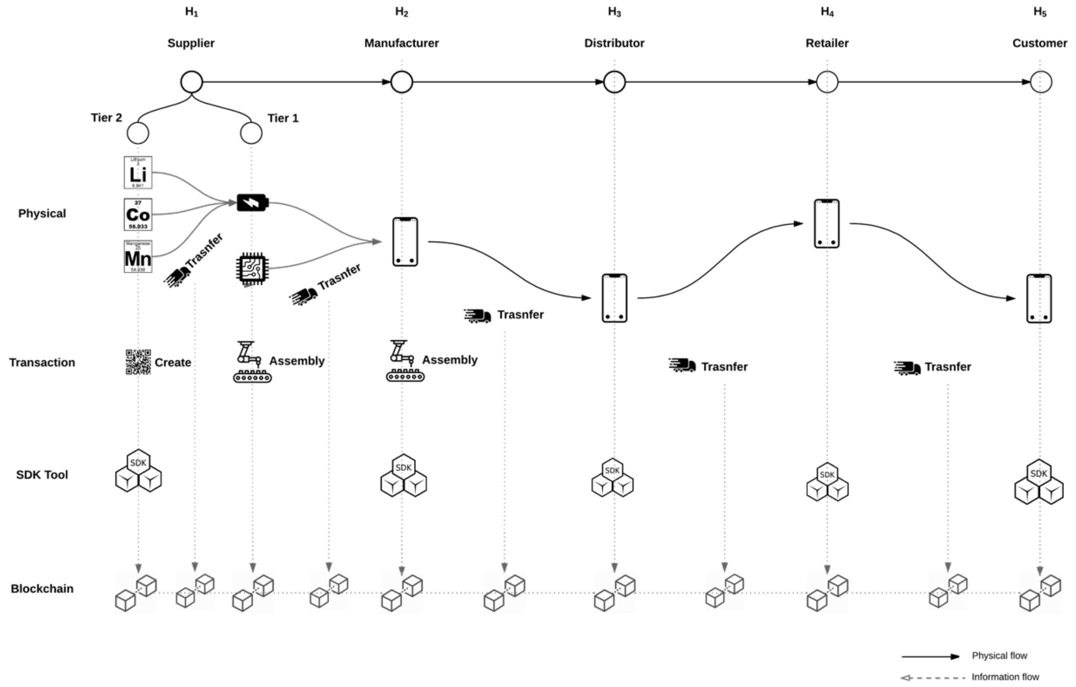


Figure 21. *A Scenario of phone industry recording transactions on blockchain*

As it mentioned in 4.1.1, there are five different kinds of participants: Suppliers, manufacturer, distributor, retailer, customer. For role supplier and manufacturer, is has transaction type of initiating PO (purchase order), assembling parts and transferring commodities. However, distributor and retailer are only responsible for initiate PO and transferring the commodities.

As it stated in 4.1.2, there are two main type of asset: PO (purchase order) and Commodity. Purchase order can be initiate from Manufacturer to Supplier. PO is created when one participant issue to another by transaction function (initiate PO). Commodity is created either by top upstream supplier (raw material supplier) or downstream suppliers and manufacturer by assembling parts.

All the possible transaction in the process from source to end user can be concluded with three main type: PO initiation, part assembly, transfer commodity transferring. Once those transaction happened, the most valuable data is generated and needed be recorded in the blockchain for further use.

4.3 Development of business network definition

This section will introduce how to use Hyperledger Composer, an open source development toolset and framework, to develop business network definition, which is a prerequisite of deploying on blockchain runtimes in the next section.

Figure 22 shows how business network definition is developed with three kinds of files and deployed finally on blockchain runtimes. The whole project can be found in GitHub[11].

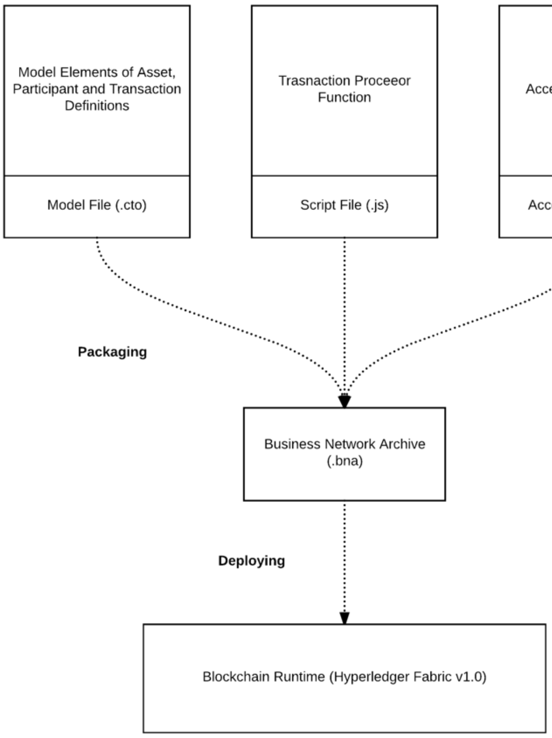

Model Elements of Asset, Participant and Transaction Definitions
Model File (.cto)
Trasnaction Proceeor Function
Script File (.js)
Access Control Rules
Access Control (.acl)
Packaging
Business Network Archive (.bna)
Deploying
Blockchain Runtime (Hyperledger Fabric v1.0)

Figure 22. *Development and Deployment of business network on blockchain runtime*

Model file defines the structure and relationship between model elements: asset, participant, and transaction. This file can be created by business analysts in real industry.

Script file contains transaction processor functions, which will run on blockchain runtime and manipulate the world state of blockchain by accessing to the asset registries.

Access control file contains a set of rules that restrict the rights of the different participants in the business network.

4.3.1 Modeling

Modeling language is an object-oriented language used to define the model of resources (asset, participant, transaction) in a business network definition. It not only defines data structure of each model but also point out the reference between different models. Program 1 shows an example of modeling commodity with modeling language.

[11] https://github.com/aprilsnows/hyperledger-composer-supply-chain-network

```
/**
 * Asset: Commodity
 */
namespace org.hcsc.network

asset Commodity identified by tradingSymbol {
    o String tradingSymbol
    o String name
    o String description optional
    o Double quantity
    o Double unitPrice optional
    o Double totalPrice optional
    o Trace[] trace
    --> Commodity[] children optional
    --> PO purchaseOrder optional
    --> Trader owner optional
    --> Trader issuer optional
}
```

Program 1. *Snippet of the modeling file*

The snippet starts with comments of explanation and namespace that declare all the resource are implicitly in. Then it continues with data type and field name. Some fields are not compulsory with *optional* flag. The filed with '`-->`' indicates the referenced relationship to other model types.

4.3.2 Transaction Processor Functions

Transaction processor function is encompassed in the script file that the transactions defined in the Business Network Definition's model files. Transaction processor functions are invoked to change the state of resource in the associated resource registry.

Program 2 illustrate a transaction processor function that updates the state of commodity when it is trade and transferred from one participant to another.

```
/**
 * Track the trade of a commodity from one trader to another
 * @param {org.hcsc.network.TransferCommodity} trade - the trade to be pro-
cessed
 * @transaction
 */
function transferCommodity (trade) {

    var NS = 'org.hcsc.network';
    var factory = getFactory();

var me = getCurrentParticipant();

    trade.commodity.issuer = me;
    trade.commodity.owner = trade.newOwner;
    trade.commodity.purchaseOrder = trade.purchaseOrder;

    var newTrace = factory.newConcept(NS, 'Trace');
    newTrace.timestamp = new Date();
    newTrace.location = trade.shipperLocation;
    newTrace.company = me;
    trade.commodity.trace.push(newTrace);


      return getAssetRegistry('org.hcsc.network.Commodity')
            .then(function (assetRegistry) {
                  return assetRegistry.update(trade.commodity);
      });
 }
```

Program 2. *Script File*

It is important to highlight that the field trace is updated and pushed to an array for backwards traceability in the future.

4.3.3 Access Control

Access control language(ACL) declare access control rule of interaction between asset, participant and transaction. After defining ACL rules that restrict which users/roles are permitted to manipulate (create, read, update or delete) resources in the business network.

There are two different type of access control: business access control and network access control. The former one defines the rules of access control for resource within a business network while the latter one declares for network administrative changes (e.g. update business network).

Program 3 shows an example of network access control for system access control.

```
/**
 * New access control file
 */

rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system
resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
```

Program 3. *A snippet of network access control*

As the snippet illustrated, the system namespace is used to implicitly reference to the resource in a business network, which appear in participant and resource fields.

In the rule definition, the field participant indicated the entity who submit transaction to change the state in blockchain. In this case, network admin is the entity.

Operation define which actions are allowed. Four kinds of actions are supported: CREATE, READ, UPDATE, and DELETE. In this case, all these four actions are supported.

Action identifies the action of the rule. In the example, it is '*allow'* means it grant the access to the certain action. On the contrary, it could be '*deny'* which declare the action is forbidden.

Program 4 shows a snippet of access control within business network. In this matter, the rule grants all participants to transfer the commodity which is own by themselves.

```
rule ConditionRuleWithTransaction {
    description:"Allow all participants to transfer its own commodity only by
TransferCommodity"
    participant(m): "org.hcsc.network.*"
    operation: UPDATE
    resource(v): "org.hcsc.network.*"
    transaction(tx): "org.hcsc.network.TransferCommodity"
    condition: (v.owner.getIdentifier() == m.getIdentifier())
    action: ALLOW
}
```

Program 4. *A snippet of business access control*

It deserves attention for two new fields: transaction and condition.

The field transaction specifies the prerequisite of operation on the certain resource. The rule will not allow access of manipulating resource when participant attempt to modify without submitting the specific transaction.

The field condition is a Boolean conditional equation to check coming from JavaScript if(…) expression. In this matter, only when the owner of the resource is equal to the identifier of the participant, the action is allowed.

4.4 Deployment on Blockchain

After defining business network definition, it is ready to deploy it on the blockchain runtime. Figure 23 illustrate the progress of setting up blockchain runtime and deploying business network definition on runtime.

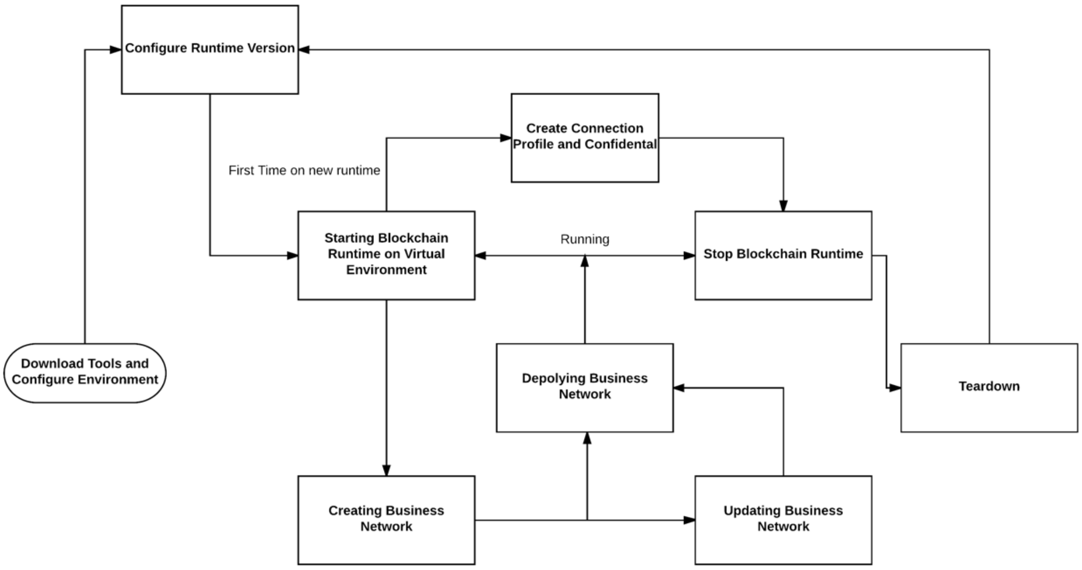

Configure Runtime Version
Create Connection Profile and Confidental
First Time on new runtime
Starting Blockchain Runtime on Virtual Environment
Running
Stop Blockchain Runtime
Download Tools and Configure Environment
Depolying Business Network
Teardown
Creating Business Network
Updating Business Network

Figure 23. *Diagram of deployment on Blockchain*

In the initial phase of deployment, who should be network administer and its access rule should be designed and agreed by all participant on the network. However, each deployment and modification will be recorded in the immutable blockchain database so even the deployment is centralized but every action is monitored and traceable.

It is also important to highlight that updating business network definition on runtime during running state is feasible. However, the blockchain will record the updates of business network with executor and time as same as first time to deploy business network on runtime.

Table 4 shows five images running on Docker[12] with image, ports, name and size. First one is local composer playground, which is a web sandbox allows to deploy, edit and test business network definitions. Secondly, fabric-peer is for peers endorsing and committing. Thirdly, fabric-couchdb is a key value database used to store its state. Forth is the ordering service that takes responsibility of ordering the endorsed transaction. Last but not the least, fabric-ca is certificate authority that take over registration of identities and issuance of enrollment certificate.

Table 4. *Blockchain Runtime (Hyperledger Fabric) on Docker Images*

NO	CONTAINER ID	IMAGE	PORTS	NAMES	SIZE
1	e2aad55ef6dd	hyperledger/composer-playground	0.0.0.0:8080->8080/tcp	composer	6.55MB (virtual 284MB)
2	677a2dfc35fb	hyperledger/fabric-peer:x86_64-	0.0.0.0:7051->7051/tcp, 0.0.0.0:7053-	peer0.org1.example.com	3.01MB (virtual 185MB)

[12] https://www.docker.com/

		1.0.1	>7053/tcp		
3	f5d7a8263324	hyperledger/fabric-couchdb:x86_64-1.0.1	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp	couchdb	125kB (virtual 1.48GB)
4	48331ca78470	hyperledger/fabric-orderer:x86_64-1.0.1	0.0.0.0:7050->7050/tcp	orderer.example.com	94.3kB (virtual 179MB)
5	a382e964bbd3	hyperledger/fabric-ca:x86_64-1.0.1	0.0.0.0:7054->7054/tcp	ca.org1.example.com	37.1kB (virtual 238MB)

Figure 24 show a workflow of operation on blockchain from perspectives of company or organization. It is a simple process to submit transaction and update information in the blockchain after registering and enrolling in MSP (membership service provider).

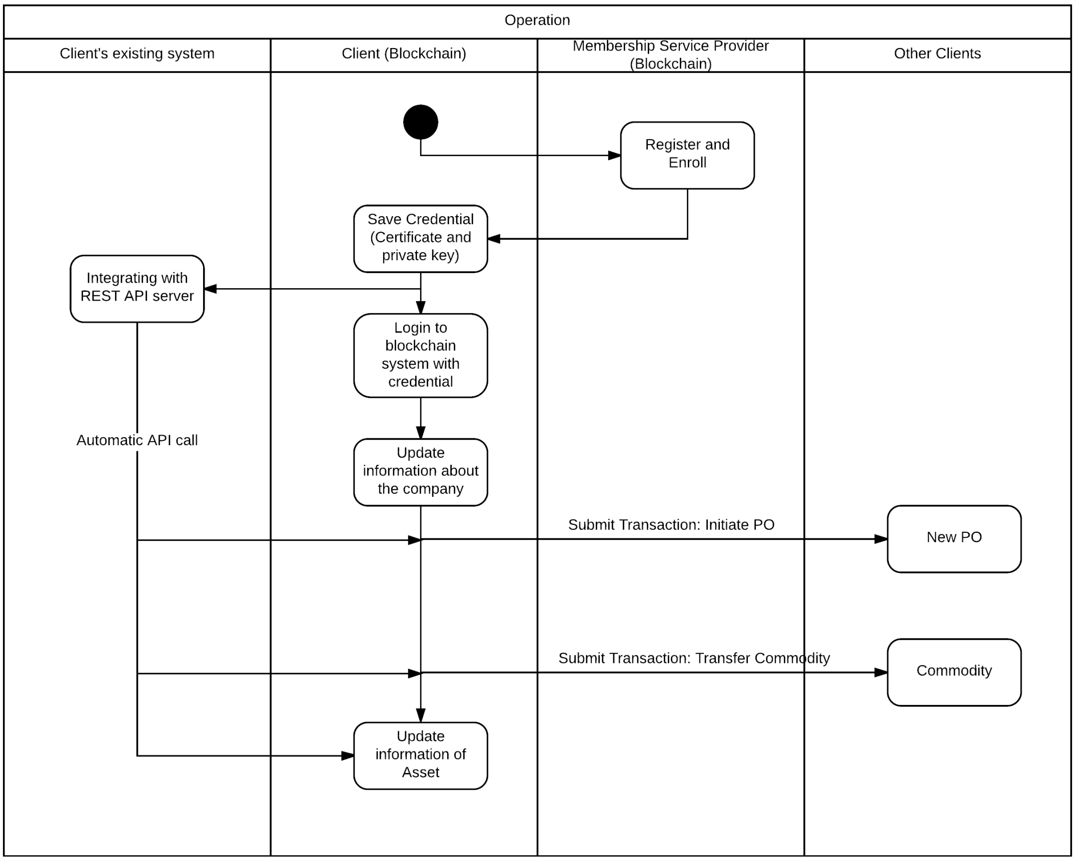

Operation
Client's existing system
Client (Blockchain)
Membership Service Provider (Blockchain)
Other Clients
Register and Enroll
Save Credential (Certificate and private key)
Integrating with REST API server
Login to blockchain system with credential
Automatic API call
Update information about the company
Submit Transaction: Initiate PO
New PO
Submit Transaction: Transfer Commodity
Commodity
Update information of Asset

Figure 24. *Activity diagram for operation from one company's point of view*

The integration with existing system is an optional step, however it can benefit from automatic API call through REST API server to execute all the possible transaction without human interference.

4.5 Integration Interface with Existing Systems

Blockchain network can be integrated with existing business systems (e.g. ERP system) or other application by using a Loopback[13] API. With integration, it is feasible to pull data from existing systems and convert it to assets or participants in a blockchain business network.

Hyperledger Composer offer a tool to generate Node.js[14] server that exposes a business network as a REST API. The Node.js server is using LoopBack framework, which is used to generate an Open API[15], described by a Swagger[16] document. Program 5 is a snippet from terminal showing the progress of generating REST server with a few steps of setting.

```
Composer-rest-server

? Enter the name of the business network card to use: admin@hyperledger-
composer-supply-chain-network
? Specify if you want namespaces in the generated REST API: always use
namespaces
? Specify if you want to enable authentication for the REST API using Pass-
port: No
? Specify if you want to enable event publication over WebSockets: Yes
? Specify if you want to enable TLS security for the REST API: No

To restart the REST server using the same options, issue the following com-
mand:
   composer-rest-server -c admin@hyperledger-composer-supply-chain-network -n
always -w true

Discovering types from business network definition ...
Discovered types from business network definition
Generating schemas for all types in business network definition ...
Generated schemas for all types in business network definition
Adding schemas for all types to Loopback ...
Added schemas for all types to Loopback
Web server listening at: http://localhost:3000
Browse your REST API at http://localhost:3000/explorer
```

Program 5. *Steps of generating a REST API server*

The REST server plays a role of gateway, subscribing to the events emitted from the deployed business network and publishing them to client applications. The communication protocol between server and client is WebSocket, enabling two-way real-time communication without requesting from client to server.

[13] https://loopback.io/

[14] https://nodejs.org/en/

[15] https://github.com/OAI/OpenAPI-Specification

[16] https://swagger.io/

For the security concern, the REST server is secured with HTTPS and TLS (Transport Layer Security), which allow all transmitted data between the REST server and clients to be encrypted.

Calling external REST service is also possible in transaction processor function, allowing to move complex computation off the blockchain.

In conclusion, Figure 25 summarize the possibilities of integration between blockchain and other service with REST API server.

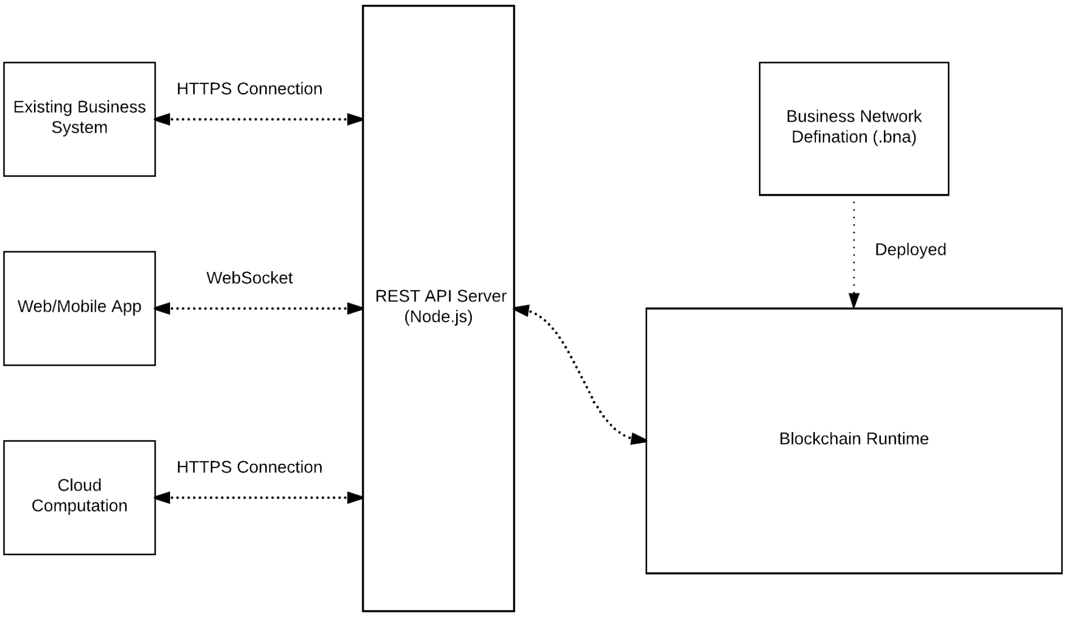


Figure 25. *Integrating with existing system, application and cloud services*

In conclusion, the blockchain system shows the possibility to securely integrate with other systems and service with different protocols.

5. RESULT ANALYSIS

As introduced in 4.2, the proposed validation scenario emphasizes an industrial use case. It is trivial to declare that this validation scenario shows promising results. In other words, the expected results show how well the blockchain technology help to increase transparency and traceability of supply chain while keep privacy and interoperability at the same time.

This chapter will focus on result analysis from different perspective: transparency, traceability, confidentiality and interoperability.

5.1 Transparency

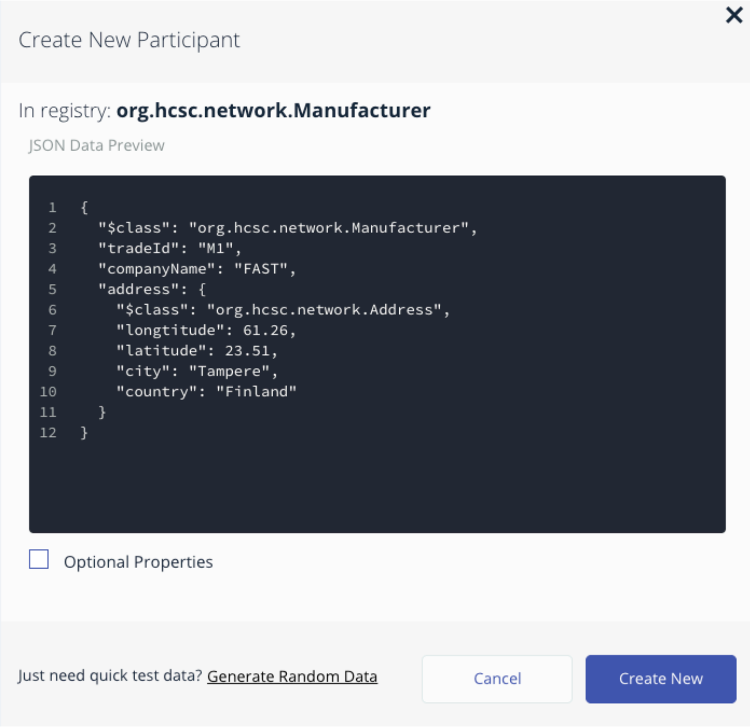
This section mainly discusses the transparency, which is visibility and operability of the system. Even through the ability to gain access to the information also depends on the access control and identity, the capability to record data and access data is unprecedentedly effortless and trustful compared to traditional database system.

This section will illustrate the transparency from three types of data registry: Participant, Asset, Transaction.

5.1.1 Participant Registry

In the proposed validation scenario, we have five roles/ participant model: supplier, manufacturer, distributor, retailer and customer, which is pre-defined in the model. However, the model can be updated later in the runtime if needed.

Figure 26 shows an example of creating an instance of manufacturer model by inputting all the necessary information. However, there are more detailed information can be input but not shown here for the simplicity and clearness of the example.



Create New Participant

In registry: **org.hcsc.network.Manufacturer**

JSON Data Preview

```
{
  "$class": "org.hcsc.network.Manufacturer",
  "tradeId": "M1",
  "companyName": "FAST",
  "address": {
    "$class": "org.hcsc.network.Address",
    "longtitude": 61.26,
    "latitude": 23.51,
    "city": "Tampere",
    "country": "Finland"
  }
}
```

☐ Optional Properties

Just need quick test data? Generate Random Data

Figure 26. *Create a new participant*

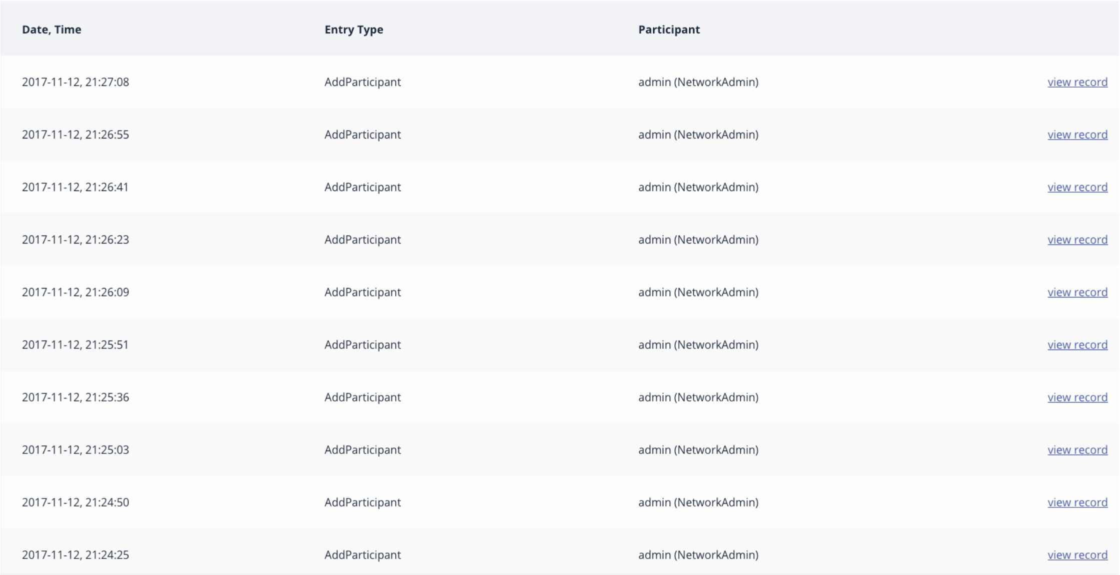
In this example, the class field indicates which role we are creating. *TradeId* is the identifier of this instance, which is M1. Other information related to this instance, like *companyName* and *address* is also needed to create a new instance of manufacturer.

After confirmation of creating new participant M1, Program 6 is a JSON format transaction data recording to blockchain with resource information, transaction Id and timestamp:

```
{
 "$class": "org.hyperledger.composer.system.AddParticipant",
 "resources": [
  {
   "$class": "org.hcsc.network.Manufacturer",
   "tradeId": "M1",
   "companyName": "FAST",
   "address": {
    "$class": "org.hcsc.network.Address",
    "longitude": 61.26,
    "latitude": 23.51,
    "city": "Tampere",
    "country": "Finland"
   }
  }
 ],
 "targetRegistry":                                                "re-
source:org.hyperledger.composer.system.ParticipantRegistry#org.hcsc.network.M
anufacturer",
 "transactionId": "69b0e489-3f03-4ae8-8fd9-13bf849c115f",
 "timestamp": "2017-11-12T19:23:06.052Z"
}
```

Program 6. *Record of the successfully created participant*

After creating all the participants of different roles for the validation scenario, all the creation of participants has been recorded with time, entry type, participant(executor) and details. See Figure 27:



Date, Time	Entry Type	Participant	
2017-11-12, 21:27:08	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:26:55	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:26:41	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:26:23	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:26:09	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:25:51	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:25:36	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:25:03	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:24:50	AddParticipant	admin (NetworkAdmin)	view record
2017-11-12, 21:24:25	AddParticipant	admin (NetworkAdmin)	view record

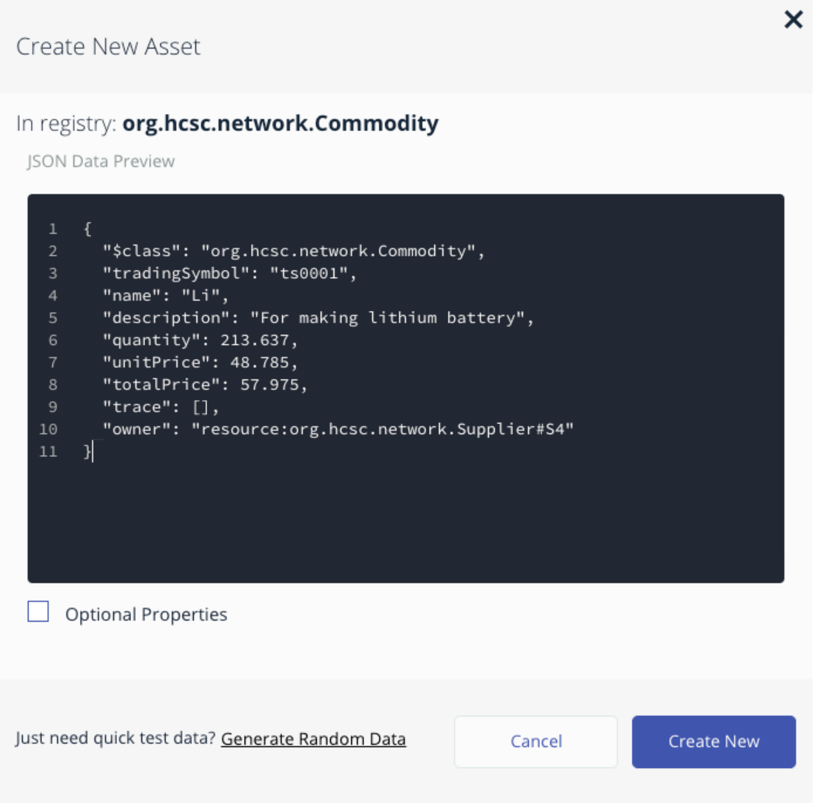
Figure 27. *Historian Record of adding participants on blockchain*

After this step, all the participants in the supply chain are created. In other words, they are ready to interact with each other with transaction. One thing still need to highlight is all the data recording in the system is tamper-proof, which is the single source of trust.

5.1.2 Asset Registry

There are two types of assets in the validation scenario: Commodity and PO (purchase order). Asset registry supports to create, update and delete assets with access control.

Figure 28 demonstrates how to create a new commodity by inputting compulsory information for constructing the asset.

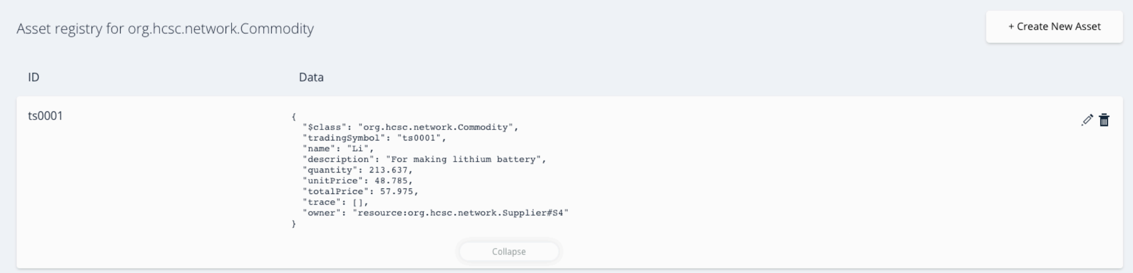

Create New Asset
In registry: org.hcsc.network.Commodity
JSON Data Preview
□ Optional Properties
Just need quick test data? Generate Random Data
Cancel
Create New

```
{
  "$class": "org.hcsc.network.Commodity",
  "tradingSymbol": "ts0001",
  "name": "Li",
  "description": "For making lithium battery",
  "quantity": 213.637,
  "unitPrice": 48.785,
  "totalPrice": 57.975,
  "trace": [],
  "owner": "resource:org.hcsc.network.Supplier#S4"
}
```

Figure 28. *Creating a new commodity*

In this example, the class field indicates which model we are creating, which is commodity. *TradingSymbol* is the unique identifier of this instance, which is ts0001. Other information related to this instance, like *name*, *description*, *quanitity* is also needed to create a new instance of manufacturer. Some fields like *unitPrice, totalPrice and trace* are optional fields. It is important to highlight the field *owner* that it is the key field which show the ownership between assets and participants.

After confirmation of creating new Commodity *ts0001*, Figure 29 is a JSON format transaction data recording to blockchain with all the data we input.



```
{
  "$class": "org.hcsc.network.Commodity",
  "tradingSymbol": "ts0001",
  "name": "Li",
  "description": "For making lithium battery",
  "quantity": 213.637,
  "unitPrice": 48.785,
  "totalPrice": 57.975,
  "trace": [],
  "owner": "resource:org.hcsc.network.Supplier#S4"
}
```

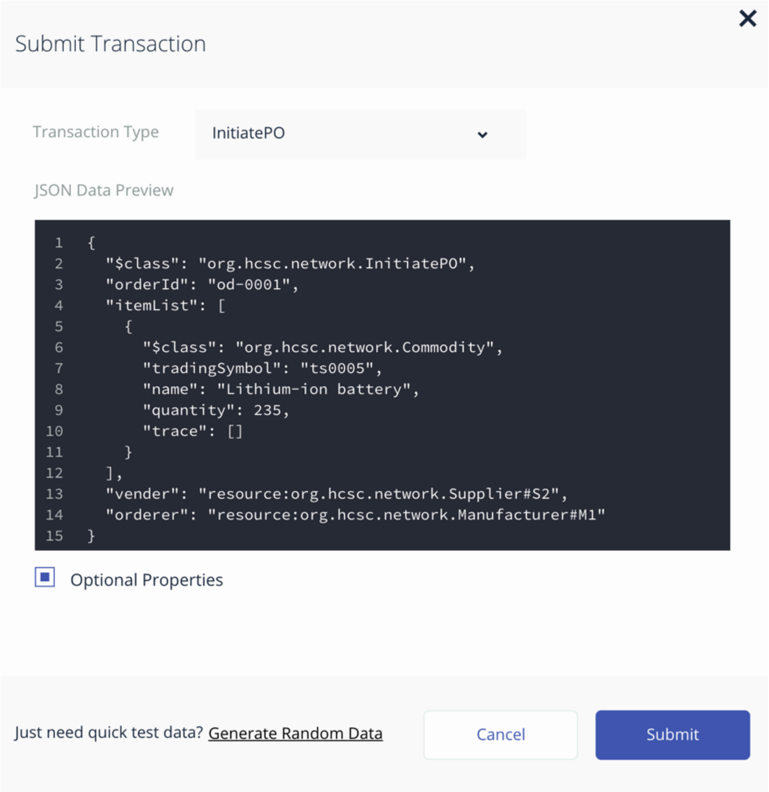
Figure 29. *New Asset being created in the asset registry*

Asset registry features the item-level transparency of entire supply chain, which means all the items has its own information resided in the blockchain system. Compared to easy tampering of traditional database system, blockchain ensures the trust of data by its distributed network feature.

5.1.3 Transaction Registry

Transactions are the mechanism by which participants interact with assets. After creating assets and participants in the previous two sections, it is possible to trigger transaction to fulfill the real activities in the daily supply chain. There are three main interaction which defined in the validated scenario: Initiate PO, assemble parts and transfer commodity.

Figure 30 shows that manufacturer M1 intends to issue a PO to supplier S2 by submitting *initiatePO* transaction. The field *orderId* is the identifier of the order, which can be referenced later when vender tend to transfer commodity to orderer. ItemList is the array format field include all the items needed to purchase from the vendor.



Submit Transaction

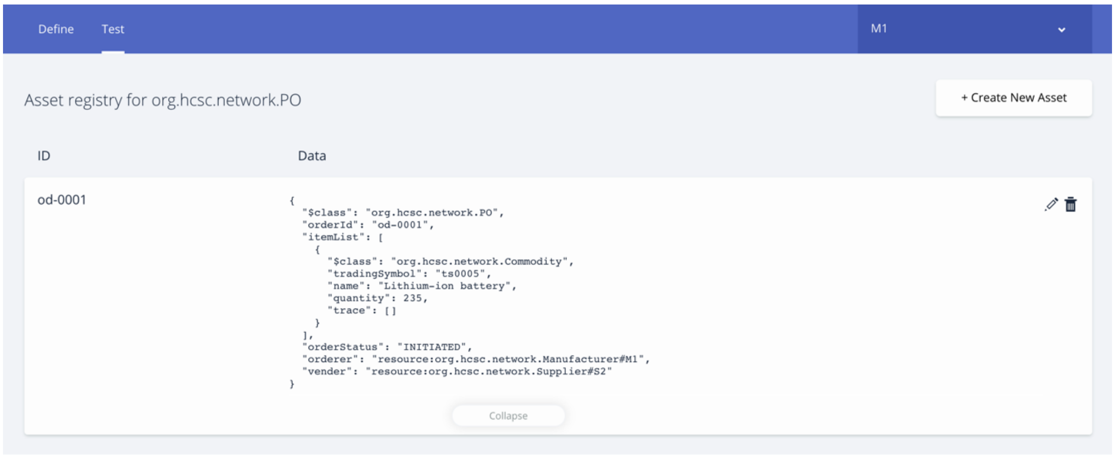
Transaction Type: InitiatePO

JSON Data Preview

```
{
  "$class": "org.hcsc.network.InitiatePO",
  "orderId": "od-0001",
  "itemList": [
    {
      "$class": "org.hcsc.network.Commodity",
      "tradingSymbol": "ts0005",
      "name": "Lithium-ion battery",
      "quantity": 235,
      "trace": []
    }
  ],
  "vender": "resource:org.hcsc.network.Supplier#S2",
  "orderer": "resource:org.hcsc.network.Manufacturer#M1"
}
```

Figure 30. *Issue a Purchase Order(PO)*

Afterwards, one new PO will be created and appear in the user interface for Manufacturer M1 and supplier S2. See Figure 31.



Asset registry for org.hcsc.network.PO

ID	Data

od-0001

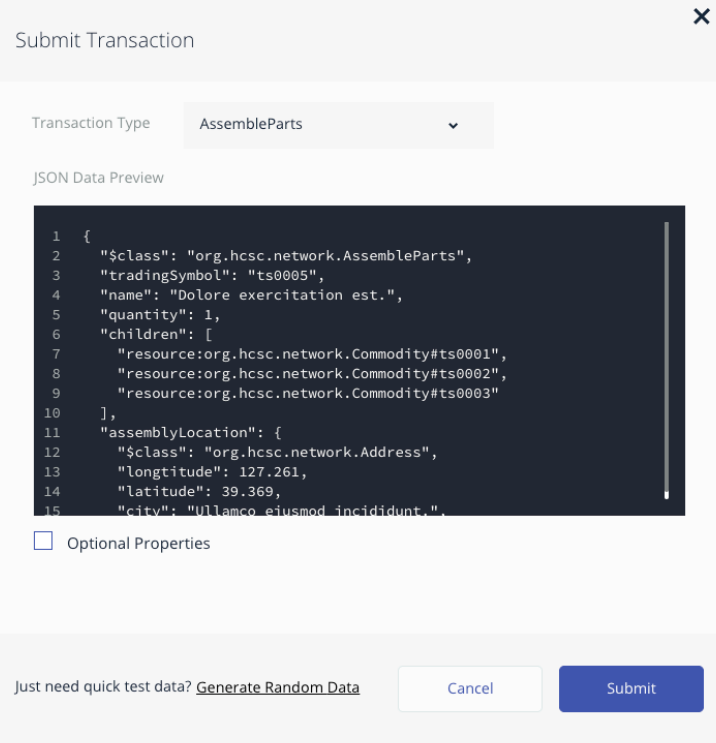
```
{
  "$class": "org.hcsc.network.PO",
  "orderId": "od-0001",
  "itemList": [
    {
      "$class": "org.hcsc.network.Commodity",
      "tradingSymbol": "ts0005",
      "name": "Lithium-ion battery",
      "quantity": 235,
      "trace": []
    }
  ],
  "orderStatus": "INITIATED",
  "orderer": "resource:org.hcsc.network.Manufacturer#M1",
  "vender": "resource:org.hcsc.network.Supplier#S2"
}
```

Figure 31. *Record of PO in asset registry*

It is critical to mention that, thanks to access control, only the orderer and vender of the PO can access the information of this PO. Orderer is the signature from the participant who issue the purchase order while vender is the target seller for the PO.

Assembly usually happens in the upstream of the supply chain, mostly suppliers and manufacturer. In this process, all the parts will be assembled into one new item, thus all the information of parts should be contained and referenced in the new item.

Figure 32 shows the submitting transaction from suppler S2. The key field *children* include all the referenced parts or raw materials, which is crucial to keep the source data for tracking and tracing in the future. Besides, the field *assemblyLocation* is the important piece of information to record the location of happening of assembly, which is ideal to input from the GPS device.



Submit Transaction

Transaction Type: AssembleParts

JSON Data Preview

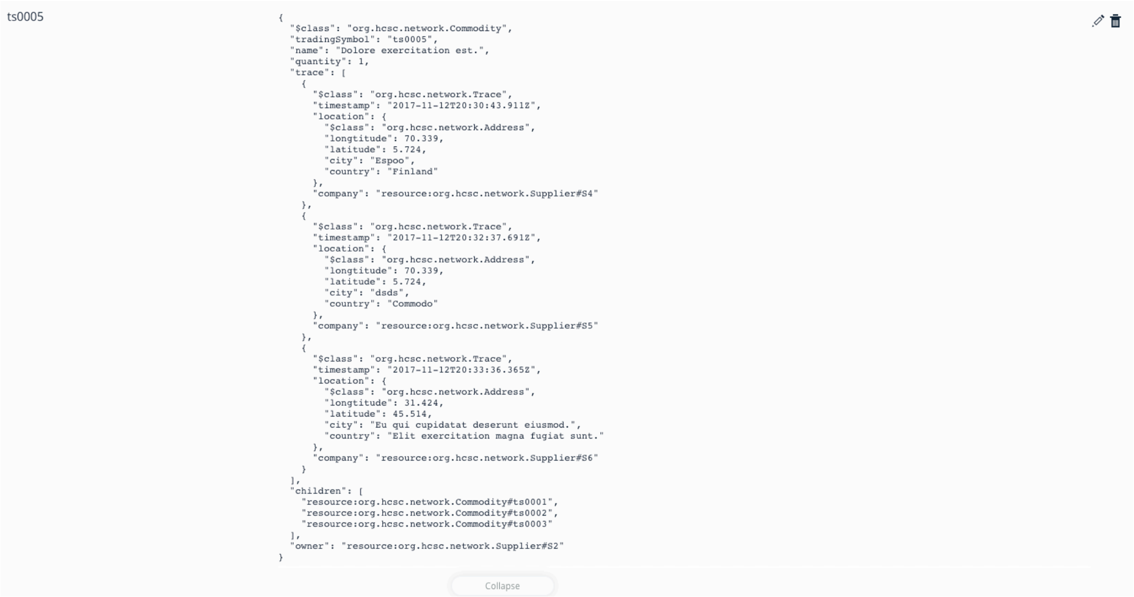
```
{
  "$class": "org.hcsc.network.AssembleParts",
  "tradingSymbol": "ts0005",
  "name": "Dolore exercitation est.",
  "quantity": 1,
  "children": [
    "resource:org.hcsc.network.Commodity#ts0001",
    "resource:org.hcsc.network.Commodity#ts0002",
    "resource:org.hcsc.network.Commodity#ts0003"
  ],
  "assemblyLocation": {
    "$class": "org.hcsc.network.Address",
    "longtitude": 127.261,
    "latitude": 39.369,
    "city": "Ullamco eiusmod incididunt.",
```

□ Optional Properties

Just need quick test data? Generate Random Data

Figure 32. *Assembles Parts transaction*

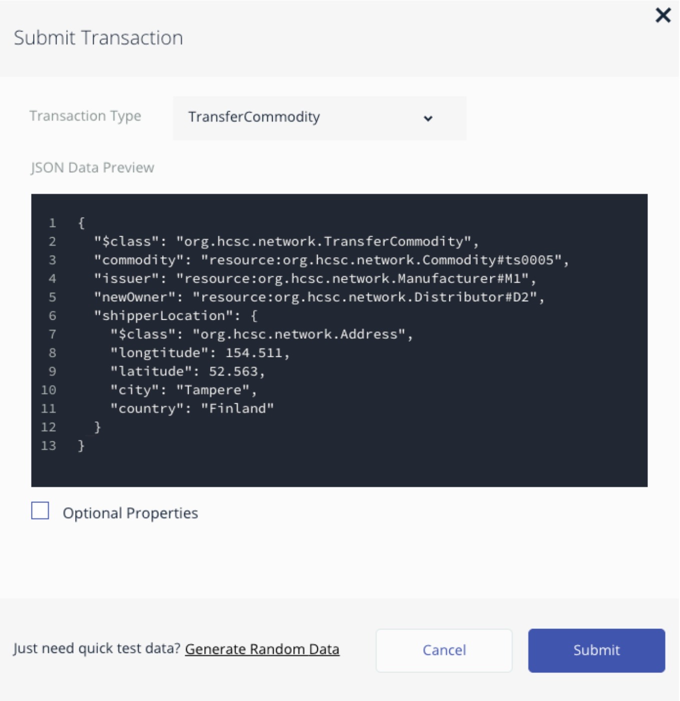
Subsequently, a new commodity *ts0005* is created and inherited the information of its parts. It is important to highlight the field *trace*, which emerges all the previous record of trace from its children.



```
{
  "$class": "org.hcsc.network.Commodity",
  "tradingSymbol": "ts0005",
  "name": "Dolore exercitation est.",
  "quantity": 1,
  "trace": [
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:30:43.911Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 70.339,
        "latitude": 5.724,
        "city": "Espoo",
        "country": "Finland"
      },
      "company": "resource:org.hcsc.network.Supplier#S4"
    },
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:32:37.691Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 70.339,
        "latitude": 5.724,
        "city": "dsds",
        "country": "Commodo"
      },
      "company": "resource:org.hcsc.network.Supplier#S5"
    },
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:33:36.365Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 31.424,
        "latitude": 45.514,
        "city": "Eu qui cupidatat deserunt eiusmod.",
        "country": "Elit exercitation magna fugiat sunt."
      },
      "company": "resource:org.hcsc.network.Supplier#S6"
    }
  ],
  "children": [
    "resource:org.hcsc.network.Commodity#ts0001",
    "resource:org.hcsc.network.Commodity#ts0002",
    "resource:org.hcsc.network.Commodity#ts0003"
  ],
  "owner": "resource:org.hcsc.network.Supplier#S2"
}
```

Figure 33. *Result of assemble parts transaction*

Another most frequent activity is transferring commodities between participants from upstream to downstream in the supply chain. Figure 34 shows manufacture M1 tend to submit transaction to transfer commodity to distributor D2. The key fields are *issuer*, *newOwner* and *shipperLocation*.

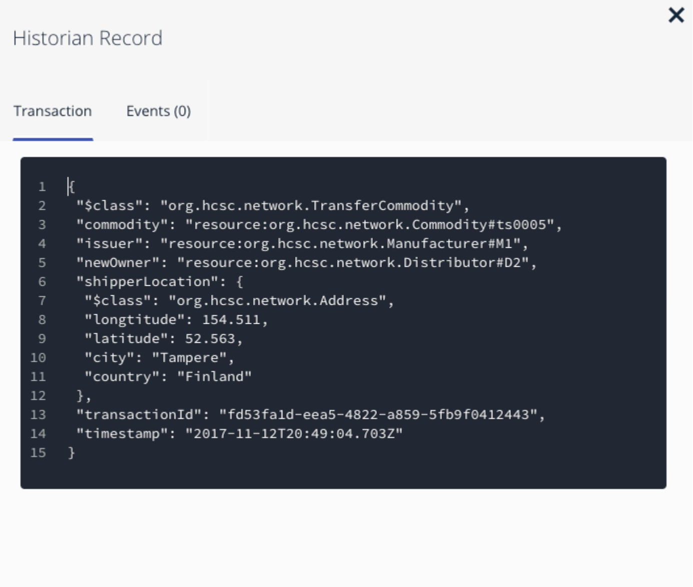


Transaction Type: TransferCommodity

```
{
  "$class": "org.hcsc.network.TransferCommodity",
  "commodity": "resource:org.hcsc.network.Commodity#ts0005",
  "issuer": "resource:org.hcsc.network.Manufacturer#M1",
  "newOwner": "resource:org.hcsc.network.Distributor#D2",
  "shipperLocation": {
    "$class": "org.hcsc.network.Address",
    "longtitude": 154.511,
    "latitude": 52.563,
    "city": "Tampere",
    "country": "Finland"
  }
}
```

Figure 34. *An example of transferring Commodity*

Afterwards, transaction of transferring the commodity is recorded in the blockchain forever. The information includes transaction id, timestamp and transaction-related information. See Figure 35:



```
{
 "$class": "org.hcsc.network.TransferCommodity",
 "commodity": "resource:org.hcsc.network.Commodity#ts0005",
 "issuer": "resource:org.hcsc.network.Manufacturer#M1",
 "newOwner": "resource:org.hcsc.network.Distributor#D2",
 "shipperLocation": {
  "$class": "org.hcsc.network.Address",
  "longtitude": 154.511,
  "latitude": 52.563,
  "city": "Tampere",
  "country": "Finland"
 },
 "transactionId": "fd53fa1d-eea5-4822-a859-5fb9f0412443",
 "timestamp": "2017-11-12T20:49:04.703Z"
}
```

Figure 35. *Transaction Record*

At this time, the asset is no longer belonging to manufacturer M1 but distributor D2. The asset *ts0005* is not visible to M1 anymore, but the history information related to this asset is still available to M1.

5.2 Traceability

This section will discuss how blockchain business network support to achieve traceability across supply chain. The previous section illustrates how assets, participants and transaction registry recorded in blockchain. Thus, the system is available to track parts, materials and transaction by their identifier.

Compared to traditional database system, blockchain store immutable data and provide deep search ability through as many as 5 horizontal layers. This backwards search capability is the base of establishing provenance of any asset which is made up of other components from supplies.

Figure 36 shows a scenario how backwards traceability works in traditional data base. End user – Customer C2 started to find the faulty part of the phone and asked Retailer R2 for information. Then R2 have to track back with their own data base then find if it is from distributor D1 or D2. Then D1 and D2 have to asked their own manufacturer. It will take much of effort and time to track back if the horizontal layers are deep. And most important thing is the data is mutable in everyone’s own system so that there is no trust of fault. In conclusion, it is difficult to realized backwards traceability in traditional way with centralized database.

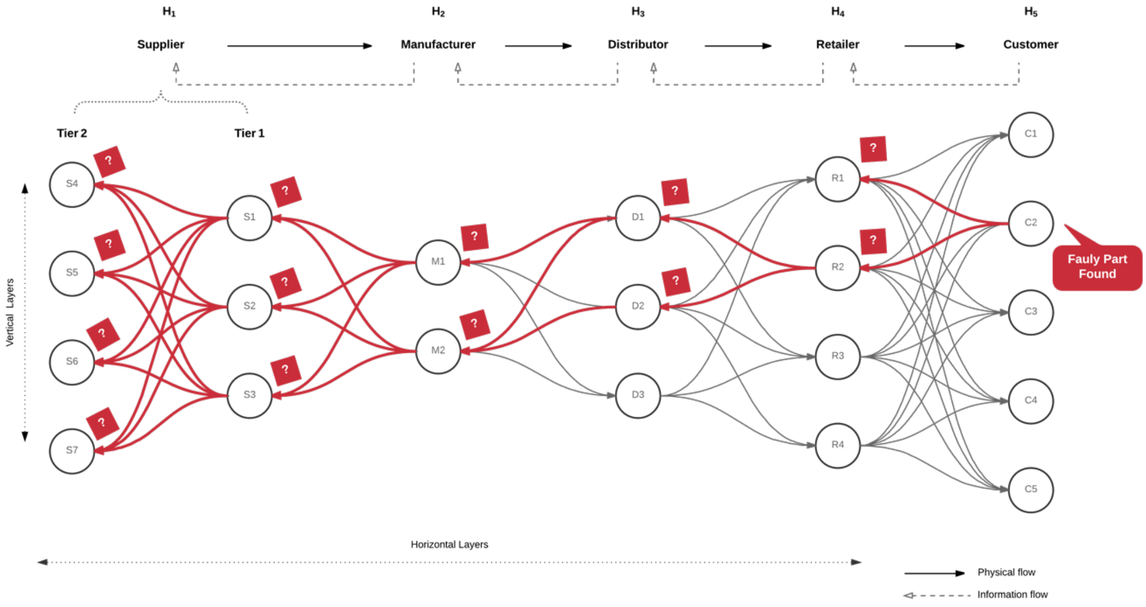


Figure 36. Backwards Traceability in traditional ways

On the contrary, the blockchain provides a trusted way to allow every participant on a supply chain network to input and track numbered parts that are produced and used on a specific phone.

Program 7 explicitly shows trace data of commodity *ts0005.* The field *trace* is an array that chronologically enumerates all the footprint of commodity with time, place and participants. Besides, this trace data that resides in blockchain is immutable.

```
{
  "$class": "org.hcsc.network.Commodity",
  "tradingSymbol": "ts0005",
  "name": "Dolore exercitation est.",
  "quantity": 1,
  "trace": [
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:30:43.911Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 70.339,
        "latitude": 5.724,
        "city": "Espoo",
        "country": "Finland"
      },
      "company": "resource:org.hcsc.network.Supplier#S4"
    },
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:32:37.691Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 70.339,
        "latitude": 5.724,
        "city": "dsds",
        "country": "Commodo"
      },
      "company": "resource:org.hcsc.network.Supplier#S5"
    },
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:33:36.365Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 31.424,
        "latitude": 45.514,
        "city": "Eu qui cupidatat deserunt eiusmod.",
        "country": "Elit exercitation magna fugiat sunt."
      },
      "company": "resource:org.hcsc.network.Supplier#S6"
    },
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:43:04.775Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 165.057,
        "latitude": 51.97,
        "city": "Tampere",
        "country": "Finland"
      },
      "company": "resource:org.hcsc.network.Supplier#S2"
    },
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:49:04.887Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 154.511,
        "latitude": 52.563,
        "city": "Tampere",
        "country": "Finland"
      },
      "company": "resource:org.hcsc.network.Manufacturer#M1"
    },
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:52:55.326Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 53.874,
        "latitude": 55.693,
        "city": "Officia do occaecat ad.",
        "country": "In incididunt."
      },
      "company": "resource:org.hcsc.network.Distributor#D2"
    },
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T20:54:15.375Z",
      "location": {
        "$class": "org.hcsc.network.Address",
        "longtitude": 158.232,
        "latitude": 35.091,
        "city": "Irure sit cillum labore.",
        "country": "Nulla dolor."
      },
      "company": "resource:org.hcsc.network.Retailer#R2"
    }
  ],
  "children": [
    "resource:org.hcsc.network.Commodity#ts0001",
    "resource:org.hcsc.network.Commodity#ts0002",
    "resource:org.hcsc.network.Commodity#ts0003"
  ],
  "owner": "resource:org.hcsc.network.Customer#C2",
  "issuer": "resource:org.hcsc.network.Retailer#R2"
}
```

Program 7. *Trace data of a commodity in blockchain*

Figure 37 shows how backwards traceability work in blockchain's way. Compared to traditional database, blockchain offers a faster and clearer backwards traceability to find the source of problem. And the information related to trace is tamper-proof thus trustful due to consensus mechanism and distributed network.

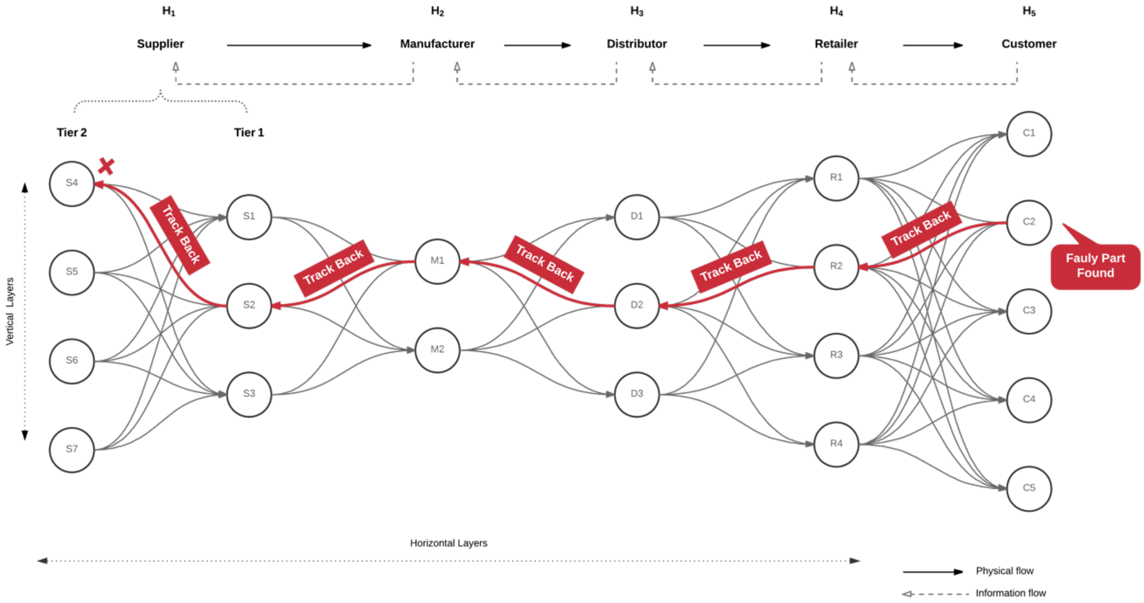


Figure 37. *Backwards Traceability in blockchain business network*

5.3 Confidentiality

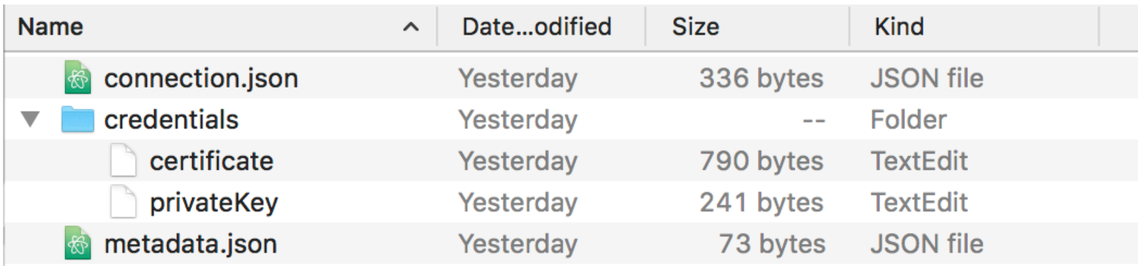
Even transparency and traceability are so important to supply chain management. However, without confidentiality enable between competitors, the whole system will be totally useless.

Fortunately, confidentiality is achieved in our validation scenario with the conjunction with the two aspects. One is membership service, which provides access control service to any data recorded. The other one is business logic deployed on a blockchain network, which is mentioned in 4.3.3.

5.3.1 Membership service

As illustrated in 3.3.4, membership service provider (MSP) is a component that aims to offer an abstraction of cryptographic mechanisms and protocols. Thus, MSP is responsible for issuing and validating certificates, and user authentication.

Figure 38 shows all the files of identity card issued by MSP, which is used to identify client and connect to blockchain system securely. The connection profile (connection.json) defines all the necessary configuration parameters, which contains the TCP/IP addresses and ports for the peers and certificate authority. Credential is made up of X.509 certificate and private key. Metadata is the data of identity card itself.



Name	Date...odified	Size	Kind
connection.json	Yesterday	336 bytes	JSON file
▼ credentials	Yesterday	--	Folder
certificate	Yesterday	790 bytes	TextEdit
privateKey	Yesterday	241 bytes	TextEdit
metadata.json	Yesterday	73 bytes	JSON file

Figure 38. *Credential in Identity Card*

With issued identity card, the company or organization can easily connect to blockchain network and access data confidentially and safely.

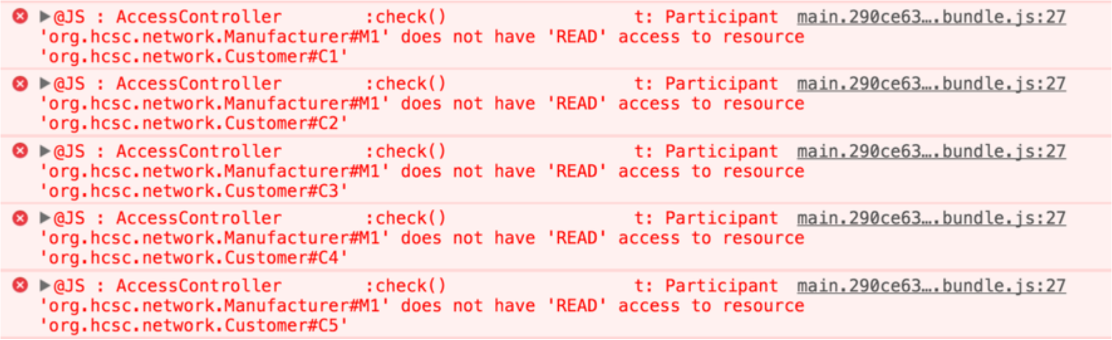
5.3.2 Access Control in business Logic

Permission file (permission.acl) defines the access control rules for the business network, to enforce which participants have access to the data on the ledger and under which conditions. A short snippet shows (see Program 8) how to allow participants to access its own commodity.

```
rule ReadCommodity {
    description: "All participants can read its own goods"
    participant(m): "org.hcsc.network.*"
    operation: READ
    resource(v): "org.hcsc.network.Commodity"
    condition: (v.owner.getIdentifier() == m.getIdentifier())
    action: ALLOW
}
```

Program 8. *Snippet in permission file to allow participants to access its own commodity*

Figure 39 shows the result when Manufacturer M1 try to access other participants' resource, which clearly indicate the operation is denied because M1 do not have 'Read' access to the resource.



```
▶@JS : AccessController        :check()                t: Participant  main.290ce63….bundle.js:27
'org.hcsc.network.Manufacturer#M1' does not have 'READ' access to resource
'org.hcsc.network.Customer#C1'
▶@JS : AccessController        :check()                t: Participant  main.290ce63….bundle.js:27
'org.hcsc.network.Manufacturer#M1' does not have 'READ' access to resource
'org.hcsc.network.Customer#C2'
▶@JS : AccessController        :check()                t: Participant  main.290ce63….bundle.js:27
'org.hcsc.network.Manufacturer#M1' does not have 'READ' access to resource
'org.hcsc.network.Customer#C3'
▶@JS : AccessController        :check()                t: Participant  main.290ce63….bundle.js:27
'org.hcsc.network.Manufacturer#M1' does not have 'READ' access to resource
'org.hcsc.network.Customer#C4'
▶@JS : AccessController        :check()                t: Participant  main.290ce63….bundle.js:27
'org.hcsc.network.Manufacturer#M1' does not have 'READ' access to resource
'org.hcsc.network.Customer#C5'
```

Figure 39. *Asset registry refuse to delete resource duo to the access control*

5.4 Interoperability

Once the business network is tested and in place, integration with existing systems (like ERP system) and other web/mobile application is one of the most practical problem in reality. This section will show how blockchain network can be integrated with existing systems by using a REST API, which allows company or organization to pull data from existing business systems and convert it to assets or participants in a Composer business network.

In the implementation of validation scenarios, a REST Server generated by composer tools on the business network. Table 5 illustrates all the REST API collection for the scenario designed. The table contains the endpoints and methods to manipulate all the assets, participants, transactions and system-related activities. Besides, it can be configured to authenticate the participants in the business network, ensuring that credentials and permissions are enforced.

Table 5. *REST API collection*

Type	Resource	Endpoint	Available Methods	Description
Asset	Commodity	/Commodity	GET, POST, HEAD, PUT, DELETE	Find, create, update, delete instance(s) of the commodity asset
	PO	/PO	GET, POST, HEAD, PUT, DELETE	Find, create, update, delete instance(s) of the PO asset
Participant	Supplier	/Supplier	GET, POST, HEAD, PUT, DELETE	Find, create, update, delete instance(s) of the Supplier Participant
	Manufacturer	/Manufacturer	GET, POST, HEAD, PUT, DELETE	Find, create, update, delete instance(s) of the Manufacturer Participant
	Distributor	/Distributor	GET, POST, HEAD, PUT, DELETE	Find, create, update, delete instance(s) of the Distributor Participant
	Retailer	/Retailer	GET, POST, HEAD, PUT, DELETE	Find, create, update, delete instance(s) of the Retailer Participant
	Customer	/Customer	GET, POST, HEAD, PUT, DELETE	Find, create, update, delete instance(s) of the Customer Participant
Transaction	InitatiatePO	/InitiatePO	GET, POST	Find all instances or create a new instance of the initiatePO transaction
	TransferCom-	/TransferCom modity	GET, POST	Find all instances or create a new instance

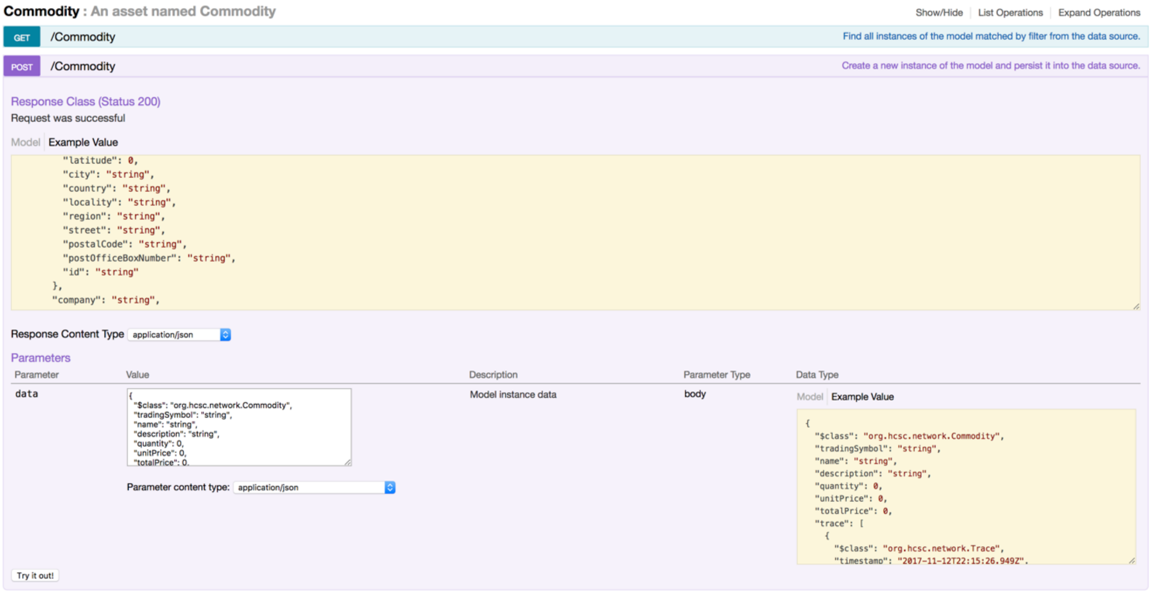
	modity			of the TransferCommodity transaction
System	Historian	/system/historian	GET	Get all Historian Records from the Historian
	Identities	/system/identities	GET	Get all identities from the identity registry
		/system/identities/{id}/revoke	POST	Revoke the specified identity
		/system/identities/bind	POST	Bind an identity to specified identity
		/system/identities/issue	POST	issue an identity to spcified participant
	ping	/system/ping	GET	Test the connection to the business network

Table 6 dive into one example of finding, creating, updating and deleting commodity by calling URL with GET, POST, PUT, HEAD and DELETE methods.

Table 6. *One example of REST API of Commodity*

Resource	Method	URL	Description
Commodity	GET	/Commodity	Find all instances of the Commodity matched by filter from the blockchain
	POST	/Commodity	Create a new instance of the Commodity and persist it into the blockchain
	GET	/Commodity/{id}	Find a Commodity instance by {{id}} from the blockchain
	HEAD	/Commodity/{id}	Check whether a Commodity instance exists in the blockchain
	PUT	/Commodity/{id}	Replace attributes for a Commodity instance and persist it into the blockchain
	DELETE	/Commodity/{id}	Delete a Commodity instance by {{id}} from the blockchain

Figure 40 shows a user interface for testing API of creating a commodity. All the input parameters should be including as JSON format data in the body.



Commodity : An asset named Commodity

GET /Commodity — Find all instances of the model matched by filter from the data source.

POST /Commodity — Create a new instance of the model and persist it into the data source.

Response Class (Status 200)

Request was successful

Model | Example Value

```
        "latitude": 0,
        "city": "string",
        "country": "string",
        "locality": "string",
        "region": "string",
        "street": "string",
        "postalCode": "string",
        "postOfficeBoxNumber": "string",
        "id": "string"
      },
      "company": "string",
```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	{ "$class": "org.hcsc.network.Commodity", "tradingSymbol": "string", "name": "string", "description": "string", "quantity": 0, "unitPrice": 0, "totalPrice": 0, Parameter content type: application/json	Model instance data	body	Model | Example Value

```
{
  "$class": "org.hcsc.network.Commodity",
  "tradingSymbol": "string",
  "name": "string",
  "description": "string",
  "quantity": 0,
  "unitPrice": 0,
  "totalPrice": 0,
  "trace": [
    {
      "$class": "org.hcsc.network.Trace",
      "timestamp": "2017-11-12T22:15:26.949Z",
```

Figure 40. *User interface of API test field created by Loopback*

When deploying REST API server in a production environment, the REST server can be configured to be secured with HTTPS and TLS (Transport Layer Security). Once the REST server has been configured with HTTPS and TLS, all data transferred between the REST server and all of the REST clients is encrypted.

6. DISCUSSION

Regrading to the previous chapter where the result of validation scenario has been implemented in blockchain platform, this chapter discuss the personal perspective as founding of the presented solution.

Returning to the addressed hypothesis in 1.2, the usage of blockchain technology is expected to improve transparency and traceability. In this regard, the presented approach illustrates this feature by the conducted result. As shown, the solution validates the operability and visibility of registry of asset, participants and transaction. At every important data during segment of creating, assembling and transferring commodity, all the data is recorded in to blockchain that is immutable state. Besides, every trace of the commodity across supply chain is stored to enable strong backwards traceability no matter of deep layers of supply chain.

The second point in the hypothesis list considers using permissioned blockchain technology to keep confidentiality at the same time. In this matter, the result prove that the confidentiality is achieved with membership service and access control mechanism. Participant can only access and manipulate the data with permission and pre-defined rules that is agreed with all participants. Access control roles are also store as smart contract in the blockchain to ensure immutability

The last point in the hypothesis is the expectation of integration between blockchain and other application. The validation scenario verifies the possibility to interoperate with existing ERP system and other web or mobile application by generating REST API server for operating and handling entry and access of data. When deploying in a production environment, the REST API server could be configured to be secured with HTTPS and TLS (Transport Layer Security).

In conclusion, the result verifies the hypothesis with use case and validation scenario. The presented solution proves that blockchain technology can improve transparency and traceability of industrial supply chain, while keeping confidentiality and achieving interoperability. Table 7 summarized the comparison between traditional database and permissioned blockchain solution.

Table 7. *Comparison between traditional database and permissioned blockchain*

	Before	After
Technology	Traditional Database	Permissioned Blockchain
Network	Centralized	Decentralized
Immutability	Editable	Append-only
Trust	Depend on trust of authority	Trust on technology
Transparency	Low: keep their own ledgers	High: Ledgers are distributed and shared
Traceability	Hard to do backwards trace due to centralized system	Easy to trace back because immutable data
Confidentiality	Strong encryption but legacy issues still enable fraud (e.g. signature	Strong access control and membership service providing with certificate and confidential
Interoperability	Different systems, hard to integrate with other system	Easy to integrate with other system with REST API
Performance	Fast	Normal
Scalability	Harder and expensive	Easier and cheaper
Security	Strong encryption but legacy issues still enable fraud (e.g. signature	Strong access control and membership service providing with certificate and confidential

7. CONCLUSION

This is the final chapter of this thesis that reiterates the work for the thesis and highlight of the founding. Furthermore, results for the presented approach and validation scenarios of the research are discussed. Chapter closes with the recommendations for future work.

7.1 Research Conclusion

This thesis has been focusing on an approach of improving transparency and traceability of industrial supply chain with blockchain technology.

First of all, an introduction has been presented with the motivation, problem statement, objectives, and research limitation. In this chapter, concern for transparency and traceability of modern supply chain is addressed as well as corresponding incentive and goal to solve the problem.

The second chapter presented a State of the Art of current supply chain data management and blockchain technology. In this chapter, the difference between traditional database and blockchain has been compared in different aspects. Besides, it also introduced the advantages of the blockchain technology applied in supply chain.

The third chapter presented a methodology of creating supply chain model on blockchain network, allowing company or organization to model and apply their business into blockchain network. The methodology showed the architecture of solution and the progress of business network modeling.

After that, chapter four introduced the implementation of business network on blockchain with concrete use case. In this chapter, a validation scenario was being designing, modeling and deploying on the permissioned blockchain.

Then, chapter five presented the results of deploying business network on blockchain with validation scenario. The result analyses how blockchain technology assist to improve transparency and traceability of industrial supply chain as well as keep confidentiality and interoperability.

Last but not least, chapter six summarize the founding and highlight most critical features. As well, a comparison was done between traditional database and permissioned blockchain solution.

7.2 Future Work

This thesis purposes a promising approach of improving transparency and traceability of supply chain by blockchain technology with solid implementation. However, due to the limitation of research and implementation, there are still some aspects worth researching in the future:

1. Automatic payment

During the research, one possible further research direction is automating financial transaction flow, which is potential solution to decrease inefficiency of payment across supply chain. Smart contract on blockchain can help to issue the payment automatically when certain condition is fulfilled.

2. Performance test

One concerns before applying this approach is reliability and stability of the entire system. Some benchmarking tests (like latency test, stress test, long run test, concurrency test, complex transaction test, scalability test) should be done before coming to production. Because in the modern supply chain, reliability and stability are the most critical factors of data system to manage the risks.

3. Integration with Internet of Things and Machine Learning

As illustrated before, the system provides REST API to integrate with other system and web application. However, the data entry might be related to internet of things. And how to integrate blockchain with IoT is one of the challenges. Meanwhile, IoT data will cause the data flood problem, so using machine learning to extract the most valuable information could be a great solution.

REFERENCES

[1] UN Global Compact and BSR, “A Guide to Traceability: A Practical Approach to Advance Sustainability in Global Supply Chains | UN Global Compact,” *Sustainable Supply Chains: Resources & Practices*, 2014. [Online]. Available: https://www.unglobalcompact.org/library/791. [Accessed: 01-Jul-2017].

[2] S. New, “The Transparent Supply Chain,” *Harvard Business Review*, 01-Oct-2010. [Online]. Available: https://hbr.org/2010/10/the-transparent-supply-chain. [Accessed: 01-Jul-2017].

[3] U. Barchetti, A. Bucciero, M. D. Blasi, L. Mainetti, and L. Patrono, “RFID, EPC and B2B convergence towards an item-level traceability in the pharmaceutical supply chain,” in *2010 IEEE International Conference on RFID-Technology and Applications*, 2010, pp. 194–199.

[4] R. Wang and W. A. Günthner, “Design and development of a Traceability Service for EPC-enabled food supply chains,” in *SoftCOM 2012, 20th International Conference on Software, Telecommunications and Computer Networks*, 2012, pp. 1–6.

[5] S. A. Abeyratne and R. P. Monfared, “Blockchain ready manufacturing supply chain using distributed ledger,” *Authors Publ. ESAT*, 2016.

[6] “Blockchain Supply Chain Infographic: The Paper Trail of a Shipping Container,” 03-Mar-2017. [Online]. Available: https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=XI912347USEN&. [Accessed: 10-Oct-2017].

[7] S. Banker, “Blockchain In The Supply Chain: Too Much Hype,” *Forbes*. [Online]. Available: https://www.forbes.com/sites/stevebanker/2017/09/01/blockchain-in-the-supply-chain-too-much-hype/. [Accessed: 09-Oct-2017].

[8] M. Pilkington, “Blockchain Technology: Principles and Applications,” Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 2662660, Sep. 2015.

[9] S. Apte and N. Petrovsky, “Will blockchain technology revolutionize excipient supply chain management?,” *J. Excip. Food Chem.*, vol. 7, no. 3, Sep. 2016.

[10] S. New and R. Westbrook, *Understanding Supply Chains: Concepts, Critiques, and Futures*. OUP Oxford, 2004.

[11] “ISO 9000 2015 Definitions in Plain English.” [Online]. Available: http://www.praxiom.com/iso-definition.htm#Traceability. [Accessed: 13-Oct-2017].

[12] European Commission, “General food law. Traceability factsheet.” [Online]. Available: https://www.scribd.com/document/64344685/Factsheet-Trace-Ability. [Accessed: 13-Oct-2017].

[13] Anonymous, “GS1 Global Traceability Standard,” 25-Nov-2014. [Online]. Available: https://www.gs1.org/traceability/traceability/1-3-0. [Accessed: 13-Oct-2017].

[14] B. Mitchell, “Exploring Computer Network Topologies Like Bus, Ring and Star,” *Lifewire*. [Online]. Available: https://www.lifewire.com/computer-network-topology-817884. [Accessed: 18-Nov-2017].

[15] D. Groth and T. Skandier, *Network+ Study Guide, 4th Edition*, 4 edition. San Francisco, Calif. ; London: Sybex, 2005.

[16] P. Baran, “On Distributed Communications Networks,” *IEEE Trans. Commun. Syst.*, vol. 12, no. 1, pp. 1–9, Mar. 1964.

[17] "Conversation Agent - Valeria Maltoni - How Everything is Connected to Everything Else and What it Means." [Online]. Available: http://www.conversationagent.com/2016/05/how-everything-is-connected-to-everything-else-and-what-it-means.html. [Accessed: 20-Oct-2017].

[18] "Introduction — hyperledger-fabricdocs master documentation." [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html#what-is-hyperledger-fabric. [Accessed: 16-Oct-2017].

[19] G. Gideon, "Four genuine blockchain use cases | MultiChain," *Private blockchains*. [Online]. Available: http://www.multichain.com/blog/2016/05/four-genuine-blockchain-use-cases/. [Accessed: 01-Jul-2017].

[20] M. Sato and S. Matsuo, "Long-Term Public Blockchain: Resilience against Compromise of Underlying Cryptography," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017, pp. 1–8.

[21] "Bitcoin: A Peer-to-Peer Electronic Cash System." [Online]. Available: https://bitcoin.org/en/bitcoin-paper. [Accessed: 21-Oct-2017].

[22] "Mastering Bitcoin." [Online]. Available: http://chimera.labs.oreilly.com/books/1234000001802/ch07.html#merkle_trees. [Accessed: 21-Nov-2017].

[23] J. A. D. Donet, C. Pérez-Solà, and J. Herrera-Joancomartí, "The Bitcoin P2P Network," in *Financial Cryptography and Data Security*, 2014, pp. 87–102.

[24] antonylewis2015, "A gentle introduction to blockchain technology," *Bits on blocks*, 09-Sep-2015. .

[25] H. Halpin and M. Piekarska, "Introduction to Security and Privacy on the Blockchain," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2017, pp. 1–3.

[26] J. M. Cheong Raymond, "Consensus: Immutable agreement for the internet of value | KPMG | CN," *KPMG*, 03-Oct-2016. [Online]. Available: https://assets.kpmg.com/content/dam/kpmg/pdf/2016/06/kpmg-blockchain-consensus-mechanism.pdf. [Accessed: 20-Nov-2017].

[27] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A Review on Consensus Algorithm of Blockchain," *2017 IEEE Int. Conf. Syst. Man Cybern. SMC*, Oct. 2017.

[28] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," *Microsoft Res.*, vol. 20, Jan. 2017.

[29] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557–564.

[30] M. Vukolić, "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication," in *Open Problems in Network Security*, 2015, pp. 112–125.

[31] "Nick Szabo -- Smart Contracts: Building Blocks for Digital Markets." [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html. [Accessed: 20-Nov-2017].

[32] Murphy, Sean, and C. Charley, "Can smart contracts be legally binding contracts?," 2016. [Online]. Available: https://sites-nortonrosefulbright.vuturevx.com/596/14051/uploads/r3-and-norton-rose-fulbright-white-paper-full- report-144581.pdf. [Accessed: 20-Nov-2017].

[33] M. Iansiti and K. R. Lakhani, “The Truth About Blockchain,” *Harvard Business Review*, 01-Jan-2017. [Online]. Available: https://hbr.org/2017/01/the-truth-about-blockchain. [Accessed: 19-Nov-2017].

[34] A. Alimoğlu and C. Özturan, “Design of a Smart Contract Based Autonomous Organization for Sustainable Software,” in *2017 IEEE 13th International Conference on e-Science (e-Science)*, 2017, pp. 471–476.

[35] A. Hahn, R. Singh, C. C. Liu, and S. Chen, “Smart contract-based campus demonstration of decentralized transactive energy auctions,” in *2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2017, pp. 1–5.

[36] P. Sreehari, M. Nandakishore, G. Krishna, J. Jacob, and V. S. Shibu, “Smart will converting the legal testament into a smart contract,” in *2017 International Conference on Networks Advances in Computational Technologies (NetACT)*, 2017, pp. 203–207.

[37] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, “Blockchain Challenges and Opportunities: A Survey,” *Int. J. Web Grid Serv.*, Dec. 2017.

[38] X. Xu *et al.*, “A Taxonomy of Blockchain-Based Systems for Architecture Design,” in *2017 IEEE International Conference on Software Architecture (ICSA)*, 2017, pp. 243–252.

[39] A. Varshney, “Types of Blockchain — Public, Private and Permissioned,” *Darwin Labs*, 01-Apr-2017. [Online]. Available: https://blog.darwinlabs.io/types-of-blockchain-public-private-and-permissioned-5b14fbfe38d4. [Accessed: 19-Oct-2017].

[40] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, “Survey of consensus protocols on blockchain applications,” in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2017, pp. 1–5.

[41] *corda: Corda is a distributed ledger platform designed to record, manage and automate legal agreements between business partners. Designed by (and for) the world&#39;s largest financial institutio..* Corda, 2017.

[42] A. Stanciu, “Blockchain Based Distributed Control System for Edge Computing,” in *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, 2017, pp. 667–671.

[43] “What Is Hyperledger? How the Linux Foundation builds an open platform,” *Blockgeeks*, 28-May-2017. [Online]. Available: https://blockgeeks.com/guides/what-is-hyperledger/. [Accessed: 22-Oct-2017].

[44] G. Volpicelli, “How the blockchain is helping stop the spread of conflict diamonds,” *WIRED UK*. [Online]. Available: http://www.wired.co.uk/article/blockchain-conflict-diamonds-everledger. [Accessed: 20-Nov-2017].

[45] “Provenance | From shore to plate: Tracking tuna on the blockchain,” *Provenance*. [Online]. Available: https://www.provenance.org/tracking-tuna-on-the-blockchain. [Accessed: 20-Nov-2017].

[46] F. Tian, “A supply chain traceability system for food safety based on HACCP, blockchain Internet of things,” in *2017 International Conference on Service Systems and Service Management*, 2017, pp. 1–6.

[47] C. Thomas, “Evaluating information communication technology (ICT) usage in humanitarian response : a SWOT analysis and proposal,” Jul. 2017.

[48] M. Pilkington, “Does the FinTech Industry Need a New Risk Management Philosophy? A Sequential Blockchain-based Typology for Virtual Currencies and e-

Money Services in Luxembourg," Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 2744899, Mar. 2016.

[49] A. Deshpande, K. Stewart, L. Lepetit, and S. Gunashekar, "Distributed Ledger Technologies/Blockchain," 2017. [Online]. Available: https://www.rand.org/pubs/external_publications/EP67133.html. [Accessed: 19-Nov-2017].

[50] M. E. Peck, "Blockchain world - Do you need a blockchain? This chart will tell you if the technology can solve your problem," *IEEE Spectr.*, vol. 54, no. 10, pp. 38–60, Oct. 2017.

[51] P. Sandner, "Comparison of Ethereum, Hyperledger Fabric and Corda," *Medium*, 25-Jun-2017. .

[52] "UsageFAQ - Hyperledger Fabric." [Online]. Available: http://fabricrepo.readthedocs.io/en/0928_master/FAQ/usage_FAQ/. [Accessed: 17-Nov-2017].

[53] "Top 6 technical advantages of Hyperledger Fabric v1.0 for blockchain networks," 21-Aug-2017. [Online]. Available: http://www.ibm.com/developerworks/cloud/library/cl-top-technical-advantages-of-hyperledger-fabric-for-blockchain-networks/index.html. [Accessed: 17-Nov-2017].

[54] "Typical Solution Architecture | Hyperledger Composer." [Online]. Available: https://hyperledger.github.io/composer/introduction/solution-architecture.html. [Accessed: 16-Oct-2017].

[55] "Typical Solution Architecture | Hyperledger Composer." [Online]. Available: https://hyperledger.github.io/composer/introduction/solution-architecture.html. [Accessed: 17-Nov-2017].

[56] "Membership Service Providers (MSP) — hyperledger-fabricdocs master documentation." [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release/msp.html. [Accessed: 17-Nov-2017].

[57] Linux Foundation, "Hyperledger Architecture, Volume 1 - Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus." Aug-2017.

[58] H. Min and G. Zhou, "Supply chain modeling: past, present and future," *Comput. Ind. Eng.*, vol. 43, no. 1, pp. 231–249, Jul. 2002.

[59] B. M. Beamon, "Supply chain design and analysis:: Models and methods," *Int. J. Prod. Econ.*, vol. 55, no. 3, pp. 281–294, Aug. 1998.

[60] C. J. Vidal and M. Goetschalckx, "A global supply chain model with transfer pricing and transportation cost allocation," *Eur. J. Oper. Res.*, vol. 129, no. 1, pp. 134–158, Feb. 2001.

[61] J. Grabis, "Publikācija: Joint Optimization of Physical and Information Flows in Supply Chains," in *publication.editionName*, 2013, pp. 1–10.

[62] A. Chibba and J. Rundquist, *Mapping flows -An analysis of the information flows within the integrated supply chain*. 2017.

[63] D. Silver, "The Automotive Supply Chain, Explained," *Self-Driving Cars*, 31-May-2016. .

[64] D. Steil, "Hyperledger Fabric v1.0," *Dapps.ai*. . [Online]. Available: http://dapps.ai/hyperledger-fabric-v1-0/