

Choaxing Luo's dissertation

by Choaxing Luo

Submission date: 07-Sep-2022 08:38AM (UTC+0100)

Submission ID: 185765203

File name: 32573_Choaxing_Luo_Choaxing_Luos_dissertation_430701_1416847429.docx (2.25M)

Word count: 15563

Character count: 84799



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

A Study of Blockchain based Vehicle Management System

by

CHAOXING LUO

Supervisor: TIM BASHFORD

Project submitted as part of the requirements for
the award of MSc

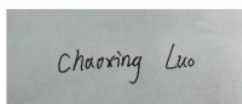
September 2022

DECLARATION

I, CHAOXING LUO declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed :

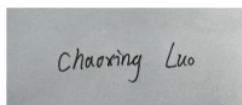
A rectangular box containing a handwritten signature in cursive script that reads "Chaoring Luo".

Date : 07/09/2022

FORM OF CONSENT

I CHAOXING LUO hereby consent that my Project, submitted in candidature for the M.Sc. Applied Computing degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed :

A rectangular box containing a handwritten signature in cursive script that reads "Chaoring Luo".

Date : .07/09/2022

Table of contents

ABSTRACT	5
1. INTRODUCTION	5
1.1 BACKGROUND	5
1.2 AIM	6
1.3 OBJECTIVES	6
1.4 PROJECT STRUCTURE	6
2. LITERATURE REVIEW	7
2.1 BLOCKCHAIN TECHNOLOGY	7
2.2 HYPERLEDGER FABRIC	11
2.3 APPLICATION OF BLOCKCHAIN TECHNOLOGY IN VARIOUS INDUSTRIES	12
3. METHODOLOGY	18
3.1 RESEARCH METHODOLOGY	18
3.2 DEVELOPMENT METHODOLOGY	18
3.2.1 Feature driven Development	18
3.2.2 FDD for blockchain vehicle information management system	19
4. BLOCKCHAIN SYSTEM DESIGN AND IMPLEMENTATION	21
4.1 HYPERLEDGER FABRIC	21
4.1.1 Distributed ledger structure	22
4.1.2 Consensus Mechanism	23
4.2 PRODUCTION NETWORK DEPLOYMENT	24
4.3 CHAINCODE INSTALLATION AND DEPLOYMENT	27
4.3.1 Lifecycle of Chiancode	30
4.3.2 Problems and Solutions	31
4.4 BLOCKCHAIN EXPLORER	31
5. RESULTS	34
5.1 BLOCKCHAIN NETWORK AND CHAINCODE IMPLEMENTATION	34
5.2 BLOCK EXPLORER IMPLEMENTATION	35
6. DISCUSSION	37
7. CONCLUSION	38
8 REFLECTION AND FUTURE WORK	38
8.1 REFLECTION	38
8.2 FUTURE WORK	39
REFERENCE	39

Table of figures

FIGURE 1 THE PROCESS OF FEATURE-DRIVEN DEVELOPMENT	18
FIGURE 2 THE PRINCIPLE FOR HOW SMART CONTRACTS WORK IN HYPERLEDGER FABRIC	20
FIGURE 3 THE PROCESS OF BUILDING BLOCKCHAIN VEHICLE MANAGEMENT SYSTEM	21
FIGURE 4 THE STRUCTURE OF BLOCKS IN FABRIC	23
FIGURE 5 THE PROCESS OF CONSENSUS IN HYPERLEDGER FABRIC	24
FIGURE 6 CONFIGURATION OF CA DIRECTORIES	25
FIGURE 7 THE CONFIGURATION OF DOCKER-COMPOSE.YAML	27
FIGURE 8 SUCCESSFUL DEPLOYMENT OF PEER NODES	27
FIGURE 9 THE Go CODE OF THE SMART CONTRACT	28
FIGURE 10 THE Go CODE OF INITLEDGER	29
FIGURE 11 THE Go CODE OF CREATECAR	29
FIGURE 12 THE Go CODE OF QUERYCAR	29
FIGURE 13 THE Go CODE OF CHANGECAROWNER	30
FIGURE 14 PACKAGING THE CHAINCODE	30
FIGURE 15 INSTALLATION CHAINCODE ON PEER	30
FIGURE 16 SUCCESS RESPOND AFTER INSTALLATION	30
FIGURE 17 APPROVING THE CHAINCODE ON THE CHANNEL	31
FIGURE 18 COMMITMENT OF THE CHAINCODE	31
FIGURE 19 CONFIGURATION OF ORGANIZATIONS IN THE BLOCK EXPLORER	32
FIGURE 20 CONFIGURATION OF THE BLOCK EXPLORER	33
FIGURE 21 CONFIGURATION OF ORG1 IN THE BLOCK EXPLORER	34
FIGURE 22 CONFIGURATION OF ORG2 IN THE BLOCK EXPLORER	34
FIGURE 23 SUCCESSFUL INVOKING OF QUERYCAR AND QUERYALLCARS	34
FIGURE 24 CREATECAR	35
FIGURE 25 SUCCESSFUL INVOKE OF THE FUNCTION	35
FIGURE 26 SUCCESSFUL TRANSFERRING THE OWNERSHIP	35
FIGURE 27 THE LOGIN PAGE OF THE EXPLORER	35
FIGURE 28 THE DASHBOARD OF THE EXPLORER	36
FIGURE 29 NETWORK ORGANIZATIONS	36
FIGURE 30 CHANNEL	36

Abstract

Blockchain has earned its dominance in the field of trending technology and enjoyed a rising popularity. As a matter of fact, dependent on different types of blockchain, the use cases are not just limited to the field of cryptocurrency. Blockchain has attracted attentions from all over the world, with increasing number of researches on it. This project aims to utilize a specific kind of blockchain, consortium blockchain, to build a vehicle management system based on the blockchain platform, Hyperledger Fabric. Also, to visualize the blockchain network and monitor its status, an auxiliary block explorer will also be designed and implemented. As a result, this project succeeded in using fabric to provide a secure, transparent and immutable system to store, share and exchange car information. Besides, the details of network deployment and chaincode deployment are illustrated. This project can be used a starting point for a simple exploration of blockchain technology application in the field of automotive services related areas.

1. Introduction

1.1 Background

Distributed database is not a new concept. Born in 2009, blockchain has ridden on its wave and become the most popular concept of the 21th century. It has attracted researchers from both academic and business. People all over the world cannot stop talking about it, right after the boom of the Bitcoin and Ethereum. It might sound so intense and pressure when people are using terms like hash value, POW or difficulty. However, it is not rocket science. To understand the core value and theory behind it is not hard at all.

Blockchain is a node-based distributed database system. It has the qualities of being unchangeable and unforgeable, and it also functions as a ledger system. A copy of the whole Bitcoin blockchain, which records each transaction involving its token, is a key component of Bitcoin. We can determine the value of any address at any time in history using this information.

A chain of data blocks created using cryptographic techniques makes up a blockchain. From the genesis block until the current area, each block carries the hash value of the previous block. constructing a blockchain. In order for the hash value of the preceding block to be known, each block must be generated chronologically after the one before it. Double spending is exceedingly challenging in Bitcoin due to these features. The fundamental innovation behind Bitcoin is blockchain.

Blockchain is a new type of decentralized protocol that can securely store Bitcoin transactions or other data. Information cannot be forged or tampered with, and smart contracts can be automatically executed without any centralized agency review. Transactions can be digital currencies such as Bitcoin, or digital assets such as debt, equity, copyright, etc. Blockchain technology solves the Byzantine general problem, greatly reduces the cost of trust and accounting in the real economy, and redefines the

Internet era, property rights system.

In this project, a consortium blockchain platform will be utilized to design and build a blockchain network. Consortium blockchain refers to a blockchain whose consensus process is controlled by pre-selected nodes; for example, 15 financial institutions form a community, each of which runs a node, and in order to make each block valid, it needs to obtain one of the 10 agency confirmations. The blockchain may be readable by everyone, or limited to participants, or take a hybrid route, such as the root hash of the block and its API (application programming interface), which can be used by the outside world. to make a limited number of queries and obtain information about the state of the blockchain. These blockchains can be considered "partially decentralized". To demonstrate the process of configuration and the feasibility of such system to manage car information, this project can be considered as a simple build of the system.

1.2 Aim

The aim of this project is to understand the theory behind blockchain and build a blockchain network to manage car information by using the consortium blockchain platform.

1.3 Objectives

To achieve the aim of this project, several objectives is to be completed to measure progression. They are as follows:

- 1) Conduct a comprehensive literature review into numerous areas related to the topic of this project, including blockchain technology, hyperledger fabric and automotive industry.
- 2) Determine a software development methodology, that will be used to produce accurate result during experimentation
- 3) Build the blockchain system according to the step
- 4) Deploy the smart contract written in Go language.
- 5) Simulate implementations for the blockchain network.
- 6) Analyze the result.

1.4 Project Structure

The structure of this project is essential to ensure that every objective is achieved. This project will be split into 8 chapters to clarify ideas and implementations. This will also help readers understand the ideas and concepts illustrated in the project. The structure is as follows:

Chapter 1: Introduction about the project, mainly about the background, aims and objectives.

Chapter 2: a review of the current literature around the subject of this project.

Chapter 3: An explanation of the way the author intends to conduct the research and experimentation whilst underlining an overall methodology for the project.

Chapter 4: the whole design and building process of the blockchain network, and the configuration and design of the auxiliary block explorer.

Chapter 5: an explanation of the implementation results of the blockchain network.

Chapter 6: after the deployment of the network and chaincode on the platform, a discussion part is added to analyze the pros and cons of Fabric platform.

Chapter 7: conclusions are drawn in this chapter. Also, this chapter will summarize the work that have been done in this project and

Chapter 8: the final will elaborate the reflection on the project and point out which area could be improved if the project was to be repeated or expanded upon

2. Literature review

2.1 Blockchain technology

Blockchain, as we all know, was first created by Nakamoto Satoshi[1]. It was created as a response to the trust crisis that swept the world in the wake of the 2008 financial crisis. Years later, a young man named Vitalik made some improvements and designed a new structure for the development of cryptocurrency, Ethereum[2]. By introducing the application of smart contracts, the blockchain was no longer limited in the field of cryptocurrency. Blockchain technology is now of great interest to and being studied by researchers around the world. It can provide solutions in different industries such as automotive, healthcare and insurance, due to its immutability, tamper proofing and transparency.

Smart contract is another concept that can reshape traditional industry and business process. It is embedded into blockchain after the invention of Ethereum, which enables automatic execution of contractual terms without a third party's involvement. Consequently, according to Zheng et al[3], smart contracts can reduce costs and improve efficiency. By conducting a survey on smart contracts, they discussed the technical advancements of smart contracts as well as their difficulties. Also, several platforms were categorized and compared with some representative examples. In this paper, they claimed that there are 4 phases of challenges in smart contracts. And to remedy these challenges, it was pointed out that integration of artificial intelligence and blockchain is worth a try. But they did not conduct any empirical research according to the paper.

While, Fillip, Mannan and Reijers[4] argued that instead of a trustless technology, blockchain should be seen as a confidence machine. In their paper, they took a deep analysis on the definition of trust and confidence, with the relationship between these two concepts. It was claimed that it is actually those cryptographic rules, mathematics, and game-theoretical incentives in a blockchain system that increase confidence in a computational system. They summarized that blockchain based systems are not only made up of code, but also of a larger variety of participants, such as miners, programmers, validators, token holders, and end users, etc. Therefore, they argued that having confidence ultimately means that trusting the whole assemblage of actors associated with that network.

Jaoude and Saade[5] summarized blockchain usage in different domains. They conducted a systematic literature review of the blockchain technology, tracking its increase in popularity and analyzing the major fields of study and areas of application

for which blockchain offers a valuable solution. Through the investigation, it was discovered that blockchain's asymmetric encryption, anonymity, decentralization, and immutability can offer significant advantages to many other sectors and topics. Beyond that, in areas, such as healthcare, Internet of Things, energy and finance, they stand to benefit disproportionately from the application of blockchain technology. It is safe to say that certain characteristics hardwired with the invention of blockchain has made it a valuable tool for many industries.

Zyskind et al[6], researchers from MIT and Tel-Aviv, brought up that blockchain can be used to protect personal data for the sake of privacy. In their paper, they described a decentralized personal data management system. Several protocols were implemented to transform a blockchain into an automated access-control manager that were used to store, query and share data, without involvement of a third party making the management system susceptible to malicious attacks and misuses.

To better understand the performance metrics of a blockchain system, Dinh et al [7] presented a benchmarking framework for assessing performance of private blockchains, BLOCKBENCH. Unlike public blockchain where everyone can have access to the blockchain as long as they are registered, in a private one, all the nodes need to be authenticated and authorized to do so. For the BLOCKBENCH examination, three significant blockchain platforms—Ethereum, Parity, and Hyperledger Fabric—were chosen. The findings highlighted several trade-offs in the design space as well as notable performance variations between blockchain and database systems. A hope to bring blockchain closer to the field of databases was brought up, implying their future research direction.

To improve blockchain's performance and tackle the hinderance caused by growth of the blockchain size, Xu and Huang[8] proposed a segment blockchain mechanism, an approach that can split whole blockchain into segments with nodes only storing a small copy of one segment. Their results demonstrated that t segment blockchain largely reduced the data requirement compared to the original blockchain designed by Nakamoto[1]. It is fair to say using segment blockchain to power data-heavy blockchains can be of more advantage.

Mitani and Otsuka[9] approached the blockchain technology from the perspective of mathematic and cryptography. They have achieved privacy protection and high transparency in a permission blockchain. To improve the transparency of the permissioned blockchain under privacy protection, they set the focus of their research on the traceability of the permissioned blockchain from three properties: trade privacy, preservation and noninvolvement. They picked hidden Markov model as their traceability model and encrypted the model with homomorphic encryption. Their research showed that the protocol that can verify the knowledge that the plaintext is equal to zero by using the noninteractive zero-knowledge proof. And it is worth noting that they are the first to achieve both preservation and noninvolvement while protecting the privacy of transactions.

However, it is not guaranteed that blockchain is the safest way to protect all kinds of data and unsusceptible to any attacks. There are still some vulnerabilities and threats to blockchain. From the cryptocurrency to smart contract, blockchain has been used in

numerous sectors.

Pal et al[10] proposed a group key management scheme to level up the security over sensitive records over the Blockchain network. First of all, they had an overview of Blockchain, analysis of existing PKI for Blockchain with key management for Blockchain wallet discussed. Then, to increase security level for confidential sensitive records stored in the blockchain, the group key Management scheme was proposed. After conducting the security analysis, it was concluded that the proposed scheme is secure and efficient. But, there are still something lacked in the research. There are no detailed models built on any environment and also no experiments conducted to get the exact result of the proposed scheme to confirm its effectiveness.

Li et al [11], noticed that there was a research gap that a systematic review about the security of blockchain systems was lacked. Li et al, noticed that there was a research gap that a systematic review about the security of blockchain systems was lacked. Therefore, they conducted a survey on the security threats to blockchain and Therefore, they conducted a survey on the security threats to blockchain and examined the popular blockchain systems such as Ethereum, Bitcoin and Monero etc. For each risk or vulnerability, they came up with its consequences and causes. Even more, they took some of the real attacks that had happened over the years as examples to analyze susceptibilities exploited by hackers so that blockchain systems can be unsusceptible to these threats by fixing the problems.

Kim et al[12], set their focus on the security matters of blockchain. They adopted an innovative thinking skill which has been used to develop new prototype from the scratch. A security perspective was provided to avoid the sybil attack against blockchain networks by using a hybrid stochastic model. Also, new IoT networks were considered to build the new framework to support blockchain network based on the IoT technology automotive services. However, this paper did not demonstrate the details of the blockchain network and the stochastic model, but only serve as a guideline for the direction of future research.

Zhou et al[13] aim to solve the scalability problems in blockchain systems. Consequently, several scaling solutions for blockchain were presented and classified by level. Existing popular solutions can be categorized into three layers: layer 1, layer 2 and layer 0. Layer 1 focuses on consensus, network and data structure of blockchain, which can also be comprehended as on-chain solutions because they are all executed inside the blockchain. Layer 2 turns to off chain methods to scale out blockchain, such as off-chain channel, side chain, and cross-chain protocols. Besides, On the basis of the drawbacks discovered, they also outlined a number of potential research topics and remaining concerns, such as the enormous volume of blockchain data that has to be compressed or pruned.

Another interesting survey[14] conducted by Tas and Tanrıöver investigated the application of blockchain in the field of E-voting. It provides an overview of blockchain based e-voting systems. First, an introduction to the basic structure and features of blockchain in relation to e-voting was given through a systematic view approach. Then, a conceptual description of the desired e-voting application was provided. As a result of this research, it was determined that blockchain systems can provide solutions to

certain problems that prevail in current election systems. Although, the authors hold that our perception about blockchain technology is still not full and implementing blockchain voting practices can bring unknown security risks and vulnerabilities, which is why applying the e voting systems to a few small pilot regions first would be more sensible and its scope could be extended based on further analysis. Their research provides a good strategy and posed a reasonable attitude towards such novel technology.

Zheng et al[15], observed that it is usually costly and difficult for most common developers and teams to build and maintain a blockchain network due to its complexity, which consequently cannot guarantee the reliability and security of the system. Based on this, Zheng and his team proposed a Blockchain-as-a-Service Platform, called NutBaas. It offers blockchain services across cloud computing environments, including network deployment and system monitoring, smart contracts analysis, and testing. With the help of such services, developers may concentrate on writing codes to practice their business logic and better integrate blockchain technology into their business models. Beyond this, other advanced services are also introduced, such as Identity chain and smart contracts security vulnerability detection.

Muzammal et al[16] integrated blockchain with traditional distributed databases, presenting an application platform that gains features from both decentralized distributed database and centralized distributed database. It was named CHAINSQL, an open-source system, featuring a tamper-resistant and consistent multi-active database with the ability to recover backup after data-level disaster. The effectiveness of the system is demonstrated with the result of a comprehensive experimental evaluation.

The goal of a proper tracking system is to reduce poor quality goods and unreliable distribution using proper labeling and accurate tracking. Thus, based on its immutable, and tamper-proof characteristic, blockchain technology has been widely experimented and applied in healthcare and supply chain industry to build tracking systems.

Applications such as BitHealth and MedRec[17] are notable blockchain based applications to support the healthcare industry. Unlike public chains which allow any user to join the blockchain and have access to all the information stored in the blocks, nodes in private chains need to apply for permission, certificate, before gaining access to the network and consensus. Therefore, private chain is more suitable choices instead of public chains in terms of data privacy and security.

Hussien et al[18] discovered that the growth in the use of blockchain in the healthcare is incredible and can put great impact on the industry. In their research, they addressed the gap between the healthcare industry and blockchain technology, to identify the trend of blockchain technology in healthcare. To offer a concise description of the research that has been done thus far, bibliometric analysis of dataset distribution, venues, keywords, and citations was carried out. The security and privacy of case studies including telecare medical information systems and E-health were also investigated. Overall, this paper covered the possible drawbacks and opportunities of using blockchain technology in the healthcare sector and emphasized the driving forces behind academic and research endeavors.

A blockchain based decentralized reputation system was proposed by Zhou to

address issues as fraud and forgery in e-commerce industry[19]. By designing and deploying smart contracts to achieve the incentive mechanism, it helps form a virtuous circle in online shopping. Due to the immutability of blockchain, it is difficult for attackers to tamper the product information and reputation data in the proposed system, which also resists unfair ratings and collusions.

Blockchain platforms can use distributed ledger technology to create a digital system and database to track transactions across the supply chain. Focusing on new blockchain-based supply chain management, Park and Li[20] examined how well it performed in terms of social equality, environmental preservation, and governance efficiency. A thorough literature analysis and two case studies using blockchain technology were used to evaluate if the three sustainability indicators may be improved in some way along supply chains. And it was determined that blockchain technology has the potential to enhance the performance of supply chains' sustainability. An empirical study that quantifies the impacts of blockchain technology, however, may still hold promise for the future advancement of the technology.

It is no doubt that there are different types of blockchain based on its accessibility and consensus mechanism. Overall, blockchain can be fundamentally divided into three types, namely public chain, private chain, and consortium chain. A public chain network is open to anyone who can participate without restrictions. Most types of cryptocurrencies run on a public blockchain that is governed by rules or consensus algorithms. However, a private, or permissioned, blockchain allows organizations to set controls on who can access blockchain data. Only users who are granted permission can access specific sets of data. For example, Oracle Blockchain Platform is a permissioned blockchain. Finally, what distinguishes a consortium chain from the former two kinds of blockchain lies in the consensus mechanism. For public or private chain, they usually adopt Proof of Work consensus algorithm, which requires involvement of all the nodes in the network, while the consensus process is closely controlled by a preselected set of nodes or by a preselected number of stakeholders in a consortium chain.

2.2 Hyperledger Fabric

First introduced by IBM, Hyperledger Fabric is an open source, permissioned blockchain that has already been adopted in various industries to store and manage information in a decentralized way.

Unlike the power intensive nature in Proof of Work consensus mechanism in which miners are competing every minute to gain reward, causing tremendous amount of computing resources consumption, Hyperledger Fabric takes a consortium consensus mechanism in which the process of consensus can be determined and completed by only limited numbers of nodes, saving a lot of computation resources and energy for users and organizations engaged, also increasing the efficiency of the consensus process.

Nasir et al[21] compared the performance between two different versions of Hyperledger Fabric. First released in its 0.6 version, and then upgraded into v1.0 in 2017, there were no accurate methodology for evaluating and assessing the different blockchain platforms back then. In order to fill this gap, they first conducted a

performance analysis of the two versions of Hyperledger Fabric. Metrics as execution time, latency, and throughput were used to assess performance of the two platforms by increasing the number of transactions in each platform up to 10000. By altering the number of nodes in each platform by up to 20, another study was done to assess how scalable the two systems were. Overall, the performance analysis findings show that Hyperledger Fabric v1.0 surpasses Hyperledger Fabric v0.6 in all assessment measures, including scalability, throughput, execution time, and latency. The performance of the Hyperledger Fabric v1.0 platform, however, fell short of contemporary conventional database systems under heavy workload conditions.

Yan et al [22], to improve the efficiency and data security of transaction handling in Hyperledger Fabric, approached to the problems with 2 ideas: conditional queries and data storage and proposed novel solutions as follows. Firstly, Composite keys are created with all attributes of the query before it is executed to improve the performance of conditional queries. Then, an index named AUP is built in a second method. The extensive experiments conducted on the real-world dataset demonstrate that the proposed methods can achieve high performance in terms of efficiency and memory cost. The traditional key-value method is used as the baseline method to be compared with the other two methods proposed by them, namely composite key based method (CCK) and AUP index based method (AIM). The results revealed that both CCK and AIM easily outperformed the baseline.

Uddin et al[23] identified a sensitive issue in the healthcare industry, which is that, Providing adequate security, trustworthy access control, and resolving privacy and secrecy difficulties and obstacles in present hospital infrastructures are generally areas where the existing EHR systems fall short. They developed a Blockchain-enabled Hyperledger Fabric Architecture for several EHR systems to address this issue. Channels can be set up in the system they develop to permit safe and confidential communication between various ledger network parties. To carry out various EHR transactions over the network, several chaincodes were developed and put into use. A EHR management system's interoperability, scalability, and availability are all guaranteed by the planned network.

2.3 Application of blockchain technology in various industries

For real-time food tracing, Tian [24] suggested a food supply chain traceability system. The Internet of Things and blockchain technologies are combined. Access to the platform is open, transparent, dependable, and secure for all members of the network who have the appropriate permissions. Additionally, a novel idea called BigchainDB was put up to address the issues brought on by the fundamental flaw in poor scalability of blockchain technology. The technology has the potential to considerably increase the efficiency and openness of the food supply chain, which would certainly improve food safety and restore public faith in the food sector.

Kuo et al [25], presented a systematic review on blockchain technology and its benefits, pitfalls and latest applications to the biomedical and health domains. In total, 5 advantages of blockchain for biomedical and healthcare applications were extracted when compared with traditional distributed databases, which are decentralized

management, immutable audit trail, data provenance, robustness/availability, and security/privacy. They came to a similar conclusion that blockchain distributed ledger technology has the potential to advance the biomedical and health care domains in various novel ways. Even though there might be some challenges, such as scalability or sybil attack, they can still be addressed through careful design and implementation.

Antwi et al [26], also investigated the use of private blockchain technologies in healthcare application to assess their feasibility through different tests. After applying Hyperledger Fabric to 5 different scenarios, it was concluded that there is a trade-off between transparency and confidentiality that developers and governance departments should be aware of when developing blockchain applications for healthcare industry.

Uddin [27] proposed a novel track and trace blockchain system that leveraged Hyperledger Fabric permission blockchain platform with smart contracts by using chaincode. The proposed Medledger system enables pharmaceutical stakeholders to access the blockchain network with permission granted, diminishing the need for a centralized authority. Potential risks of tampering with confidential information of patients and drugs can also be mitigated by introducing a more robust and attack-resistant network.

Fan and Zhang[28] proposed a smart grid data aggregation and regulation mechanism based on consortium blockchain. The signcryption algorithm was applied to encrypt and decrypt multidimensional data. Additionally, smart contracts were set up so that grid managers could get feedback on how users were regulating their power usage. The suggested approach benefited from lower processing and communication costs through security analysis and performance comparison.

Toyoda et al [29], to identify easily cloned counterfeits which could be hardly recognized in the process of radio frequency identification (RFID) technology, proposed a blockchain-based product ownership management system, which enables each party, to transfer and prove the ownership of RFID tag-attached products. Performance evaluation results have shown that the cost for managing products with the proposed system is less than 1 US dollar when the number of owner transfers is less than or equal to six. Their results showed that the proposed system based on blockchain can be cost-efficient and effective dealing with such situation.

The automobile industry's procedures may alter as a result of blockchain technology. The use of blockchain technology is becoming more commonplace across several sectors. It seems sense that blockchain would enter the sector in its early phases of growth given the exponential velocity at which the automotive landscape is growing. In the automotive industry, the use cases and purposes are quite similar as mentioned above, to guarantee data security, traceability and provide tamper-proofing. Most applications are revolving areas like used car trading, supply chain management and car service management etc.

Reddy et al [30], reported a systematic literature review on blockchain applications in automotive supply chain, intending to verify that blockchain can be utilized to address issues in the automotive supply chain, namely Volatile environment, Uncertain schedules & information; Complex supply chain networks, and Ambiguous decisions. They are figuratively abbreviated as VUCA world by the authors and ASC for

automotive supply chain. By the method of content analysis, their results implied that exploration on linking the supply chain visibility and transparency with blockchain can be a efficient and resilient solution for a ASC in VUCA world. Furthermore, a blockchain implementation framework was proposed for ASC, to provide a decision-making approach for practitioners in VUCA world. This study makes a significant contribution towards the research gaps identified in the literature, providing useful information on investigating the scope for BCT in the ASC. The review addressed status of the automotive industry towards the digital era, and the implementation of BCT in ASC is addressed in the RQ1.

The blockchain system's underlying security principles were analyzed by Fraga-Lamas et al [31], who also examined the blockchain's enormous potential for use in the automobile sector while highlighting its cybersecurity aspects. By evaluating the state-of-the-art and developing the current difficulties facing the major stakeholders, they evaluate the applicability of blockchain to help organizations and entrepreneurs determine the need of blockchain technologies. Also, the article provides recommendations after SWOT analysis with aim of guiding researchers and companies in future industry developments. Overall, this article summarized relevant stakeholders and their confronting challenges. In addition, they presented a holistic approach to a blockchain based advanced automotive industry with a review of the main use cases and the optimization strategies for designing and deploying these applications.

Yoo and Ahn[32] proposed a used car transaction management system based on Ethereum. The introduction of smart contracts guaranteed no involvement of third-parties. This system eliminates the asymmetry of information between buyers and sellers through features of blockchain, which also reduces and averts brokerage commissions. Therefore, transparency of the whole process can get enhanced. To be more precise, this system returns the hash function provided in the smart contract and leverages IPFS to store data using the Mongo database running on a Node.js server. This study shows how to solve the existing information asymmetry problem by using blockchain technology. What it implies is that it cannot only be applied to manage used car data but also overall car data of similar kind.

Sharma et al[33], to address issues like disruptions to government regulations, insurance services and maintenance services, which can be caused by increased adoption of auto-pilot cars, proposed a blockchain-based distributed framework for the automotive industry in the smart city. They posed a novel algorithm to select miner node in the blockchain network. The suggested model was evaluated using a dataset of mined blocks from litecoinpool.org that was acquired in order to assess the viability of the proposed framework on a private blockchain platform. The outcomes demonstrated the suggested model's proof-of-concept, which may be used to a broad variety of future smart applications.

Xu et al[34], used collective case study method to investigate the application of blockchain technology in German OEMs that solve specific problems in the current automotive supply chain. Through in-depth interviews, they observed the technology adoption of blockchain and identified the potential and barriers of blockchain adoption in detail. And it was concluded that blockchain technology has advantages in real-time

information sharing, reducing enterprise costs and improving product quality, etc.

Miehle et al[35] devised a decentralized traceability application called PartChain for the supply chain management in the production stage of automobiles. In this blockchain network system, the representation of a physical part can be traced through the whole process from its creation to transportation and assembly, monitoring and sharing of the data across the blockchain network. PartChain provides a cost-efficient possibility to establish a supply chain traceability system with a low barrier to entry, which is particularly important for small- and medium-sized companies. Also, a case study was conducted to demonstrate the feasibility and potential of the concept.

To tackle challenges such as vehicle information tampering, mileage falsification and poor traceability in the automotive market of Mauritius, Luchoomun, et al[36] proposed a proof of concept blockchain system to manage car importations, sells and registrations. The examination of the application highlighted the fact that they were able to increase transparency, enhance traceability, and security, reduce costs and increase efficiency in the traditional system, but they did not explain the details of the method.

Kuhn et al[37] first proposed a traceability architecture based on blockchain technology with the aim of the aspired level of transparency to reduce security vulnerabilities in the manufacturing process of electronic parts for cars. Their experiment tests showed that the proposed blockchain based traceability solution works well in storing and protecting the production history of complex product configurations. Their system was built on the consortium Ethereum network with proof of authority consensus deployed, which is similar to Hyperledger Fabric architecture that each node needs to get permission before gaining access to the whole network.

Lamberti et al[38] believed that through combination of blockchains and sensors installed on a vehicle, it could help activate insurance coverage the moment accidents happen. They built a prototype that could be running on mobile devices as an application and also a portable electronic device to be installed onboard. Drivers can change the status of insurance coverage through the app and every modification would be stored on a smart contract in the blockchain so that the status is resistant to tamper with and easy to track for each vehicle. What's more, the passengers' data and the vehicle's data could be collected by the sensors installed. Because the consumer may directly regulate the activation/deactivation of a coverage by engaging with the smart contract, the suggested approach might reduce the expenses associated with policy updates. Since the system will record the vehicle's status at each change, fraud may also be decreased.

Fraud in automobile odometers has become a billion dollar problem worldwide. Abbade et al[39] proposed a blockchain architecture for vehicular odometers. A different consensus algorithm, the Practical Byzantine Fault Tolerance algorithm, was adopted than traditional ones, making the system more suitable for vehicular networks. From the obtained results, it is more promising than the proof of work consensus in such use case scenario. Nevertheless, a more thorough evaluation needs to be performed to further reveal its scalability, cost and speed advantages of the proposed PBFT algorithm.

Another integration between vehicular networks and blockchain technology was

conducted by Ortega et al[40]. They proposed a network combining content-centric network and permissioned blockchains that enable the dynamic control of source reliability, and integrity and validity of the information exchanged. Their research concluded that with the combination of blockchain technology and 5G vehicular network, they will easily form a solution to the challenges in the field of fully autonomous driving without adding other costs.

Wang et al[41] outlined a blockchain-based PSS framework, including several stakeholders and application scenarios. By increasing data immutability and transparency using blockchain technology, mutual trust among stakeholders can be maintained effectively. However, they did not demonstrate the detailed network of blockchain, but only descriptions of the work flow in the framework.

Nizamuddin and Abugabah[42] focus their research on the insurance section of automobile industry. To battle against challenges that the industry is facing such as lack of trust, mismanagement of insurance records, pricing problems, they came up with a decentralized blockchain-based framework for the auto insurance sector. It can regulate claims-related actions for automotive insurance and automate payments. Only after the conclusion of a transaction is accepted by all participating miner nodes is it recorded in the blockchain. Similarly, there are still some challenges existing for the implementation of such framework. For example, the scalability issues from blockchain slow down the system with the increasing number of transactions, eventually becoming unsustainable. There is also a lack of government standards, making it hard to establish definite regulations and laws, especially more difficult for a blockchain system which are multiple-parties involved and even located across the world.

Patro et al[43] proposed a blockchain approach to overcome problems in the product recall management of automotive industry. Four smart contracts were developed to automate the process of vehicle recall and repair, record events data and maintain data provenance. Furthermore, by cost analysis, security analysis and comparing the proposed method with non-blockchain-based method, it was proved that the former was cost-efficient and secure when dealing with malicious attacks. However, it is still unknown if the smart contracts can work really well on the real Ethereum network.

Lu and his team[44] built a blockchain-based counterfeiting prevention system for automotive supply chains. In this system, Hyperledger Fabric was utilized to restrict the permissioned parties to gain access to the blockchain network. The benefits and challenges of blockchain technology application in supply chain were studied in this work. Lessons were summarized through the experiment process so that they can also be applied to other cases. Finally, the performance of Hyperledger Fabric was assessed and a solution to improve its privacy issue were proposed, which is creating multiple endorsing nodes with multiple chaincodes.

Chen et al[45] proposed a blockchain-based vehicle management service system, enabling instant uploading of vehicle history records by vehicle owners, vehicle authorities, mechanic shops, insurance brokers, and other connected people and organizations. Nodes that are registered to join the blockchain network can have access to requests for the data about vehicle information such as ownership history information,

maintenance history information and insurance claims.

Dorri et al[46] aimed to strengthen the private intensity of users and increase the security of the vehicular ecosystem, thus proposing a blockchain based structure. Emerging services like wireless remote software updates and dynamic insurance fees are included in the architecture to demonstrate its effectiveness. The privacy of users is ensured by innovating the asymmetric encryption with changeable public keys. The architecture can be applied to several applications, including remote software updates, insurance, electric vehicles and smart charging services and car sharing services. However, like the drawbacks of [12], this paper only proposed a framework of blockchain, instead of the implementation of the system with its experimentation on testing networks of real blockchain networks.

He et al[47] drew their inspiration from bitcoin, proposing a blockchain-based truthful incentive mechanism for decentralized P2P applications. In order to incentivize users to cooperate with each other, those who help with a successful transaction will get re rewarded in this mechanism. Furthermore, a secure validation method and a novel pricing strategy are integrated into the mechanism to prevent users and miners from going against or colluding with one another. By conducting a game theoretical analysis and evaluation study, it can ensure the effectiveness and security strength of the proposed mechanism. In the incentive mechanism, miners verify transactions by using commutative encryptions. Of course, one of the contradictions of this mechanism is between incentive and privacy. Another thing that this paper did not research is the situation of collusion of the sender and the receiver. The author pointed out a possible solution which is to introduce reputation into the incentive mechanism, while whether it would work is still pending.

Similarly, Turkanović et al[48], based on the concept of the European Credit Transfer, and Accumulation System, proposed a global blockchain-based higher education credit platform. Students and higher education institutions can find globally unified viewpoint on this platform, as well as other potential stakeholders, such as commercial companies and governance departments. ECTX tokens, issued by the blockchain system, were used to represent the credit points that student can gain for completed courses, and all the tokens shall be managed, controlled and processed by the platform. Students gain by having a single, clear picture of their completed coursework, while HEIs profit from having access to the most recent information, regardless of the student's educational background. However, the platform could get further simplified on how students can get tokens. For example, course addresses could be created so that students can get tokens directly from them once they finish their course instead of from institution. In this way, the platform can run more efficiently and save much more resource and time for the staff of institution.

Iftekhhar et al[49], To show access control and provide the foundation of trust for IoT devices, coupled the Hyperledger Fabric with IoT devices. For Hyperledger Fabric components gaining access to the IoT device, an attribute-based access control (ABAC) mechanism has to be created. Also, in this study, the Raspberry Pi 4 Model B based on ARM64 architecture, a type of single board computers which are increasingly powerful and popular in automation applications, is used as the IoT device. The entire system

utilized Hyperledger Fabric to segregate people and devices using rules and programmatic access management built into the chaincode itself. The study implied that Hyperledger Fabric can work well with tools to manage IoT devices on the blockchain, which can really open up a new door for technology combination between IoT and Blockchain in the automotive industry

This project is to apply blockchain technology to the automotive industry. Based on the Hyperledger fabric platform, a vehicle management blockchain system will be proposed to store and manage vehicle information.

3. Methodology

3.1 Research methodology

As written above, the research method in this dissertation is literature review. Starting from the basics of blockchain, the reason of its birth and theories behind blockchain are elaborated. Besides, the development and evolution of this technology is also presented. Therefore, to provide insights and applications for blockchain to be utilized in other industries than just cryptocurrency, latest literatures from across industries are the best to study and learn from to have a firm grasp of the latest trends on the application of blockchain. Due to some of the characteristics of the blockchain technology, at the moment it is best applied in areas of transparency and data security. Then, different types of blockchain are also introduced to fit in different contexts. A little introduction about Hyperledger Fabric is also revealing some of its characteristics, such as permissioned access and more efficient consensus mechanism. Finally, I study many blockchain applications in various industries, such as healthcare, e-commerce and the most important one, automotive, which is also the industry this dissertation sets focus on. Extracting inspirations and perspectives from applications in various areas, the blockchain vehicle information management system is finally proposed and built to test its functionality.

3.2 Development methodology

3.2.1 Feature driven Development

Feature driven Development (FDD), is an agile, highly adaptive software development process. It was first introduced in 1997, when Jeff De Luca was the project Manager of a large software development project in Singapore. He devised such development method to meet the needs of a 15 months, 50 persons software development project for a Singapore bank back then. There have been several implementations of FDD since its successful use on the Singapore project. The whole process of FDD is as shown in Figure 1, consisting of five basic activities.

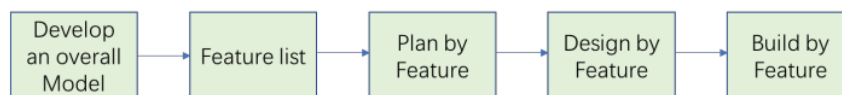


Figure 1 the process of Feature-Driven Development
An FDD project starts with a high-level general overview of the scope of the

system that will be designed. Detailed domain models are created in small groups. In each domain area, one or more of the proposed models can be selected. Then, models in all the domain areas are progressively merged into the overall model.

Then, after the initial modeling stage, certain knowledge could be gathered to determine a feature list of the overall model. By dividing the domain into subject areas, they would contain business activity to form the basis for a feature list. Features in such context are small pieces of client-valued functions expressed in the form "<action> <result> <object>". An example of features is: Calculate[action] the total purchase[result] by a customer[object]. Features should not take more than two weeks to complete, else they should be broken down into smaller pieces. After the feature list is completed, a plan for development has to be designed to assign ownership of features as classes to developers who is responsible for the ultimate contents and results of the codes

A design package is what is coming next for each feature. Usually, a small group of features that are to be developed within 2 weeks will be selected by the chief programmer. The corresponding class owners shall work with the chief programmer to establish a detailed sequence diagram for each feature development and refine the overall model. Next, the class and method prologues are written and finally a design inspection is held.

After a successful design inspection for every activity to achieve the expected feature is planned, the class owners will code for their classes. the completed feature should be promoted to the main build when unit testing and code inspection are conducted successfully.

3.2.2 FDD for blockchain vehicle information management system

In this project, feature-driven development is best suitable for building such an information management system. As an iterative and incremental software development process, it is agile and lightweight for developing software. The structure of building the blockchain system in this project is shown as figure 2. Construction of the whole system is made up of 2 major steps. The first one is the deployment of the permissioned blockchain network that is running on the Hyperledger Fabric, while the other is to implement smart contracts with packing and installing chaincodes on the network by writing Go language. The data is stored on the blockchain network. To query or update the status of the information, smart contract has to be invoked. Figuratively, smart contracts have become a controller or an interface for users in the application layer to access all the data in the blockchain system. Also, it is immutable and tamper-proof for data to be stored inside a blockchain system. The whole system is implemented on a virtual machine installed with Ubuntu operating system.

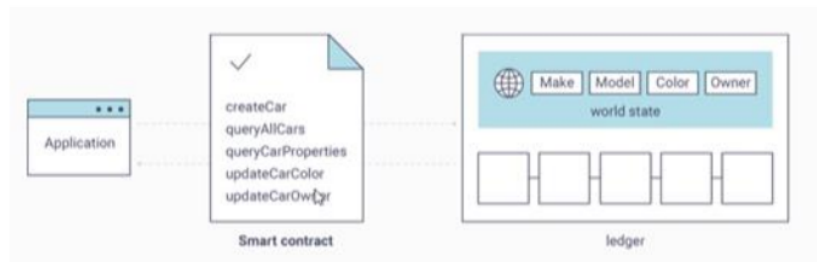


Figure 2 the principle for how smart contracts work in Hyperledger Fabric

First of all, to achieve immutability, tamper-proofing features, and restricted access to the network, certificate authority needs to be configured and set up to regulate and issue certificates which grant organizations access and interactions on the network. Furthermore, functions like adding or updating vehicle information, updating ownership and querying should be available for all the nodes. To sum up, there are basically four functions that can be implemented through the installation of chaincodes, also known as, smart contracts. The four main functions are as follows:

- Adding a new car information, including its brand, model, make, VIN, and owner ID
- Querying through key value
- Updating the ownership of a car
- Querying by range

Other than these four main functions, another auxiliary function to be devised is a blockchain system explorer, which can be used to check status quo of all the nodes and the status of all blocks and transactions visually and intuitively.

Now it is clear that the process of data storage and query is completed through the joint application of smart contracts and distributed ledger. Therefore, to successfully achieve the expected features, first thing to do is to set up the blockchain network correctly by applying the distributed ledger. Usually, Hyperledger Fabric is deployed on a Linux operating system. Thus, to simplify the process of preparing a Linux machine, Virtual Machine that can be run on a windows system should be taken into consideration to simulate a Linux operating environment. As shown in Figure 3, there are several steps of deploying a production network before running the chaincodes. Hyperledger Foundation has given out a simple tutorial guide to all the users who are willing to run their own projects by using Hyperledger Fabric. It is very clear to understand and developers can get started very quickly.

To sum up, the process of designing and building this blockchain system is shown in Figure 3

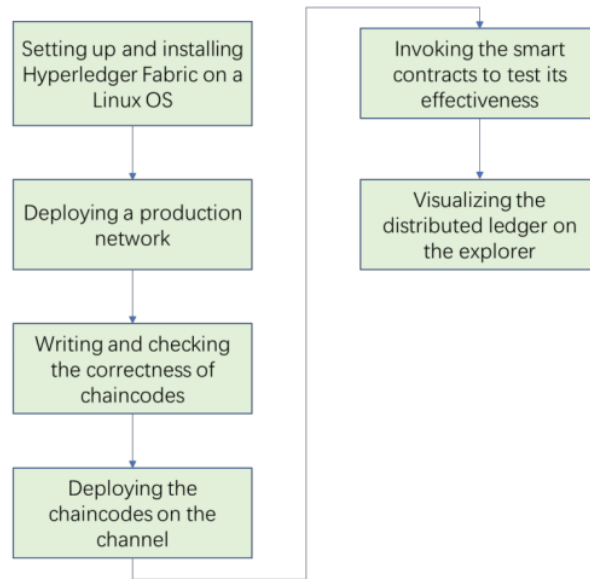


Figure 3 the process of building Blockchain Vehicle Management System

The first step is to prepare and set up the Linux Operating system, Ubuntu, in this case. Then the Hyperledger Fabric can be installed on this Linux OS. Then to deploy a production network, there are several tasks to be accomplished before a successful deployment. They will be specifically talked about in the later part of the dissertation.

The correctness of chaincodes, namely the codes of smart contracts, essentially matters. In 2016, the DAO attack[50] was conducted by hackers who used a minor mistake in codes of a smart contract deployed on Ethereum, causing around 4 million ETC loss in total on the Ethereum network at that time, which were worth 50 million dollars back then. Once the code of smart contract is deployed or installed on the blockchain, it is irreversible and immutable, the same as all the transactions. Only if a forking operation is done, a smart contract can be updated or changed. However, no matter it is hard forking or soft forking, it will end up with two different blockchains that can no longer work as a whole system. Therefore, it requires the maximum check after finishing the writing of smart contracts.

Last but not least, it is also important to validate the results after deploying chaincodes on the channel and invoking the smart contracts. To visualize the structure of blockchain ledger, a blockchain explorer will be coded and built as an auxiliary feature where the states of blocks and nodes can be easily observed as well as chaincodes and channels.

Due to the limitation of time and resources, the chief programmer and class owner both are the author alone

4. Blockchain System Design and Implementation

4.1 Hyperledger Fabric

Started in December 2015, Hyperledger project is an umbrella project of open

source blockchains developed by Linux Foundation. It has received contributions from IBM, Intel and SAP Ariba as a joint development of blockchain-based distributed ledger. Hyperledger Fabric, is a permissioned blockchain infrastructure, originally contributed by IBM and Digital Assets. It can be perceived as a modular architecture that delineates different roles between nodes in the infrastructure with the execution of smart contracts under consensus mechanism and configuring membership services. Usually, a fabric network consists of three major parts. First of all, peer nodes take on the job of chaincode execution, data access, transaction endorsement and applications interface. The other node that matters is orderer node. To make sure the consistency of the blockchain and deliver the endorsed transactions to the peers of the network, orderer nodes are designed to accomplish the tasks. And, the Membership Service Providers (MSPs) is used to manage certificates to identify authorities and roles, each generally implemented as a Certificate Authority. Every member needs to provide its certificate to authenticate its identity before invoking any function in the channel or during the approval stage of a chaincode. Therefore, it is very obvious that with such mechanism, traceability and immutability can be further strengthened.

4.1.1 Distributed ledger structure

Similar to Ethereum and Bitcoin, the blockchain structure in Hyperledger Fabric also includes block headers, transactions and hash values, etc. As shown in figure X, a block is made up of block header and block body or block data and block metadata. The block header mainly stores the essential information about a block, which are for example, block height, current block hash, previous block hash, etc. The block body consists of various transactions arranged in order. A transaction could be created when a node invoked a smart contract or sends out a transfer. And it would be added to a block after the consensus is reached. To be more specific, more detailed data is inside the transaction, such as transaction ID, previous transaction ID, which are both hash values, and channel ID, etc. Last but not the least, the block metadata contains the timestamp describing when the block was written, as well as the certificate, public key and signature of the block writer. And all the blocks are chained up by block hash values in each header to make up of the whole distributed ledger that should be maintained by every organization on this platform, thus explaining the figurative name, blockchain.

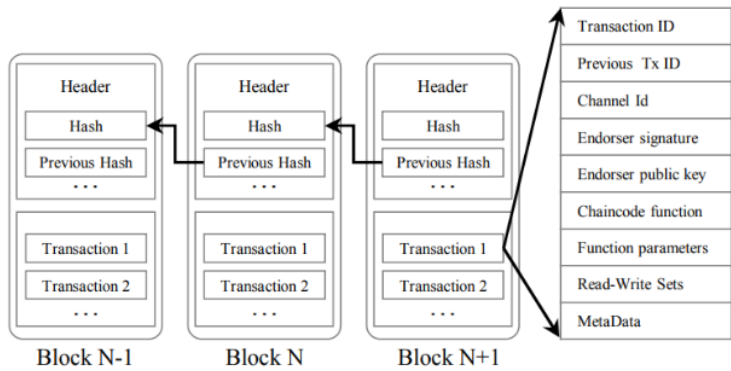


Figure 4 the structure of blocks in Fabric

4.1.2 Consensus Mechanism

Public chains like Ethereum, Bitcoin and Litecoin, as the name suggests, they can allow anyone to join the blockchain network, to see transactions on the ledger and to participate in its consensus progress. However, as a consortium chain, Hyperledger Fabric uses CA- Certificates Authority to issue identity certificates to the nodes that submitted application, which means that no one can be allowed to join and access the information on the ledger unless it is permissioned. The real reason behind this difference lie in the consensus mechanism that these two types of blockchain adopt.

The definition of consensus originated from distributed database. For distributed database, usually a consensus algorithm is used in a computer system to achieve agreement on a single affair among distributed systems. In public chains, Proof of Work (POW) mechanism is adopted. First of all, in order to understand POW, a brief explanation about the mining process that happens in Bitcoin is needed. When a new block is created, a collateral reward is also given to the owner of the new block, which is what we call bitcoin. However, it is not very easy and smooth to be the owner of the next block and win the prize. Therefore, to gain that coin reward, nodes need to compete for the ownership of the new created block by solving advanced mathematical problems. And the nodes that are competing for the reward is called miners. After a miner wins the competition, the verification process occurs, other computer nodes need to confirm whether the data on the hash block is accurate. All the nodes can update their blockchains with the new data if the information is correct. And the status of the distributed ledger gets updated. Typically, in order to validate the block information, 51% of all the nodes in the network needs to approve the block information. After the network has rewarded the miners for their efforts in validating the transactions, the miners then engage in a new competition to validate further transactions and supply Proof-of-Work. In essence, Proof-of-Work is the miner's way of assuring the verifier that the mining operation was legal and accurate. Or, demonstrating that the procedure is effective. The premise behind Proof-of-Work is that although verifying the veracity of the verification is quite simple, solving the miner's algorithm is extremely difficult.

Nevertheless, for a miner to validate a transaction, a lot of power and other resources are needed. The majority of these expenses are offset, though, when the miners are paid in free cryptocurrency in exchange for their labor. For a consortium chain, the POW consensus mechanism is not quite appropriate and also not eco-friendly. The consensus in Hyperledger Fabric is very different from public chains. Usually for a production network, Kafka consensus algorithm is used in Hyperledger Fabric. In Kafka, for the verification process, not every miner needs to participate into the verification. Only the leader does the ordering and only the in-sync replicas can be voted as a leader. This provides crash fault-tolerance and finality happens in a matter of seconds, while for POW, it takes about 10 minutes for a new block to be created, verified and added into the network, making the consensus more efficient and fast in a Hyperledger Fabric network.

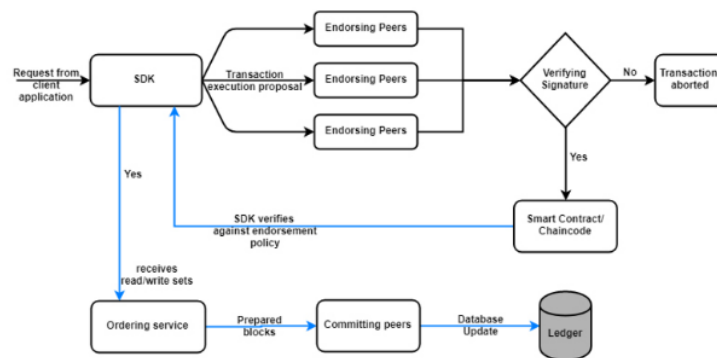


Figure 5 the process of consensus in Hyperledger Fabric

As shown in Figure 5, consensus in Hyperledger Fabric consists of three aspects: Endorsement, Ordering and Validation. First of all, the SDK receives transaction from a client application and generates a transaction proposal, sending it to endorsing peers. Then, the endorsing peers verify the form of the proposed transaction, if it got submitted before, the validity of digital signature and the authority of the client. After verifying the proposal is valid, the transaction proposal gets passed to the chaincode to invoke functions and further back to the SDK. As the application verifies the signature and compares the proposal responses to be the same, it also broadcasts the transaction containing read/write sets to ordering nodes. New blocks are prepared and transmitted to all the nodes in the network by ordering nodes. Last but not the least, all the nodes append the newly created block to the ledger, and for each valid transaction, the write sets are committed to the database. Consensus is essentially the full-circle confirmation of the accuracy of a group of transactions making up a block.

4.2 Production network deployment

After figuring out how the consensus works in Hyperledger Fabric, it is important to make sure all the components of the consensus and network are in place, this including CAs configuration and MSPs configuration, etc. In Hyperledger Fabric, network existed in channels. For the intention of carrying out private and secret transactions, a channel can be thought of as a private "subnet" of communication between two or more particular network participants. Every transaction on the network takes place on a channel, where each participant is required to be verified and approved in order to transact. When a peer joins a channel, a membership services provider (MSP) assigns that peer a unique identity and authenticates that peer with the other channel peers and services.

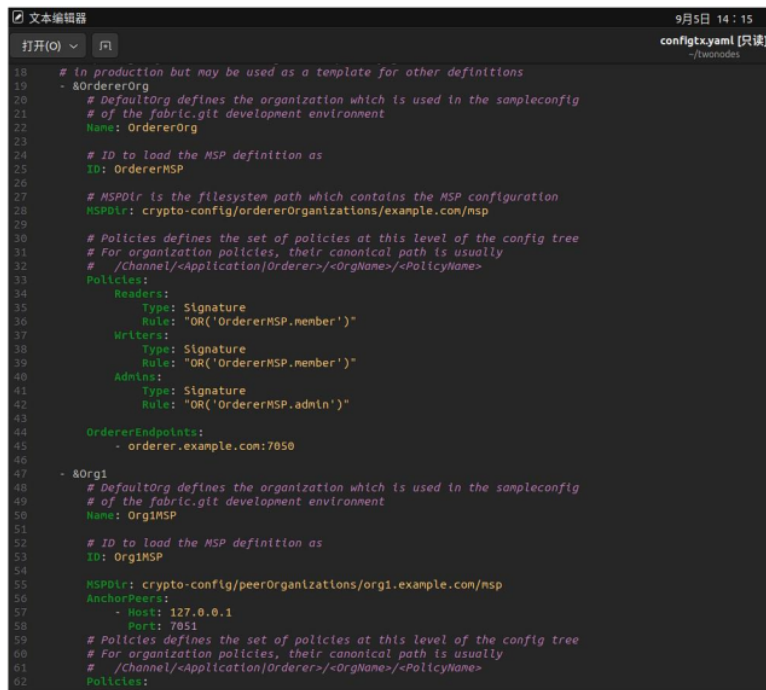
As shown in the following figure, there are basically five steps of deploying a production network. In the first step, there are several decisions to be made before deploying components. It is more like an overall consideration about the network you are going to set up. For example, as part of overall decisions, how the CAs for will be deployed needs to be decided. Production network should be using Transport Layer Security, which necessitates setting up a TLS CA and utilizing it to produce TLS certificates. Other than this, other questions like whether to use organizational units,

what type of database the ledger should adopt and if to create a system channel or not should all be taken into consideration.

After an initial decision on the network configuration, next step is to set up a cluster for resources. Frankly speaking, enough resources need to be guaranteed for components to run effectively. The use case determines the sizes of hardware. A single peer will require significantly more CPU and memory if we intend to link it to several high-volume channels as opposed to just one. We will be able to determine the necessary resources by establishing a proof-of-concept network and testing it under load.

Step three is setting up CAs. As discussed above, the first component that must be deployed in a Fabric network is a CA. Certificates must be created before the node itself is enrolled into the network to identify the node. There are mainly two types of CAs. Besides the aforementioned TLS CA, the other CA is used to generate the certificates of the admin of an organization, the MSP of that organization, and any nodes owned by that organization. It is named after its role on enrolling identities.

The certificates generated by TLS CA are usually attached to actions, preventing "middle man" attacks. It is worth noting that the TLS CA is solely used to issue certificates for nodes and can be shut down when the activity is over. Both one-way TLS (client alone) and two-way TLS (server and client), commonly known as "mutual TLS," are available to users. There is a field in the YMAL configuration file for enabling TLS, and a decision should be made before the deployment of the enrollment CA.



```
18 # In production but may be used as a template for other definitions
19 - &OrdererOrg
20   # DefaultOrg defines the organization which is used in the sampleconfig
21   # of the fabric.git development environment
22   Name: OrdererOrg
23
24   # ID to load the MSP definition as
25   ID: OrdererMSP
26
27   # MSPDir is the filesystem path which contains the MSP configuration
28   MSPDir: crypto-config/ordererOrganizations/example.com/msp
29
30   # Policies defines the set of policies at this level of the config tree
31   # For organization policies, their canonical path is usually
32   # /Channel/<Application/<Orderer/<OrgName/<PolicyName>
33   Policies:
34     Readers:
35       Type: Signature
36       Rule: "OR('OrdererMSP.member')"
37     Writers:
38       Type: Signature
39       Rule: "OR('OrdererMSP.member')"
40     Admins:
41       Type: Signature
42       Rule: "OR('OrdererMSP.admn')"
```

Figure 6 configuration of CA directories

After the CA settings have been configured and the CAs have been successfully created, they can be used to provide certificates for the identities and organizational components. This process is called MSP, membership service provider. The administrator of the CA assigns a username and password for the identity in the first stage. The identity can also be offered attributes and associations. Using the username and password, the identity can be enrolled after it has been registered. The CA will simultaneously create two certificates, referred to as the public key and private key. The private key is used to sign actions conducted by the identity, whereas the public key or public cert is broadcast to other network users. The CA will also provide a set of files named an "MSP" that contain the root of trust for the CA as well as the public certificate of the CA that issued the certificate.

The registration and enrollment of a node's identity must be managed by both a TLS CA and an enrollment CA, just like they are for an organization admin identity. When registering a node with the enrollment CA, it assigns the role of peer or orderer to an organization node rather than admin or user. As the identities' given rights are only important at the local (node) level, the MSP structure for a node is referred to as a "local MSP". Customizing the configuration file of every node is required before deployment. This configuration file is known as `orderer.yaml` for ordering nodes, whereas `core.yaml` is the name of the configuration file for the peer.

Once we have acquired all the certificates and MSPs we need, it is ready to create nodes. There are primarily three choices for peer node configuration: 1) Edit the YAML document that is included with the binaries. 2) When deploying, use overrides for environment variables. 3) Include flags with CLI commands. And a peer's fundamental configuration parameters are also mentioned. Environment variables must be added to or overridden in the yaml file. However, in this project, to achieve efficiency and convenience, docker containers are utilized for deployment of peer and ordering nodes. What's more, a default `core.yaml` can be found inside the peer nodes image directory of docker, meaning that by running the docker container, the `core.yaml` could be up and running as well. Therefore, it is feasible and much easier to bring up all the nodes by configuring and running the docker containers.

```

# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

version: '2.4'

volumes:
  orderer.example.com:
  peer0.org1.example.com:
  peer0.org2.example.com:

networks:
  test:
    name: twonodes_test

services:
  orderer.example.com:
    container_name: orderer.example.com
    image: hyperledger/fabric-orderer:latest
    labels:
      service: hyperledger-fabric
    environment:
      - FABRIC_LOGGING_SPEC=INFO
      - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
      - ORDERER_GENERAL_LISTENPORT=7050
      - ORDERER_GENERAL_LOCALMSPID=ordererMSP
      - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
      # enabled TLS
      - ORDERER_GENERAL_TLS_ENABLED=true
      - ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
      - ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
      - ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
      - ORDERER_KAFKA_TOPIC_REPLICATIONFACTOR=1
      - ORDERER_KAFKA_VERBOSE=true
      - ORDERER_GENERAL_CLUSTER_CLIENTCERTIFICATE=/var/hyperledger/orderer/tls/server.crt
      - ORDERER_GENERAL_CLUSTER_CLIENTPRIVATEKEY=/var/hyperledger/orderer/tls/server.key
      - ORDERER_GENERAL_CLUSTER_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]

```

Figure 7 the configuration of docker-compose.yaml

CONTAINER ID	IMAGE	CMD	CREATED	STATUS	PORTS
4061	hyperledger/fabric-tools:latest	"/bin/bash"	7 days ago	Up 4 seconds	
5d11f3367a53	hyperledger/fabric-orderer:latest	"orderer"	7 days ago	Up 4 seconds	0.0.0.0:7050->7050/tcp, :::7050->7050/tcp, 0.0.0.0:7053->7053/tcp, :::7053->7053/tcp
80a4fb0cfeef	hyperledger/fabric-peer:latest	"peer node start"	7 days ago	Up 4 seconds	0.0.0.0:7051->7051/tcp, :::7051->7051/tcp
5d11f3367a53	hyperledger/fabric-tools:latest	"/bin/bash"	7 days ago	Up 4 seconds	
80a1f7b03931	hyperledger/fabric-peer:latest	"peer node start"	7 days ago	Up 5 seconds	7051/tcp, 0.0.0.0:9051->9051/tcp, :::9051->9051/tcp

Figure 8 Successful deployment of peer nodes

After the creation of nodes, it is important to connect them with other nodes. What is left of this work is to join all the peer organizations and peer nodes into the channel. Also, to develop chaincode, it is essential to check the logic of whole chaincode before packing and having them installed on the right channel since it cannot be easily altered after being released, unless updating the whole channel. This topic is going to be covered in the later part.

In this dissertation, due to the limitation of time and resource, the deployed network will only include two organizations, each owning one peer, and a single ordering service organization that owns a single ordering node to perform some basic features and effects of blockchain system.

4.3 Chaincode installation and deployment

Chaincode is a piece of software that implements a specified interface and can be implemented in Go, Node.js, or Java. Through transactions sent by apps, Chaincode manages and initializes the ledger state while running in a distinct process from the peer. In Hyperledger, a chaincode can be referred to as a "smart contract" since it often

manages business logic that has been approved by network users. In a proposal transaction, a chaincode can be used to update or query the ledger.

In order to implement smart contracts, application developers can use the contract interface, a high level API, provided by the fabric-contract-api. An easy way to start building business logic is by using this API. In order to manage cars and vehicles from different owners, an elementary vehicle management smart contract is proposed and written to log and register car information. As mentioned before, there are mainly four functions including create cars, query specific cars, query all the car information and change ownerships.

In this project, because of its simplicity and similarity with python, go language is adopted for the writing of the smart contract, namely the chaincode.

```
1 /*
2 SPDX-License-Identifier: Apache-2.0
3 */
4
5 package main
6
7 import (
8     "encoding/json"
9     "fmt"
10    "strconv"
11
12    "github.com/hyperledger/fabric-contract-api-go/contractapi"
13)
14
15 // SmartContract provides functions for managing a car
16 type SmartContract struct {
17     contractapi.Contract
18 }
19
20 // Car describes basic details of what makes up a car
21 type Car struct {
22     Make    string `json:"make"`
23     Model   string `json:"model"`
24     Colour  string `json:"colour"`
25     Owner   string `json:"owner"`
26     VIN     string `json:"vin"`
27 }
28
29 // QueryResult structure used for handling result of query
30 type QueryResult struct {
31     Key      string `json:"key"`
32     Record   *Car
33 }
34
35 // InitLedger adds a base set of cars to the ledger
36 func (s *SmartContract) InitLedger(ctx contractapi.TransactionContextInterface) error {
37     cars := []Car{
38         Car{Make: "Toyota", Model: "Prius", Colour: "blue", Owner: "Tomoko", VIN: "USUM50791529TBM"},
39         Car{Make: "Porsche", Model: "Panamera", Colour: "purple", Owner: "Tomoko", VIN: "MFLUF01973843XPNB"},
40         Car{Make: "Ford", Model: "Mustang", Colour: "red", Owner: "Brad", VIN: "NFMHC426X1752DZJL"},
41         Car{Make: "Hyundai", Model: "Tucson", Colour: "green", Owner: "Jin Soo", VIN: "TYAFU94815018PPKG"},
42         Car{Make: "Volkswagen", Model: "Passat", Colour: "yellow", Owner: "Max", VIN: "BFREG17424597MMCT"},
43         Car{Make: "Tesla", Model: "S", Colour: "black", Owner: "Adriana", VIN: "ZRLHE45154012VXCL"},
44         Car{Make: "Peugeot", Model: "205", Colour: "purple", Owner: "Michel", VIN: "JXTC238901567YXHZ"},
45         Car{Make: "Chery", Model: "S22L", Colour: "white", Owner: "Aarav", VIN: "BYVRT21448408CFZD"},
46     }
```

Figure 9 The Go code of the smart contract

1) Initilization

```
func (s *SmartContract) InitLedger(ctx contractapi.TransactionContextInterface) error {
    cars := []Car{
        Car{Make: "Toyota", Model: "Prius", Colour: "blue", Owner: "Tomoko", VIN: "USUM50791529TBM"},
        Car{Make: "Porsche", Model: "Panamera", Colour: "purple", Owner: "Tomoko", VIN: "MFLUF01973843XPNB"},
        Car{Make: "Ford", Model: "Mustang", Colour: "red", Owner: "Brad", VIN: "NFMHC426X1752DZJL"},
        Car{Make: "Hyundai", Model: "Tucson", Colour: "green", Owner: "Jin Soo", VIN: "TYAFU94815018PPKG"},
        Car{Make: "Volkswagen", Model: "Passat", Colour: "yellow", Owner: "Max", VIN: "BFREG17424597MMCT"},
        Car{Make: "Tesla", Model: "S", Colour: "black", Owner: "Adriana", VIN: "ZRLHE45154012VXCL"},
        Car{Make: "Peugeot", Model: "205", Colour: "purple", Owner: "Michel", VIN: "JXTC238901567YXHZ"},
        Car{Make: "Chery", Model: "S22L", Colour: "white", Owner: "Aarav", VIN: "BYVRT21448408CFZD"},
        Car{Make: "Flat", Model: "Punto", Colour: "violet", Owner: "Parl", VIN: "GSKGY44145448GBY"},
    }

    for i, car := range cars {
        carAsBytes, _ := json.Marshal(car)
        err := ctx.GetStub().PutState("CAR"+strconv.Itoa(i), carAsBytes)

        if err != nil {
            return fmt.Errorf("Failed to put to world state. %s", err.Error())
        }
    }

    return nil
}
```

Figure 10 The Go code of InitLedger

After importing the packages we need for necessary dependencies and defining the car struct, it is actually optional to populate the ledger with some initial data. Here, to demonstrate the effect of queryallcars function, we put in some test data. To query the data, the key-value set is applied in Hyperledger. Therefore, all of these test data are attached with keys. And this process is achieved by using a loop of number added behind string "CAR".

2) Create car

```
func (s *SmartContract) CreateCar(ctx contractapi.TransactionContextInterface, carNumber string, make string, model string, colour string, owner string, vin string) error {
    car := Car{
        Make: make,
        Model: model,
        Colour: colour,
        Owner: owner,
        Vin: vin,
    }

    carAsBytes, _ := json.Marshal(car)

    return ctx.GetStub().PutState(carNumber, carAsBytes)
}
```

Figure 11 The Go code of CreateCar

To create cars, all the information has to be typed in as the order demonstrated in the function. By invoking this function, a new piece of information about a car that has never been registered into the system will be created. It can serve as the equivalent of car registration in the traditional car management system.

3) Query a specific car

```
func (s *SmartContract) QueryAllCars(ctx contractapi.TransactionContextInterface) ([]QueryResult, error) {
    startKey := ""
    endKey := ""

    resultsIterator, err := ctx.GetStub().GetStateByRange(startKey, endKey)

    if err != nil {
        return nil, err
    }
    defer resultsIterator.Close()

    results := []QueryResult{}

    for resultsIterator.HasNext() {
        queryResponse, err := resultsIterator.Next()

        if err != nil {
            return nil, err
        }

        car := new(Car)
        _ = json.Unmarshal(queryResponse.Value, car)

        queryResult := QueryResult{Key: queryResponse.Key, Record: car}
        results = append(results, queryResult)
    }

    return results, nil
}
```

Figure 12 The Go code of QueryCar

To look for information about a specific person or features, this function can be of great use. The key-value pair is very compatible with such non-relational database. Due to the simplicity of key-value pairs, it's easy to write and read them. Also, Key-value pairs foster speedy and scalable key-value stores, which allows them to be flexible.

4) Change ownership

```

func (s *SmartContract) ChangeCarOwner(ctx contractapi.TransactionContextInterface, carNumber string, newOwner string) error {
    car, err := s.QueryCar(ctx, carNumber)

    if err != nil {
        return err
    }

    car.Owner = newOwner

    carAsBytes, _ := json.Marshal(car)

    return ctx.GetStub().PutState(carNumber, carAsBytes)
}

```

Figure 13 The Go code of ChangeCarOwner

This function enables the information update of car ownership from one owner to another. Not only can it be used in such scenario, but also feasible under various business contexts such as used car trade.

4.3.1 Lifecycle of Chiancode

The chaincode lifecycle is the process of agreeing and implementing chaincode installations, upgrades, etc. when operating chaincode on a channel. It can be best summarized as four tasks:

- ### 1) Package the chaincode

This can be completed by any organization in the channel.

```
bash-5.1# peer lifecycle chaincode package VM.tar.gz\
> --path github.com/hyperledger/fabric-cluster/chaincode/go/\
> --label VM_1
```

Figure 14 Packaging the chaincode

The first command is to package the chaincode into a compressed package. And the second line is the directory of the chaincode. The last line is to set up the label for this chaincode. Once this command is implemented, the package VM.tar.gz will be created in the working directory of cli1. Therefore, we need to copy it into the other cli, the cli2.

- ## 2) Install the chaincode on peers

```
peer lifecycle chaincode install VM.tar.gz
```

Figure 15 Installation chaincode on peer

Every organization that will use the chaincode to endorse a transaction or query the ledger needs to complete this step. By using peer lifecycle chaincode install syntax, the packaged chaincode could be installed on the peer node. After the successful installation, it is important to note the package id as shown below. It is a hash value to identify the installed chaincode, which will be used in the approval step.

[illegible]

Figure 16 Success respond after installation

- 3) Approve a chaincode definition for the organization

Every organization that will use the chaincode needs to complete this step. The chaincode definition needs to be approved by a sufficient number of organizations to satisfy the channel's Lifecycle Endorsement policy before the chaincode can be started on the channel. For this step, the path of CA file of orderer node has to be provided in

the command, together with those of Org1 and Org2.

```
sudo docker exec -it cli01 /bin/bash
root@cli01:/home/charles/fabric# ./scripts/test-network.sh peer lifecycle chaincode approveformyorg -a localhost:7050 -ordererURLmyorderer1de.orderer.example.com --tls --cafile /home/charles/fabric/scripts/test-network/organizations/peerOrganizations/org1.example.com/tlsca/tlsca.example.com-cert.pem --channelID mychannel --name VM --version 1.0 --packageID fabric --sequence 1
```

Figure 17 Approving the chaincode on the channel

4) Commit the chaincode definition to the channel

Lastly, after the requisite number of organizations on the channel have authorized the chaincode, any organization inside the network may submit the commit transaction. Before submitting the transaction to commit the chaincode definition, the submitter must first compile sufficient endorsements from peers of the organizations that have approved. Once the chaincode is committed, the content can no longer be changed. Only by chaincode upgrading can its contents be changed, which also needs to be permitted by all the organizations.

```
sudo docker exec -it cli01 /bin/bash
root@cli01:/home/charles/fabric# ./scripts/test-network.sh peer lifecycle chaincode commit -a localhost:7050 -ordererURLmyorderer1de.orderer.example.com --tls --cafile /home/charles/fabric/scripts/test-network/organizations/peerOrganizations/org1.example.com/tlsca/tlsca.example.com-cert.pem --channelID mychannel --name VM --peerAddresses localhost:7051 --tls --cafile /home/charles/fabric/scripts/test-network/organizations/peerOrganizations/org1.example.com/tlsca/tlsca.example.com-cert.pem --peerAddresses localhost:7052 --tls --cafile /home/charles/fabric/scripts/test-network/organizations/peerOrganizations/org2.example.com/tlsca/tlsca.example.com-cert.pem --sequence 1
```

Figure 18 Commitment of the chaincode

It cannot be called a whole lifecycle of chaincode, until the successful implementation of these four steps. Up to now, we have successfully configured the network and set up all the nodes and CA files that is essentially required for chaincode implementation. With the completion of an intact lifecycle of a chaincode, it is ready to invoke chaincode and create transactions so that blocks will be created as well and get the whole ledger up and running.

4.3.2 Problems and Solutions

Unfortunately, the chaincode VM, abbreviated for vehicle management, cannot be installed on the channel due to some connection failure with the docker endpoint, which is caused by the virtual machine connection failure with the github repository. However, the process is successful on the test network which the built network resembles. Therefore, for the chaincode display, it was done on the test network that is provided by the Fabric itself. And for the chapter above, it is to demonstrate the work of deploying the network and clarify the work that the author has done.

4.4 Blockchain Explorer

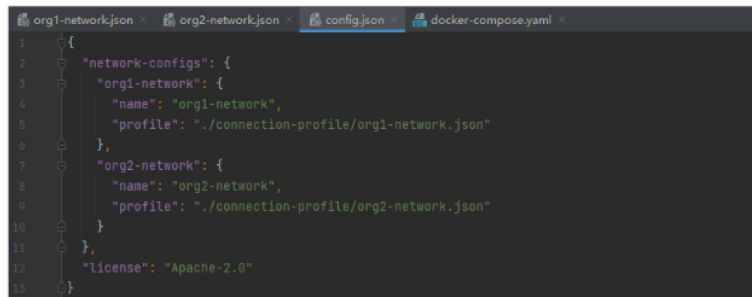
To better visualized the process of how the blockchain system works, a blockchain explorer is built and it is running on the Hyperledger. Block explorer is a website or a tool that allows one to browse through blocks, view wallet addresses, network hash rate, transaction data and other key information on the blockchain. There are block explorers for Bitcoin and also for individual altcoins. Some block explorers can be used on multiple different networks, while others are only for one specific blockchain. A BTC block explorer, for example, would only be able to find information from the Bitcoin network, such as when someone is sending Bitcoin to another wallet.

Beyond these types of tasks, block explorers can also be used to:

- Examine the history of any wallet address,
- Explore change addresses
- Explore the largest transaction that was sent in the last 24 hours.
- Discover the individual or mining pool who mined a specific block.

- Explore the genesis block
- See other information specific to a blockchain

To be specific, the way to set up and build a Hyperledger block explorer is quite similar with what we did to configure and set up the peer nodes. All we need to do is pull a few files from Hyperledger open source repository on github and configure the yaml file as we want. First of all, as shown in Figure 19 and Figure 20 the content of configuration file docker-compose.yaml and config.json can be pulled from the github repository. To fit for your customized blockchain network configuration, several adaptations needs to be done. They are quite similar with what has been done with the peer node configuration yaml file, such as matching the directories of the CA files or matching the name of the network with that of the explorer configuration file. Other than this, in the test-network.json file, Org2 configuration needs to be added since there are two organizations in the blockchain network. What's more, extra care has to be paid to the docker-compose.yaml configuration, to set the volumes of the directories consistent with the correct directories. After all is checked and ready to go, in the directory of the explorer, using the same command that created the network and peer nodes, “docker-compose up -d”, can bring up the block explorer. The explorer can only be deployed under the circumstances that the blockchain network is running as well. And the final display of the explorer will be revealed in the next section.



```
1 {
2   "network-configs": {
3     "org1-network": {
4       "name": "org1-network",
5       "profile": "./connection-profile/org1-network.json"
6     },
7     "org2-network": {
8       "name": "org2-network",
9       "profile": "./connection-profile/org2-network.json"
10    }
11  },
12  "license": "Apache-2.0"
13 }
```

Figure 19 Configuration of organizations in the block explorer

```

1
2 # SPDX-License-Identifier: Apache-2.0
3 version: '2.1'
4
5 volumes:
6   pgdata:
7   walletstore:
8
9 networks:
10  mynetwork.com:
11    name: fabric_test
12
13 services:
14
15  explorerdb.mynetwork.com:
16    image: hyperledger/explorer-db:latest
17    container_name: explorerdb.mynetwork.com
18    hostname: explorerdb.mynetwork.com
19    environment:
20      - DATABASE_DATABASE=fabricexplorer
21      - DATABASE_USERNAME=hppoc
22      - DATABASE_PASSWORD=password
23    healthcheck:
24      test: "pg_isready -h localhost -p 5432 -q -U postgres"
25      interval: 30s
26      timeout: 10s
27      retries: 5
28    volumes:
29      - pgdata:/var/lib/postgresql/data
30    networks:
31      - mynetwork.com
32
33  explorer.mynetwork.com:
34    image: hyperledger/explorer:latest
35    container_name: explorer.mynetwork.com
36    hostname: explorer.mynetwork.com
37    environment:
38      - DATABASE_HOST=explorerdb.mynetwork.com
39      - DATABASE_DATABASE=fabricexplorer
40      - DATABASE_USERNAME=hppoc

```

Document 1/1 | services: explorer.mynetwork.com: volumes: Item 4/4 | walletstore/opt/exp...

Services

Figure 20 Configuration of the block explorer

```

1
2 "name": "org1-network",
3 "version": "1.0.0",
4 "client": {
5   "tlsenable": true,
6   "adminCredential": {
7     "id": "exploreradmin",
8     "password": "exploreradmin"
9   },
10   "enableAuthentication": true,
11   "organization": "OrgMSP",
12   "connection": {
13     "timeout": {
14       "peer": {
15         "endorser": "300"
16       },
17       "orderer": "300"
18     }
19   },
20 },
21 "channels": {
22   "mychannel": {
23     "peers": {
24       "peer0.org1.example.com": {}
25     }
26   },
27 },
28 "organizations": {
29   "OrgMSP": {
30     "mspId": "OrgMSP",
31     "adminPrivateKey": {
32       "path": "/tmp/cryptos/peerOrganizations/org1.example.com/users/Admin@org1.example.com/keystore/privkey"
33     },
34     "peers": [
35       {
36         "peerId": "peer0.org1.example.com",
37         "signedCert": {
38           "path": "/tmp/cryptos/peerOrganizations/org1.example.com/users/Admin@org1.example.com/keystore/Admin@org1.example.com-cert.pem"
39         }
40       }
41     ]
42   }
43 }

```

```
org1-network.json | org2-network.json | docker-compose.yaml
{
  "name": "org1-network",
  "version": "1.0.0",
  "client": {
    "clientId": true,
    "adminCredential": {
      "id": "exploreradmin",
      "password": "exploreradminps"
    },
    "enableAuthentication": true,
    "organization": "orgMSP",
    "connection": {
      "timeout": {
        "peer": {
          "endorser": "300"
        },
        "orderer": "300"
      }
    }
  },
  "channels": {
    "mychannel": {
      "peers": {
        "peer0.org2.example.com": {}
      }
    }
  },
  "organizations": {
    "orgMSP": {
      "mspId": "orgMSP",
      "adminPrivateKey": {
        "path": "/tmp/cryptopeerOrganizations/org1.example.com/users/Admin@org2.example.com/asp/keysprv/gg1.sh"
      },
      "peers": ["peer0.org2.example.com"],
      "signedCert": {
        "path": "/tmp/cryptopeerOrganizations/org2.example.com/users/Admin@org2.example.com/asp/11certs/Admin@org2.example.com-cert.pem"
      }
    }
  }
}
```

5. Results

5.1 Blockchain network and chaincode implementation

After the successful implementation and initialization of the chaincode, it is up and ready on the channel for being invoked. What comes next with it is the running of the whole system, creating new transactions and blocks. By flowing through such process, the distributed ledger keeps updating and the blockchain expands. Here are some demonstrations of successful deployment of the network and chaincode.

By using the command “docker ps -a”, it is clear to determine if the docker container that all the nodes are running on is up. Once they are all up, through each peer node of the organization joined in the channel, functions inside the chaincode can be invoked therefore creating transactions for the network. For example, to register a new piece of information about a car, the function CreateCar has to be invoked. Besides, it is easy to check up on the status of the data inside the smart contract using QueryAllCars.

```

@nghttpcliays/homes/charles/fabric-samples/test-network peer chatnode query -c mychannel -n fabric -C '[{"Args":["querycars","CAG"]}'
{"name":"Peugeot","model":"205","colour":"purple","owner":"Michel"}

@nghttpcliays/homes/charles/fabric-samples/test-network peer chatnode query -c mychannel -n fabric -C '[{"Args":["queryallcars"]}'
{"key":"CAG","Record":{"name":"Toyota","model":"Prius","colour":"blue","owner":"Tomoko"}}, {"key":"CAG","Record":{"name":"Honda","model":"Mustang","colour":"red","owner":"Brad"}}, {"key":"CAG","Record":{"name":"Honda","model":"Accord","colour":"green","owner":"Jin Soo"}}, {"key":"CAG","Record":{"name":"Volksagen","model":"Passat","colour":"yellow","owner":"Nao"}}, {"key":"CAG","Record":{"name":"Test","name":"Suzuki","colour":"black","owner":"Adrianus"}}, {"key":"CAG","Record":{"name":"Peugeot","model":"205","colour":"purple","owner":"Michel"}}, {"key":"CAG","Record":{"name":"Chevy","model":"S200","colour":"white","owner":"Nara"}}, {"key":"CAG","Record":{"name":"Fiat","model":"Fiat","colour":"violet","owner":"Fai"}}, {"key":"CAG","Record":{"name":"Fiat","model":"Haro","colour":"indigo","owner":"Val"}}, {"key":"CAG","Record":{"name":"Nissan","model":"Fairlady","colour":"brown","owner":"Shiroto"}}

```

To register a new car, all to do is to invoke the CreateCar function. Besides submitting the information about the car, the directories of CA certificates of the orderer node and TLS root certificates of the peer nodes from both organizations has to be provided to make sure the state variation of the ledger on the nodes is consistent with each other. The invoke details and successful respond are shown below.

```
test/nginx/chaos/jhmc/fabrics/samples/test/networks/peer_chocolate_k8s -w order.example.com:7551 --tls true --cafile /home/chaos/jhmc/fabrics/samples/test/networks/organizations/order/order.example.com/certs/ca.crt --cert /home/chaos/jhmc/fabrics/samples/test/networks/organizations/order/order.example.com/certs/peer_chocolate_k8s.crt --key /home/chaos/jhmc/fabrics/samples/test/networks/organizations/order/order.example.com/certs/peer_chocolate_k8s.key --peer-addresses peer.org.example.com:7551 --tlscertfiles /home/chaos/fabrics/samples/test/networks/organizations/peer/organizations/org.example.com/certs/peer.org.example.com/certs/ca.crt --peer-addresses peer.org.example.com:9551 --tlscertfiles /home/chaos/fabrics/samples/test/networks/organizations/peer/organizations/org2.example.com/certs/peer.org2.example.com/certs/ca.crt -c "[Argv]::CreateCtrl", "CARB", "Porsche", "Panamera", "Silver Grey", "Red"]
```

Figure 24 CreateCar

```
2022-09-04 05:05:57.024 EST 0001 INFO [chaincodeCmd] chaincodeInvoke@Query -> Chaincode invoke successful. result: status:200
```

Figure 25 Successful invoke of the function

To transfer the ownership of a car, ChangeCarOwner can be utilized to accomplish such mission. Similar to the process of registering a new car, information about each node's TLS root certificates directory has to be typed in. Usually for actions which involve changing the state of the ledger such as create a new car or change its ownership, certificates are required in Fabric. On one hand, it is to verify the identity of the transaction proposer, on the other hand, it also aims to keep the status of all the nodes consistent.

```
root@blackchansys:/home/charles/fabric-samples/test-network# peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /home/charles/fabric-samples/test-network/organizations/peerOrganizations/orderer.example.com/organizations/orderer.example.com/tlsca.example.com-cert.pem -C mychannel -n fabcar --peerAddresses peer8.org1.example.com:7051 --tlsoutCertFiles /home/charles/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peer8.org1.example.com/tlsca.crt --peerAddresses peer8.org1.example.com:7051 --tlsoutCertFiles /home/charles/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peer8.org1.example.com/tlsca.crt -c '{"args":["ChangeCarOwner","CAR10","Charles"]}'
[chaincodeCmd] chaincodeInvoke@Query -> Chaincode invoke successful. result: status:200
```

Figure 26 Successful transferring the ownership

As shown in the figure 26, by invoking the function, it is already successful transferring the ownership of CAR10 which was just created to another guy named Charles.

5.2 Block Explorer implementation

By combining the visualizations from the block explorer, it is more intuitional and to understand the whole structure and the process of what is happening inside the network. After the successful deployment of the explorer, the website can be reached by typing the local host with the port. As seen in Figure 27, to login, it requires the choice of organization and userID with its password. In a real business scenario, different organizations have different passwords. However, here to simplify the demonstration, same userID and passwords are set.

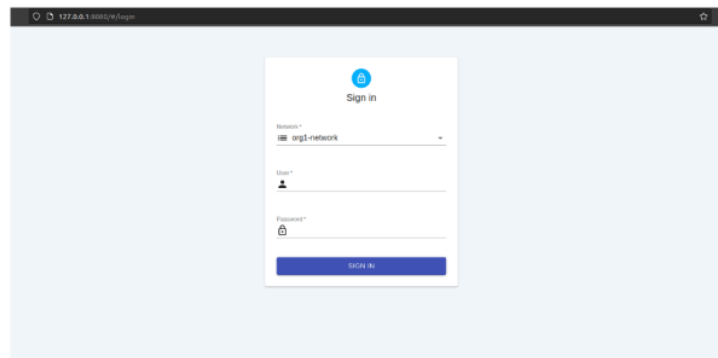


Figure 27 The login page of the explorer

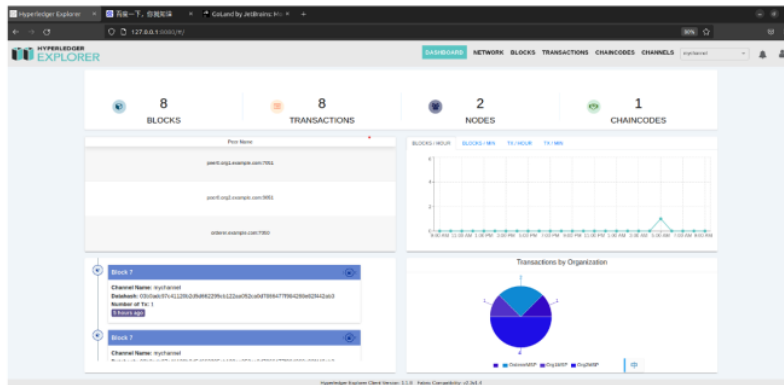


Figure 28 the dashboard of the explorer

Displayed in Figure 28, here is the dashboard of the block explorer, it is very clear to observe the chain from here. It illustrates the details of the network, such as the number of blocks, transactions and chaincodes. The activity of the network can also be observed through the dashboard by using blocks/hour or transaction/hour, etc.

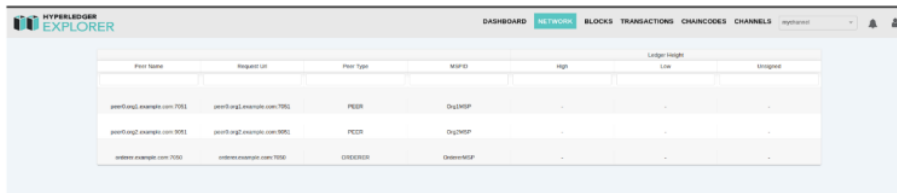


Figure 29 network organizations

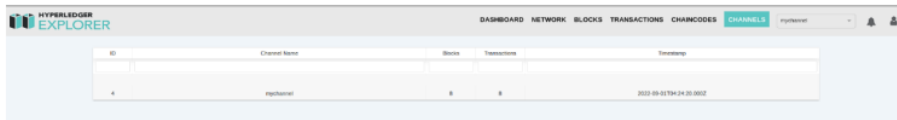


Figure 30 channel

Other than this, the explorer can also be used to monitor what is going on in the channel, and which organization is in which channel, given the situation that there might not be just one channel. For example, the details of the network is revealed in Figure 29 and Figure 30.



Figure 31 details of blocks

Of course, it records the details of blocks over time. As shown in Figure 31, in the

blockchain, only essential information about the status of the blocks would be recorded and displayed on the chain, such as block hash or prehash. The ledger data is stored in the LevelDB so that it can speed up the process of submitting and committing transactions.



Chaincode Name	Channel Name	Path	Transaction Count	Version
value	mychannel	-	2	1.0

Figure 32 chaincode

As shown in Figure 32, the explorer also monitors the status of chaincodes. It records the name of the chaincode, the channel it was deployed on, and the transaction count. For business networks that requires huge number of members, there would be demands for smart contracts to practice their business logic. Thus, the explorer can be really useful and beneficial to the management and monitoring of smart contracts.

6. Discussion

This project demonstrates how Hyperledger Fabric can be used for automotive information management. After the blockchain vehicle management system has been successfully implemented, there are some advantages of Hyperledger Fabric that can help industries other than those with applications at the level of cryptocurrencies, such as the management of vehicle information, healthcare records, and the insurance sector. Data protection laws apply to many use cases, particularly those in the financial and healthcare industries, which necessitate knowing who is a part of the network and who is accessing what data. The permissioned network of Hyperledger Fabric is far more practical and appropriate in these situations than public chains.

A particularly frequent requirement is the necessity to integrate a unique identity management system. Although many businesses want to create a permissioned blockchain, they want to continue using their current identity management system for network users. This is made possible via modular architecture.

Hyperledger uses LevelDB and CouchDB for data storage and access, where a key-value pair is typically used as the data structure. When users use this type of data structure, functions to obtain data from the ledger can be retrieved more quickly.

The modular design offered by Fabric allows for the creation of plug-in components by developers. The benefit to developers is enormous. Thanks to this capability, they can quickly add elements like customized identity management, etc. Additionally, this modular architecture divides the transaction processing into three phases: distributed logic processing and agreement, transaction ordering, and transaction validation and commitment. This division offers the following benefits: Network scalability and performance are improved by minimizing the amount of trust levels and verification, which keeps the network and processing clean.

Support of HSM (Hardware Security Modules) is essential for managing and protecting digital keys for robust authentication. For key generation, Hyperledger Fabric offers both modified and unmodified PKCS11, supporting scenarios like identity

management that demand increased security. HSM improves the security of keys and sensitive data in identity management-related circumstances.

However, there are some opposing voices and critiques about Hyperledger Fabric as well. One of Fabric's detractors claims that it is permissioned and is not a public blockchain because of this. Complete decentralization entails total openness. To firms and businesses whose commercial privacy demands protection, it may not necessarily mean a positive thing. Another criticism claims that actual immutability cannot exist without POW and complete decentralization. Since Fabric's consensus is not POW, it is generally acknowledged that complete immutability is unachievable. As a result, the difficulty of altering the structure of a ledger in Fabric is much simpler than conducting the event of a sybil attack in a public chain. However, proponents believe that a tamper-evident system, not a tamper-proof one, is required in a network with known players who are all bound by the governance and code of behavior of the corporate organization.

7. Conclusion

Blockchain technology has attracted more and more attentions, stimulating increasing researches across the world since its birth. It is no doubt that it will become an essential technology in the future. In application like this, it can be leveraged to further decrease the information asymmetry between the car buyers and sellers. Of course, the history owners of a car are also available and clear by querying in the chaincode, making the regulation much simpler for governance.

In this project, a vehicle management prototype based on blockchain platform is proposed and built. It has successfully demonstrated that blockchain technology is applicable in the area of automobile information security. It ensures safety from cyber-attacks because of the nature of its encryption method. And by applying consortium chain, Hyperledger Fabric speeds up the process of consensus and saves much more energy than the POW consensus, therefore increasing the throughput of transactions tremendously and also making the development and application of such technology eco-friendly. The system is designed to perpetually stores and records information about cars and transactions involving all the participating entities in the blockchain network using chaincodes. The process of car registration, ownership transferring and tracking was all achieved through the invoking of chaincodes. The whole system has demonstrated the possibility and availability of distributed ledger application in the field of car information management.

To sum up, the proposed fabric architecture provides a secure, transparent and immutable system to store, share and exchange car information in a peer-to-peer network of different stakeholders and organizations.

8 Reflection and Future Work

8.1 Reflection

Reaching at the end of this project, I realize I have learnt a lot when I look back. There are things that can be improved and could have been done better. Had I managed the schedule more properly, I could have more energy and time to spend on the design

of the network so that it can be more complex and its performance can be better. Also, tools to test the performance of a blockchain network, caliper, designed by HUAWEI could have been adopted and several comparisons could be made. The other lesson is always to remember to reach out to the community. I could not have achieved what I did if it not been for the help that I can get from all these blockchain communities. And I should pass on this passion and help to others in need. Last but not least, I know there are many areas that this project can be improved, such as the complexity of the network can be more intense to improve its performance. And to make the use cases more real, this blockchain network could be deployed across different devices so that CAs and organizations can have more representatives and the scenario is more like the real situation. Also, had I managed the time better, the block explorer could be further developed and its interface design can be more refined, with richer features and details to display.

8.2 Future work

It is important to look at the future and figure out where to go if this work is continued. It is quite obvious from the reflection that a big future work is to broaden the number of organizations in the network, to simulate a real business scenario level. Then, using some performance test tools to analyze the performance ability of the network should be also considered, since the transaction throughput of the network can affect the network fluency and member experience when member organizations are using the network.

Another area of future work lies in the smart contract. Other than recording car information, there can be some other functions and features that can be achieved by designing different chaincodes. For example, in a used car trading scenario, for different roles in the market, there can be different channels equipped with different smart contracts. A customer can use the smart contract to check up on the car and even appoint a time for offline meetup with the seller. The use case of the network should not only be limited to car information management. Selling contracts or insurance contracts can also be stored on the blockchain network.

Finally, it is really important to keep open minded. It would be interesting to combine IPFS technology with blockchain. To solve the problems of insufficient data storage, there are already research trends about the performances of the whole system when combining these trending technologies.

Reference

- [1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.
- [2] D. Vujičić, D. Jagodić, and S. Randić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," in *2018 17th international symposium infoteh-jahorina (infoteh)*, 2018, pp. 1-6: IEEE.
- [3] Z. Zheng *et al.*, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp.

475-491, 2020.

[4] P. De Filippi, M. Mannan, and W. Reijers, "Blockchain as a confidence machine: The problem of trust & challenges of governance," *Technology in Society*, vol. 62, 2020.

[5] J. Abou Jaoude and R. George Saade, "Blockchain Applications – Usage in Different Domains," *IEEE Access*, vol. 7, pp. 45360-45381, 2019.

[6] G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," presented at the 2015 IEEE Security and Privacy Workshops, 2015.

[7] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling Blockchain: A Data Processing View of Blockchain Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366-1385, 2018.

[8] Y. Xu and Y. Huang, "Segment Blockchain: A Size Reduced Storage Mechanism for Blockchain," *IEEE Access*, vol. 8, pp. 17434-17441, 2020.

[9] T. Mitani and A. Otsuka, "Traceability in Permissioned Blockchain," *IEEE Access*, vol. 8, pp. 21573-21588, 2020.

[10] O. Pal, B. Alam, V. Thakur, and S. Singh, "Key management for blockchain technology," *ICT Express*, vol. 7, no. 1, pp. 76-80, 2021.

[11] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841-853, 2020.

[12] S.-K. Kim, C. Y. Yeun, E. Damiani, Y. Al-Hammadi, and N.-W. Lo, "New blockchain adoption for automotive security by using systematic innovation," in *2019 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, 2019, pp. 1-4: IEEE.

[13] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to Scalability of Blockchain: A Survey," *IEEE Access*, vol. 8, pp. 16440-16455, 2020.

[14] R. Taş and Ö. Ö. Tanrıöver, "A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting," *Symmetry*, vol. 12, no. 8, 2020.

[15] W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li, and R. Chen, "NutBaaS: A Blockchain-as-a-Service Platform," *IEEE Access*, vol. 7, pp. 134422-134433, 2019.

[16] M. Muzammal, Q. Qu, and B. Nasrulin, "Renovating blockchain with distributed databases: An open source system," *Future Generation Computer Systems*, vol. 90, pp. 105-117, 2019.

[17] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd international conference on open and big data (OBD)*, 2016, pp. 25-30: IEEE.

[18] H. M. Hussien, S. M. Yasin, N. I. Udzir, M. I. H. Ninggal, and S. Salman, "Blockchain technology in the healthcare industry: Trends and opportunities," *Journal of Industrial Information Integration*, vol. 22, p. 100217, 2021.

[19] Z. Zhou, M. Wang, C.-N. Yang, Z. Fu, X. Sun, and Q. J. Wu,

"Blockchain-based decentralized reputation system in E-commerce environment," *Future Generation Computer Systems*, vol. 124, pp. 155-167, 2021.

[20] A. Park and H. Li, "The Effect of Blockchain Technology on Supply Chain Sustainability Performances," *Sustainability*, vol. 13, no. 4, 2021.

[21] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance Analysis of Hyperledger Fabric Platforms," *Security and Communication Networks*, vol. 2018, pp. 1-14, 2018.

[22] T. Yan, W. Chen, P. Zhao, Z. Li, A. Liu, and L. Zhao, "Handling conditional queries and data storage on Hyperledger Fabric efficiently," *World Wide Web*, vol. 24, no. 1, pp. 441-461, 2020.

[23] M. Uddin *et al.*, "Hyperledger Fabric Blockchain: Secure and Efficient Solution for Electronic Health Records," *Computers, Materials & Continua*, vol. 68, no. 2, pp. 2377-2397, 2021.

[24] 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: IEEE.

[25] T. T. Kuo, H. E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *J Am Med Inform Assoc*, vol. 24, no. 6, pp. 1211-1220, Nov 1 2017.

[26] M. Antwi, A. Adnane, F. Ahmad, R. Hussain, M. H. ur Rehman, and C. A. Kerrache, "The case of hyperledger fabric as a blockchain solution for healthcare applications," *Blockchain: Research Applications*, vol. 2, no. 1, p. 100012, 2021.

[27] M. Uddin, "Blockchain Medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry," *International Journal of Pharmaceutics*, vol. 597, p. 120235, 2021.

[28] M. Fan and X. Zhang, "Consortium Blockchain Based Data Aggregation and Regulation Mechanism for Smart Grid," *IEEE Access*, vol. 7, pp. 35929-35940, 2019.

[29] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A Novel Blockchain-Based Product Ownership Management System (POMS) for Anti-Counterfeits in the Post Supply Chain," *IEEE Access*, vol. 5, pp. 17465-17477, 2017.

[30] K. R. K. Reddy, A. Gunasekaran, P. Kalpana, V. R. Sreedharan, and S. A. Kumar, "Developing a blockchain framework for the automotive supply chain: A systematic review," *Computers & Industrial Engineering*, vol. 157, p. 107334, 2021.

[31] P. Fraga-Lamas and T. M. Fernandez-Carames, "A Review on Blockchain Technologies for an Advanced and Cyber-Resilient Automotive Industry," *IEEE Access*, vol. 7, pp. 17578-17598, 2019.

[32] S. G. Yoo and B. Ahn, "A study for efficiency improvement of used car trading based on a public blockchain," *The Journal of Supercomputing*, vol. 77, no. 9, pp. 10621-10635, 2021.

-
- [33] P. K. Sharma, N. Kumar, and J. H. Park, "Blockchain-Based Distributed Framework for Automotive Industry in a Smart City," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4197-4205, 2019.
- [34] X. Xu, L. Tatge, X. Xu, and Y. Liu, "Blockchain applications in the supply chain management in German automotive industry," *Production Planning Control*, pp. 1-15, 2022.
- [35] D. Miehl, D. Henze, A. Seitz, A. Luckow, and B. Bruegge, "PartChain: A Decentralized Traceability Application for Multi-Tier Supply Chain Networks in the Automotive Industry," presented at the 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), 2019.
- [36] K. Luchoomun, S. Pudaruth, and S. Kishnah, "Implementation of a proof of concept for a blockchain-based smart contract for the automotive industry in Mauritius," *International Journal of Advanced Computer Science Applications*, vol. 11, no. 3, 2020.
- [37] M. Kuhn, F. Funk, and J. Franke, "Blockchain architecture for automotive traceability," *Procedia Cirp*, vol. 97, pp. 390-395, 2021.
- [38] F. Lamberti, V. Gatteschi, C. Demartini, M. Pelissier, A. Gomez, and V. Santamaria, "Blockchains Can Work for Car Insurance: Using Smart Contracts and Sensors to Provide On-Demand Coverage," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 72-81, 2018.
- [39] L. R. Abbade *et al.*, "Blockchain Applied to Vehicular Odometers," *IEEE Network*, vol. 34, no. 1, pp. 62-68, 2020.
- [40] V. Ortega, F. Bouchmal, and J. F. Monserrat, "Trusted 5G Vehicular Networks: Blockchains and Content-Centric Networking," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 121-127, 2018.
- [41] X. Wang, Y. Wang, and A. Liu, "Trust-Driven Vehicle Product-Service System: A Blockchain Approach," *Procedia CIRP*, vol. 93, pp. 593-598, 2020.
- [42] N. Nizamuddin and A. Abugabah, "Blockchain for automotive: An insight towards the IPFS blockchain-based auto insurance sector," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, 2021.
- [43] P. K. Patro, R. W. Ahmad, I. Yaqoob, K. Salah, and R. Jayaraman, "Blockchain-Based Solution for Product Recall Management in the Automotive Supply Chain," *IEEE Access*, vol. 9, pp. 167756-167775, 2021.
- [44] D. Lu *et al.*, "Reducing Automotive Counterfeiting Using Blockchain: Benefits and Challenges," presented at the 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), 2019.
- [45] J. Chen, Y. Ruan, L. Guo, and H. Lu, "BCVehis: A Blockchain-Based Service Prototype of Vehicle History Tracking for Used-Car Trades in China," *IEEE Access*, vol. 8, pp. 214842-214851, 2020.
- [46] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "BlockChain: A Distributed Solution to Automotive Security and Privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119-125, 2017.
- [47] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A Blockchain

Based Truthful Incentive Mechanism for Distributed P2P Applications," *IEEE Access*, vol. 6, pp. 27324-27335, 2018.

[48] M. Turkanovic, M. Holbl, K. Kotic, M. Hericko, and A. Kamisalic, "EduCTX: A Blockchain-Based Higher Education Credit Platform," *IEEE Access*, vol. 6, pp. 5112-5127, 2018.

[49] A. Iftikhar, X. Cui, Q. Tao, and C. Zheng, "Hyperledger Fabric Access Control System for Internet of Things Layer in Blockchain-Based Applications," *Entropy (Basel)*, vol. 23, no. 8, Aug 16 2021.

[50] I. Mehar *et al.*, "Understanding a Revolutionary and Flawed Grand Experiment in Blockchain: The DAO Attack," *Journal of Cases on Information Technology*, vol. 21, pp. 19-32, 01/01 2019.

Choaxing Luo's dissertation

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43