

Blockchain-based supply chain schedule coordination for off-site construction

Xianxiang Zhao

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Construction Management

University of Washington

2023

Committee:

Yong-woo Kim

Lingzi Wu

Program Authorized to Offer Degree:

Construction Management

©Copyright 2023
Xianxiang Zhao

University of Washington

Abstract

Blockchain-based supply chain schedule coordination for off-site construction

Xianxiang Zhao

Chair of the Supervisory Committee:

Yong-woo Kim

Construction Management

Compared to traditional on-site construction, off-site construction shifts its primary workflow from the field to the factory. This kind of transformation also causes the key to success in construction switches from various field operations to supply chain management. At the same time, the participants in the construction supply chain often have a silo mindset, which means they are often reluctant to share their information to improve the overall supply chain performance. This research used the design science method to develop a prototype artifact as a possible solution for unreliable schedule planning. A streamlined supply chain workflow and a blockchain-based prototype have been built. The shared information structure has been realized in a smart contract. Compared with the traditional workflow, the proposed method gives a transparent and reliable way to improve supply chain performance. After the development of the prototype, its functionality was tested with a test environment in

the validation section. As a contribution to the current body of knowledge, this research explored the potential opportunity to implement blockchain in supply chain schedule coordination and management.

Table of Contents

Chapter 1 Introduction	1
1.1 Background	1
1.2 Research Objective	5
1.3 Scope and Limitation	5
1.4 Thesis Structure	5
Chapter 2 Literature Review	6
2.1 Construction Supply chain	6
2.1.1 Off-site construction current status	6
2.1.2 Supply chain management	8
2.1.3 Supply chain coordination in the construction industry	9
2.1.4 Rebate for Scheduled Delivery	10
2.2 Blockchain technology	11
2.2.1 Distributed ledger technology and smart contracts	11
2.2.2 Blockchain application in the construction industry	13
Chapter 3 Methodology	16
Chapter 4 Prototype Design	18
4.1 Status quo and possible solutions	18
4.1.1 Point of departure: information in supply chain	18
4.1.2 Underperformance with a silo mindset in the construction industry ..	19
4.1.3 The contractor's delivery request is not lean	21
4.1.4 The potential of blockchain technology and rebate mechanisms	22
4.1.5 Financial incentive mechanism - Rebate	22
4.1.6 Updating the reliability distribution	25
4.2 Selection of blockchain platform and consensus algorism	27

4.2.1 Blockchain protocol selection.....	27
4.2.2 Consensus mechanism	29
4.3 Participating organizations and system architecture.....	30
4.4 Smart Contract Design.....	33
4.4.1 Streamlined Business Workflow.....	33
4.4.2 Information sharing structure.....	36
4.4.3 Smart Contract Design.....	38
4.4.4 Realization of Smart Contract (Function Design)	40
Chapter 5 Prototype Validation	53
5.1 Validation Environment Setup.....	53
5.1.1 Validation Environment.....	53
5.1.2 Network Setup	53
5.2 Hypothetical Case Study.....	58
5.2.1 Initial steps	58
5.2.2 Order simulation	62
5.2.3 Comparism with the traditional fixed rebate	76
Chapter 6 Research Discussion.....	78
6.1 Research Limitation and future research	78
6.2 Research Contribution	79
6.3 Future of construction supply chain management	80
Chapter 7 Conclusion.....	82
Bibliography	84

List of Figures

Figure 1 The steps for design science research.....	17
Figure 2 Distribution of scheduled date (Kim and Rhee 2021).....	24
Figure 3 Reliability distribution (Kim and Rhee 2021).....	25
Figure 4 Consensus mechanism.....	30
Figure 5 Channel Configuration	31
Figure 6 Transaction flow of Hyperledger Fabric network	32
Figure 7 Business workflow	35
Figure 8 Data structure of the prototype	37
Figure 9 Smart contract flow integrated with business flow	38
Figure 10 Entries for the other functions	41
Figure 11 Main function	42
Figure 12 InitLedger function.....	42
Figure 13 Functions in the utils package	43
Figure 14 The function used to add a number to the string date.....	43
Figure 15 GetBaselineDate function.....	44
Figure 16 Part of the function CalculationStepOne.....	46
Figure 17 Part of the function CalculationStepTwo	46
Figure 18 Different structs in the model package.....	47
Figure 19 Part of the order struct	47
Figure 20 Functions in the fabric package.....	49
Figure 21 Access control	50

Figure 22 Case of EDI	51
Figure 23 Uploading the data on the blockchain	52
Figure 24 (left) configuration of the crypto-config.....	54
Figure 25 The configuration file of the channel	54
Figure 26 Blocks from step2 and step3.....	55
Figure 27 Test network configuration.....	56
Figure 28 Building the networks.....	56
Figure 29 Join the channel and update the anchor peer	57
Figure 30 Approval from each participant.....	58
Figure 31 The participants of the test network	59
Figure 32 The dashboard of Hyperledger Explorer	60
Figure 33 InitLedger function process	61
Figure 34 The Prior Information in the 0th order	61
Figure 35 Failure from the client side.....	62
Figure 36 Endorsement policy failure example – providing EDI.....	63
Figure 37 The returned value will be empty	63
Figure 38 Contractor uploads the EDI.....	65
Figure 39 Contractor query the optimal delivery date and rebate for rational decision	67
Figure 40 Supplier query the EDI for resource planning.....	67
Figure 41 Contractor uploads its IDD.....	67
Figure 42 Supplier queries the IDD for production schedule	68

Figure 43 Contractor uploads the revised delivery date	68
Figure 44 Transportation queries the RDD.....	68
Figure 45 Transportation is trying to skip the pre-inspection process.....	69
Figure 46 The information about pre-inspection, loading, arrival, and post-inspection	70
Figure 47 The ADI uploaded by the contractor for the reliability calculation	70
Figure 48 The rebate information queried by the bank.....	71
Figure 49 Main information of the third order.....	71
Figure 50 Complete information of the third order	72
Figure 51 Comparison of 3 different scenarios of the rebate mechanism	73
Figure 52 The consistent ledger height.....	74
Figure 53 Block relationship.....	75
Figure 54 The transaction proposed by different organizations	76

List of Tables

Table 1 Characteristics of different blockchain protocols	28
Table 2 The pseudocode for the rebate calculation and reliability updating	45
Table 3 The endorsement policy design	52

List of Equations

Equation 1 Optimal delivery date calculation equation (Kim and Rhee 2021)	24
Equation 2 Optimal rebate calculation equation (Kim and Rhee 2021)	24
Equation 3 Equation of EDI.....	25
Equation 4 Equation of ADI	26
Equation 5 Equation of updating μn	26
Equation 6 Equation of updating κn	26
Equation 7 Equation of updating αn	26
Equation 8 Equation of updating βn	26
Equation 9 Equation of σ^2	27

Chapter 1 Introduction

1.1 Background

The construction industry is moving towards prefabrication, where specific building components are manufactured off-site and then transported to the construction site for assembly. This approach allows for greater control over the quality and consistency of building materials and can reduce construction time and costs by allowing work to be completed in a controlled factory setting. However, this also means that the supply chain should be carefully managed to ensure that the components are delivered to the construction site promptly and efficiently. The success of a project requires coordination between contractors, suppliers, and other stakeholders, as well as a deep understanding of the construction process and the project's needs.

In the off-site construction supply chain, various stakeholders with different incentives collaborate toward the realization of a project (Dubois and Gadde 2001). However, according to Anna Dubois and Lars-Erik Gadde, the cooperation between the various stakeholders is not very close. Instead, a loosely coupled system is observed (Dubois and Gadde 2001). This loose structure has caused the existing construction supply chain to suffer from fragmentation and inefficient cooperation (Fulford and Standing 2014). Each participant makes decisions independently for their own benefit in the workflow.

The most common case is that the contractor often asks for an excessively early delivery date and changes it later. Because contractors typically have greater bargaining power than suppliers in construction, they often act in this manner with tacit approval and without penalty. The costs resulting from this situation are added to the cost of the product by the supplier (Kim and Rhee 2021) . Those self-serving actions are rational but, at the same time, detrimental to the joint cost of the supply chain. It is difficult to align the interest of project stakeholders and motivate each agent to make strategic decisions for the entire supply chain, especially since project participants do not trust each other and tend not to disclose all information.

Over the years, the construction industry has come up with many ways to align the interest of different independent stakeholders. In brief, the three main approaches are mergers and acquisitions(Choi and Russell 2004), long-term strategic alliance (W.L. Cheng et al. 2004), and short-term contract schemes(Cachon 2003). Contractual coordination is the most feasible approach in construction projects with prefabrication, but efficient coordination among project participants is challenging.

The effect of traditional short-term contractual schemes is very limited in the construction industry because opportunistic behavior is prevalent in the construction industry and poses a significant threat to trust in the relationships constituted by the contract, a threat not limited to fraud and falsification of historical information(Lu et al. 2016). The most sophisticated and efficient contract terms will be worthless if the information stakeholders rely on is fraudulent.

In recent years, electronic fraud has been prevalent and poses a significant threat to the trust and efficiency of today's construction industry. "In 2015, a scam involving the use of fake websites bearing the name of Japanese construction manufacturer Komatsu led to the fraudulent collection of deposits. Konecranes, a Finnish crane maker subsidiary, fell victim to identity theft and lost approximately 17.2 million euros through unauthorized payments.". Real-life cases like these make us keep exploring more suitable solutions, which is when blockchain comes into the picture(Lu et al. 2016).

The public first saw blockchain technology in 2008 with the application of Bitcoin(Nakamoto 2008). Frank Hofmann proposed that the definition of blockchain should be a distributed database that is practically immutable and maintained by a decentralized P2P network(Hofmann et al. 2017). According to previous research conducted by Qian, blockchain can shift the mindset of trust in the construction industry from relation-based to technology-based(Qian and Papadonikolaki 2020). Since relationship-based trust can be undermined by opportunism, this technology-based trust is more reliable.

Despite the discussion of blockchain's potential for use in the construction supply chain in recent studies(Yoon and Pishdad-Bozorgi 2022), little research has been done on developing smart contracts and systems for coordinating the supply chain and aligning the interests of multiple stakeholders. Furthermore, blockchain is still a relatively new concept for the construction industry, and there is a need for a study that explores how

to build an effective blockchain network to improve the overall performance of the supply chain.

1.2 Research Objective

This paper attempts to establish a blockchain prototype system to align the interests of various stakeholders. Moreover, this research mainly focuses on exploring a possible solution for supply chain schedule management in the current construction industry. In order to establish a functional prototype in a mutual distrust environment, several sub-objectives were established to achieve the ultimate goal.

1. What members should be included in a blockchain-based construction supply chain?
2. What kind of information should each member share on this platform?
3. How should the information interact with each other?
4. What are the benefits can blockchain provide in the supply chain?

1.3 Scope and Limitation

Due to time constraints and real-world limitations, this study could only develop the blockchain prototype. This study has not put the prototype into the production stage for this research. Therefore, a hypothesis case study has been used for the validation.

1.4 Thesis Structure

The rest of the thesis will be the following sections: Chapter 2 reviews the literature relevant to the topic, Chapter 3 introduces the design science methodology, Chapter 4 illustrates the prototype design procedure, Chapter 5 validates the prototype's functionality, Chapters 6 and 7 discussed the limitation and the contribution of this research, listed the possible future research.

Chapter 2 Literature Review

2.1 Construction Supply chain

2.1.1 Off-site construction current status

Off-site construction is seen as a rising topic in the construction research sector. In the foreseeable future, off-site construction will be an effective way to reduce the environmental impact of construction projects(Wang et al. 2019). According to Wang's research, the percentage of off-site construction adoption has recently been rapidly rising due to technological advances in recent years.

Off-site construction is also a critical method for the lean construction philosophy that has been promoted in the construction industry for many years. In traditional construction projects, it is not easy to accurately define the value of each participant from end to end. According to a previous case study, prefabrication can provide a way to deal with value stream fluctuations in the context of high levels of complexity(Björnfot and Sardén 2014). The study also illustrates that prefabrication can effectively redistribute a portion of the value stream from the design to other project stages. This redistribution will simplify the entire project workflow and give the industry a new opportunity to reduce waste and move forward.

As the technology matures, various governments have gradually begun to push through policies to accelerate the construction industry's run into prefabricated workflows.

China's latest policy calls for increasing the proportion of prefabricated buildings to 30% within the next ten years(The State Council 2016). On the other hand, Ireland and Scotland require new buildings to have a prefabricated component of 70% within a few years(Jiang et al. 2018). In Sweden, industrialized buildings have increased from 2% to 15% in recent years(Jansson 2010). Government intervention can often have a relatively practical impact on the construction industry. At the same time, as economies of scale come into play, the cost of prefabricated construction will gradually be reduced, further contributing to its adoption. Therefore, this study assumes that the construction industry's transition to prefabrication is an inevitable and promising trend.

As productivity levels become more stable and less vulnerable to such factors as the weather in a factory environment, the impact of supply chain reliability on project success becomes particularly significant. In wang's study, it was also shown that both push and pull contexts exist in the prefabricated supply chain(Wang et al. 2019), making the supply chain complex and even delicate. This feature requires more sophisticated and careful management of the construction supply chain. Therefore, the supply chain for off-site construction (prefabrication) needs further exploration on a detailed level.

2.1.2 Supply chain management

The concept of supply chain management has existed for a long time. However, the preliminary supply chain management in history was very fragmented. In the early days, the three stages of the supply chain, procurement, production, and distribution, were independently managed, and substantial inventory was taken as a buffer to reduce unexpected risk. Nevertheless, increased market competition and the trend of globalization are calling for a supply chain that can rapidly respond to changes in the market(Thomas and Griffin 1996). In 1996, the Supply Chain Management Council redefined the meaning of supply chain management(Larson and Rogers 1998). The definition is “effort to produce and deliver a final product from the supplier’s supplier to the customer’s customer”.

In most industry sectors, supply chain coordination often involves effective communication, information exchange, and performance measurement across multiple organizations. Because several different organizations are involved, product-specific supply chain coordination often involves actions that cross organizational and functional boundaries(Thomas and Griffin 1996). Fawcett’s research proposes that functional boundaries can lead to a lack of knowledge and information flow such that no one, including those at the top, has complete knowledge of the entire supply chain(Fawcett Stanley E. 1995). This management of knowledge, or information, has been the Achilles’ heel of supply chain management.

2.1.3 Supply chain coordination in the construction industry

The industry has come up with three leading solutions over the past decades to boost the supply chain's performance,

The first option is mergers and acquisitions, and research shows that the number of mergers and acquisitions in the U.S. construction industry is on a moderate and steady rise(W.L. Cheng et al. 2004). The significant advantage of M&A is that it can be done at a relatively low total capital cost and in a relatively short period to achieve growth on the financial statement. However, sloppy M&A does not lead to better supply chain performance, and numerous examples of supply chain performance deteriorating after M&A. Schoenberg conducted empirical research on the M&A, which shows that only about 56% of acquisitions can be considered successful against the original objectives back in 2006(Schoenberg 2006).

The second option is a long-term strategic alliance. In the construction industry, strategic alliances are often referred to as “partnering”(W.L. Cheng et al. 2004). Compared to horizontal integration with competitors, strategic alliance in the supply chain is one kind of vertical integration. Importantly, strategic alliances in the construction industry are commonly not regulated by formal contracts, and this relatively informal relationship will also make it appears non-binding when it comes to liability issues.

Concern about risk brings this research to the third option, the short-term contract scheme. Cachon suggests that the ideal optimal supply chain requires a precise set of actions to achieve, but these actions are often not in the best interests of the individual members(Cachon 2003). In order to be able to align the interests of the various members, subtle contract design and third-party transfer payments are required. Given the complexity of relationships in the construction sector, contractual coordination is the most feasible approach, excluding human interference.

2.1.4 Rebate for Scheduled Delivery

Contract-based supply chain coordination is widely studied in sectors other than construction, such as business and economics. In the economic domain, supply chain management is called channel management. Through different approaches, scholars have tried to align the interests of different stakeholders, such as markdown allowance (Pasternack 1985), quantity discount(Weng 1995), revenue-sharing(Giannoccaro and Pontrandolfo 2004), two-part tariff(Jeuland and Shugan 1983), quantity flexibility contract(Tsay 1999), wholesale and retail price protection(Lee et al. 2000), target-level sales rebate(Taylor 2002), holding cost subsidy(Wang and Gerchak 2001), capacity reservation contract(Barnes-Schuster et al. 2002), and guaranteed profit margins(Lee and Rhee 2008).

However, the prefabrication construction is so unique that the above methods are not very applicable. Many of these methods require changes to the product quantity itself,

but the “quantity of products” in the construction industry is exogenously pre-fixed in the design drawings and specifications.

Kim and Rhee proposed a rebate solution for the specific nature of the construction industry in their study (Kim and Rhee 2021). This rebate design is based on the sequential game theory, which has been proven by rigorous mathematical evidence to have great potential in practice. The introduction of game theory brings this rebate design closer to Perry’s description of supply chain coordination - coordinating the decisions of independent agents to improve the performance of the entire supply chain by managing the interdependencies between their activities (Perry 1989). This rebate design can be considered a financial incentive, that is, an incentive for credibility in making a reliable decision.

This rebate solution is very delicately designed but requires an advanced information management approach, as its incentives are effective only if the historical data and information are traceable and credible. A little misinformation and dishonest behavior can have a substantial negative impact on this reputational incentive. This fragility requires exploring a suitable technology path for such a design to achieve the intended purpose better.

2.2 Blockchain technology

2.2.1 Distributed ledger technology and smart contracts

Distributed ledger technology has excellent potential for the supply chain and information problems this study has encountered. This technology can effectively avoid information fraud and provide a traceable source of information for the supply chain. Unlike a centralized database, distributed ledger technology is a particular case of a distributed database. Traditional databases usually use a centralized database, which stores data on a single storage device. Distributed databases, on the other hand, store data in multiple devices that are physically independent and interconnected with each other. Even though they are physically separate, you will get the same result when you do operations on the distributed database. To conclude, such a distributed database is decentralized in form and unified in logic(Sunyaev 2020).

In order to be logically consistent, distributed databases need the help of a consensus mechanism. Consensus mechanisms often use algorithms to ensure that distributed databases remain consistent in their responses to the same operations(Sunyaev 2020).

Distributed ledger, an advanced distributed database, only allows the addition of new data and does not allow the deletion or modification of existing data(Sunyaev 2020).

At the same time, the distributed ledger still does not require a central authority to manage it. In 2008, Satoshi Nakamoto came up with a great idea to combine the techniques of game theory and cryptography to get one of the most successful distributed ledgers to date, the Bitcoin blockchain(Nakamoto 2008).

Cryptocurrencies such as Bitcoin provide a solution for transferring value between untrusted entities. Cryptocurrencies are often classified as Blockchain 1.0. One step further, Blockchain 2.0 proposes smart contracts to automate the business related to cryptocurrencies and even create virtual assets. However, in Blockchain 2.0, the use of smart contracts was restricted, and the industry realized that many non-cryptocurrency businesses could also benefit from the blockchain (distributed ledger technology), so the concept of Blockchain 3.0 was born. Since 2008, the boom in blockchain has led to the rapid maturation of distributed ledger technology. This technology is now combined with smart contracts in the non-cryptocurrency context, bringing decentralized and disintermediated systems to various industry sectors (Di Francesco Maesa and Mori 2020).

2.2.2 Blockchain application in the construction industry

Due to the growing maturity of blockchain technology (distributed ledger technology), research on its application in the construction industry has been increasing in recent years. Yoon and Pishdad-Bozorgi conducted a literature review on blockchain applications in 2022 about the construction supply chain, showing that blockchain and smart contracts as a new technology can be an effective solution to some of the challenges in the construction supply chain (Yoon and Pishdad-Bozorgi 2022). Since blockchain research in the field of the construction supply chain is minimal, there is a need to review blockchain applications in the construction industry as a whole to give us some ideas.

Zhang has developed a framework for tracking construction quality, a tool that proposes a solution to quality problems in the supply chain(Zhang et al. 2020). This research attempts to solve common construction quality disputes by leveraging the tamper-evident nature of distributed ledger technology. Katharina presents a potential workflow for managing the problem of late payments in the construction industry in 2021(Sigalov et al. 2021). This research automates the payment through the execution of smart contracts, but its use of a public chain has a high gas cost and is unsuitable for widespread adoption. Moumita proposed an integrated document management system in 2022. This solution is based on a consortium blockchain system, Hyperledger Fabric, which gives the whole blockchain system more application advantages.

Although there has been some exploration of blockchain technology in the construction industry, it is still in its early stages. Furthermore, these studies have not discussed setting up an incentive mechanism based on game theory among participants to automate the distribution of game incentives in a decentralized system to motivate the overall supply chain performance to improve continuously and positively.

The tamper-proof nature of the blockchain allows it to provide an unparalleled and ideal environment in the context of game theory, which is exactly what is needed for the rebate mechanism. The success rate of opportunism and speculation, which would have previously disrupted reliability, can be significantly reduced by technological constraints, thus effectively providing a suitable environment for a sequential game.

Chapter 3 Methodology

This research will use the methodology of design science. In 1988, Herbert Simon distinguished between natural sciences, which explain how things currently are, and design sciences, which focus on how things should be (Simon 1988). Design science is a research method concerned with creating new artifacts, such as designs, theories, methods, and systems (Dresch et al. 2015). The design science method approaches problem-solving that emphasizes the practical application of knowledge and the creation of artifacts to address real-world challenges. Unlike the traditional scientific method, which aims to understand and forecast natural events, the design science method focuses on developing and evaluating solutions to specific problems. The process typically includes identifying the problem, creating a solution, implementing the solution, and evaluating its effectiveness.

This study used an experiment to complete the last step in design science, evaluation, by doing a hypothesis case study to validate the feasibility of our proposed blockchain prototype.

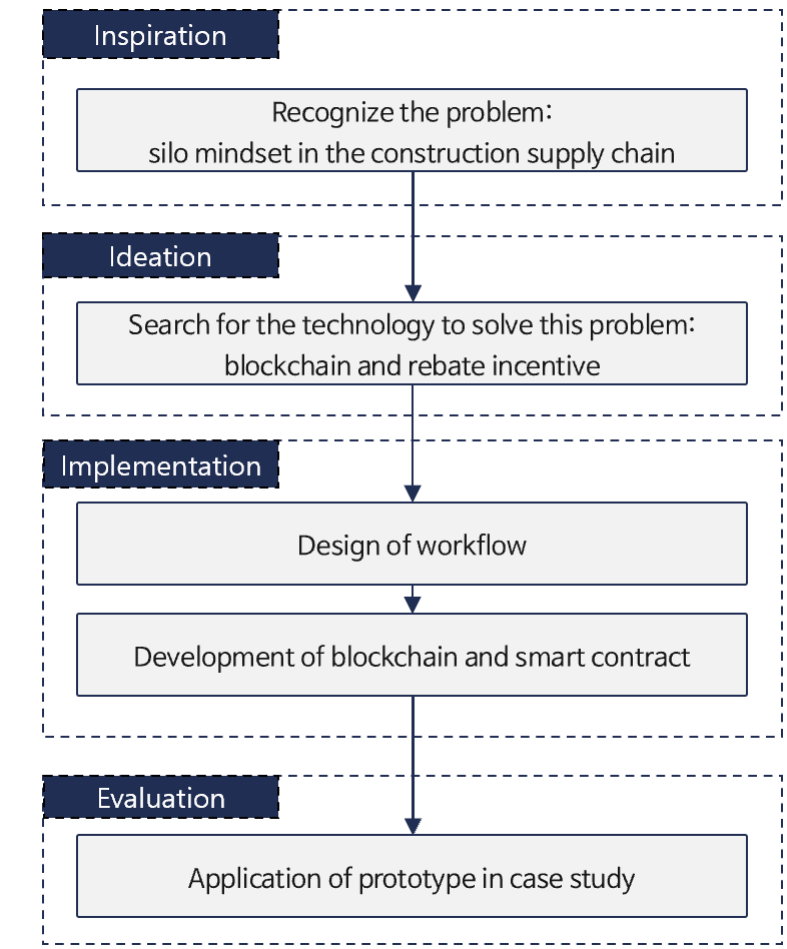


Figure 1 The steps for design science research

The contribution of design science research should be to suggest a possible solution(usually an artifact) to the problem known to the industry. The limitation of design science is that this method cannot provide evidence to support developed hypotheses, but it can help to form them and weed out those that are not worth pursuing in the future(Bider et al. 2012).

Chapter 4 Prototype Design

4.1 Status quo and possible solutions

This section presents a comprehensive review of the current literature in the relevant field, which serves as the foundation for the research presented in this study. By examining previous studies, theories, and findings, this research seeks to build upon and contribute to the existing body of knowledge in the field.

4.1.1 Point of departure: information in supply chain

Recent study shows there are six uprising area in the construction supply chain research(Nguyen and Le 2022), which include logistics and supply chain management for prefabricated construction, construction procurement, integration of construction safety and health management, supply chain management for green construction, reverse logistics in construction, and onsite construction logistics.

Among all those researchs conducted, modeling and simulation is one of the major method rising in supply chain management. One of the topic in this domain is about information sharing, Jung conducted a research about simulating the information addressing the importance of real-time information sharing to practitioners(Min and Bjornsson 2008). Another research published recently builds and validates a system dynamic model to measure the effect of ineffective information dissemination on the project's timeline(Wu et al. 2022b). Wu's research also introduces the difference of

information flow between manufacturing and construction industry. The information flow in a construction supply chain for a modularized heavy industrial construction project is more complicated than that of a typical manufacturing supply chain due to unique and customer-oriented deliverables. Construction supply chains require a large amount of information distribution and communication back and forth between stakeholders to successfully deliver the project, and stakeholders in the construction supply chain contribute to the final product and add value to the project deliverables, unlike in the manufacturing supply chain where the retailer distributes the products and increases the product's price to make non-value-added profits. These features give this research the point of departure to seek a possible solution for the industry.

4.1.2 Underperformance with a silo mindset in the construction industry

In the construction supply chain, the players often have a silo mindset and are reluctant to share valuable information. This scenario is a systemic problem in the construction industry and is caused by the loose and fragmented structure of the whole industry (Lu et al. 2016).

The silo mindset also makes it difficult for the construction supply chain to adopt the approach of long-term strategic alliances from other industries' experience. Revenue sharing and profit sharing are rarely seen in the past few decades. Therefore, a contract-based cooperation approach is more suitable and practical in the construction industry.

Nevertheless, traditional contracts often have complex items, and the relationship between the various items in the contract is intricate and not intuitive enough. It often takes a long time for people to execute these contracts, and various problems often arise in the execution stage. Given the current status of the silo mindset within the construction supply chain, the participants are more inclined to look for a reputable third party as an arbiter to resolve these issues. This approach to problem-solving is both time-consuming and effort-consuming, as well as not in line with the current trend of lean construction.

4.1.3 The contractor's delivery request is not lean

In the construction industry, project schedules are often the primary concern of on-site construction teams. Due to the distrust of suppliers and the need to avoid risks to the construction schedule, contractors are often willing to strategically advance the delivery time of the required products as much as possible. The contractors do not want delays in the construction schedule due to unexpected reasons. Nevertheless, with the development of the prefabricated supply chain, the risk of being on the supplier's end of the equation is diminishing. According to the lean construction philosophy, the contractor's conservative strategy has led to unnecessary waste. At the same time, contractors have insufficient incentive to improve their poor schedule reliability. Poor schedule reliability on the construction side has plagued the entire construction supply chain.

In many cases, contractors will request an overly ambitious delivery date and then alter it later. Those actions are often accepted and unpunished, as contractors tend to have more negotiating power than suppliers in the construction industry. As a result, the supplier bears the additional costs incurred from this situation as part of the product's overall cost. The current situation in the construction supply chain is not yet practicing the concept of lean. Therefore, the supply chain is looking for an incentive mechanism to motivate both ends of the supply chain and thus reduce waste in the value chain. In conjunction with the current state of construction, this mechanism should be a contract-

based collaboration approach to ensure enforcement in a construction industry full of defaults.

4.1.4 The potential of blockchain technology and rebate mechanisms

Contract-based collaborations are often prone to difficulties in enforcement, and parties tend to argue about the accuracy of evidence constantly and whether it has been tampered with. A particular prefabrication case was found in 2017 (Zhong et al. 2017). Inaccurate, delayed data makes decision-making extremely difficult and challenging to build trust in the system.

Numerous kinds of research have proven that blockchain technology can guarantee the authenticity of data and build trust (Zavolokina et al. 2019). The foundation of trust in blockchain lies in transforming relation-based trust into technology-based trust. Therefore, the probability of surprises and fluctuations is lower for the blockchain than for people.

Blockchain technology empowers all stakeholders in the construction supply chain with a fair and equitable platform. Also, the ability of smart contracts in the blockchain to automate many traditional operations allows prudent supply chain control to be put in place. With blockchain technology as a reliable and trustworthy “agent”, this research introduced rebate incentives to improve overall supply chain performance in the next section.

4.1.5 Financial incentive mechanism - Rebate

This research takes a rebate mechanism proposed by Kim and Rhee as a possible way for coordination between the most critical participants in the supply chain, the supplier and the contractor(Kim and Rhee 2021).

This mechanism gives a financial incentive to the supplier: The supplier will provide a price discount if the contractor agrees to accept the shipment according to the original delivery schedule. Due to cost savings, the contractor is incentivized to improve their plan reliability according to the original schedule. The supplier's inventory cost will also decrease if the contractor becomes reliable due to rebates. This mechanism encourages both parties to work together to meet the agreed-upon delivery schedule promptly and efficiently.

Rebate mechanism explanation

From Dr. Kim and Dr. Rhee's working paper, the contractor places an order at a certain period t_0 and requests for the products to be delivered at a scheduled date t_d . Once the supplier receives the order at t_0 , the supplier immediately begins production to meet the scheduled time. However, if the contractor is unreliable on its given t_d , t_d can be described as a probability distribution $F(t)$ with density $f(t)$ from the t_l to t_h . (Kim and Rhee 2021)

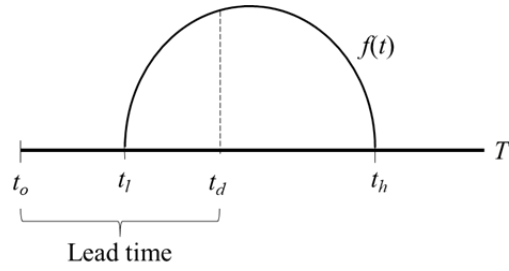


Figure 2 Distribution of scheduled date (Kim and Rhee 2021)

In this mechanism, two parameters are known in advance and fixed:

Supplier's unit inventory cost: ξ

Contractor's unit delay cost: η

The financial incentive rebate δ_i^* is calculated through the probability distribution:

$$t_{di}^* = F^{-1}\left(\frac{\xi}{\xi + \eta}\right)$$

Equation 1 Optimal delivery date calculation equation (Kim and Rhee 2021)

$$\delta_i^* = \frac{\eta F(t_{di}^*)}{f(t_{di}^*)}$$

Equation 2 Optimal rebate calculation equation (Kim and Rhee 2021)

This rebate mechanism is dependent on the estimation of the probability distribution.

However, due to the lack of research on the distribution of contractors' reliability, trying to predetermine this distribution before the actual workflow is complex and imprecise.

A self-updating process is needed to estimate the distribution in this study.

In the case of an unknown construction supply chain, the study can only get its statistical parameters by Bayesian statistics.

4.1.6 Updating the reliability distribution

This research assumes that the $F(t)$ is a truncated normal distribution(2sigma), and its variance is consistent during one construction project. The study proposes a way to estimate the distribution using data collected from different orders in a construction project.

In this study, the probability distribution is updated each time by the earliest possible installation date (EDI) and the actual installation date (ADI) to approach its distribution gradually.

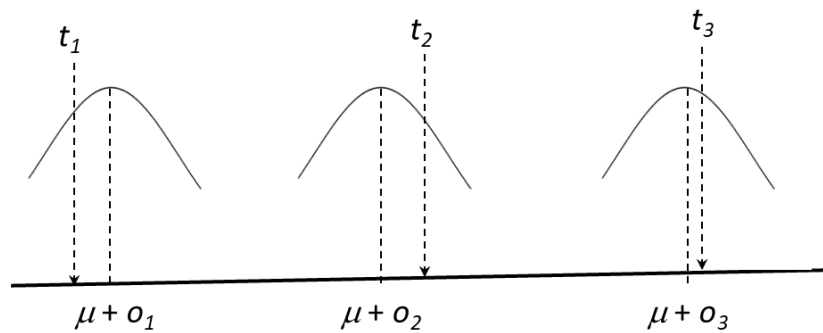


Figure 3 Reliability distribution (Kim and Rhee 2021)

$$t_{ii}(EDI) = \mu + o_i - 2\sigma$$

Equation 3 Equation of EDI

Furthermore, the t_i from the i -th batch follows the normal distribution.

$$t_i(ADI) = t + o_i$$

Equation 4 Equation of ADI

Thus, each observation of t_i is adjusted by shifting the i -th batch distribution to the first one. Then multiple observations will be obtained, several t_a , forming a single collapsed distribution.

After the collapse, both unknown μ and σ^2 will be estimated.

Let $\tau = 1/\sigma^2$

$$\mu, \tau | x_1, x_2, \dots, x_n \approx \text{Normal} - \text{Gamma}(\mu, \tau | \mu_n, \kappa_n, \alpha_n, \beta_n)$$

$$\mu_n = \frac{\kappa_o \mu_o + n \bar{x}}{\kappa_o + n}$$

Equation 5 Equation of updating μ_n

$$\kappa_n = \kappa_o + n$$

Equation 6 Equation of updating κ_n

$$\alpha_n = \alpha_o + \frac{n}{2}$$

Equation 7 Equation of updating α_n

$$\beta_n = \beta_o + \frac{1}{2} \sum_{i=1}^n (x_i - \bar{x})^2 + \frac{\kappa_o n (\bar{x} - \mu_o)^2}{2(\kappa_o + n)}$$

Equation 8 Equation of updating β_n

Since each order needs to have its rebate price calculated, let $n=1$ here.

When $n = 1$, $\kappa_n = \kappa_o + 1$, $\mu_n = \frac{\kappa_o \mu_o + x_i}{\kappa_n}$, $\alpha_n = \alpha_o + \frac{1}{2}$, $\beta_n = \beta_o + \frac{\kappa_o(x_i - \mu_o)^2}{2\kappa_n}$

The mean of $\tau = 1/\sigma^2$ is α/β , and then the mean of σ^2 is

$$\hat{\sigma}^2 = \frac{\beta}{\alpha}$$

Equation 9 Equation of $\hat{\sigma}^2$

$E(t_i) = \mu + o_i$ (the expected time of the module installation)

$$Var(t_i) = \sigma^2$$

Since the distribution follows normal distribution here, the rebate can be calculated as

Kim and Rhee describes (Kim and Rhee 2021):

$$t_{di}^* = F^{-1}\left(\frac{\xi}{\xi + \eta}\right) \text{ and, therefore, } \delta_i^* = \frac{\eta F(t_{di}^*)}{f(t_{di}^*)} \text{ (Equation 1 and Equation 2)}$$

4.2 Selection of blockchain platform and consensus algorism

4.2.1 Blockchain protocol selection

Blockchain protocols can be classified into two main categories: public and private.

(Yang et al. 2020). Detailed comparisons are listed in the following Table 1.

Mode	Characteristics

Public Blockchain	<ol style="list-style-type: none"> 1. No access control; vulnerable to malicious participants 2. Related to cryptocurrencies; vulnerable to fluctuations 3. No privacy; data is open to everyone
Private Blockchain (Consortium)	<ol style="list-style-type: none"> 1. Access control; available only for the certified user. 2. De-tokenization; more stable 3. Privacy data protection; business-friendly

Table 1 Characteristics of different blockchain protocols

After comparison, the study finds that the characteristics of private blockchain are more suitable for building our supply chain coordination platform. Consortium blockchains are the most widely used protocol among private blockchains. Although the success of public blockchains and their large user base demonstrate their dominance in the blockchain world, it is less stable because it is highly related to cryptocurrency. This kind of variability is not what this study is looking for. Consortium blockchains stand out as one of the most commonly used by other industries. The consortium blockchain protocol is well-suited for use cases where multiple organizations must collaborate and share data.

Hyperledger Fabric is one of the most popular protocols among the many consortium blockchains(Wu et al. 2022a). It allows for secure and transparent direct information

transactions to occur through the use of a peer-to-peer network, adding an element of trust to the system's efficiency.

The whole supply chain considers its data a critical asset to avoid leaks to competitors that could result in economic losses. Hyperledger Fabric uses channels to create private networks where participants can share sensitive information without it being visible to the rest of the network. On a channel, the access to the network is controlled, meaning that even if participants may not completely trust each other (in this study, their interests are not aligned), the network with channels can still function using a governance model that relies on the trust that does exist among participants, like a legal contract or a system for resolving conflicts.

On the other hand, Hyperledger Fabric allows for the creation of smart contracts, also known as chaincode, which can encode business logic and automate processes. This feature is expected to enhance the supply chain platform of the study significantly.

4.2.2 Consensus mechanism

Hyperledger Fabric offers a series of consensus options, such as Solo, Kafka, and Raft. These consensus algorithms are used to order the data-related operations within the channel. This activity is called the ordering service. Ordering service is the core service for a blockchain. Raft-based ordering services are simpler to do the configuration and

maintenance compared to Kafka-based ordering services, and they are designed to allow different organizations to contribute nodes to a shared ordering service.

Regarding performance, the Raft has significantly improved over the Kafka consensus without starting an additional Zookeeper service to manage the node collection. So as a future-oriented coordination system, Raft is the best choice for this study. The primary function of Raft is to validate and record transactions on a distributed ledger and to ensure the integrity and security of the ledger(Figure 4).

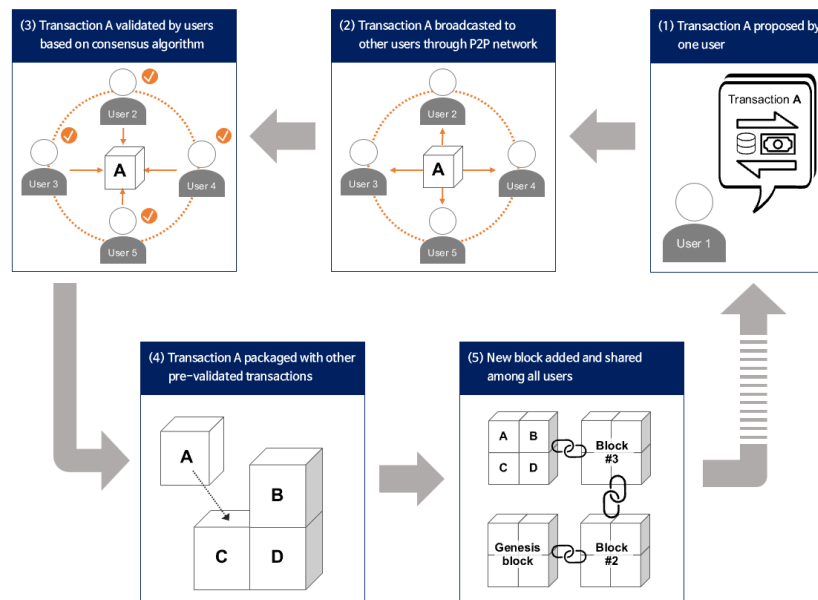


Figure 4 Consensus mechanism

4.3 Participating organizations and system architecture

To better align with the real-world construction supply chain, this study includes five organizations in the design of the blockchain platform: supplier(S1), transportation(T2), contractor(C3), inspector(I4), and bank(B5). Each organization's certificate authority(CA1-5) represents its identity(digital signature). The blockchain network architecture is shown in Figure 5.

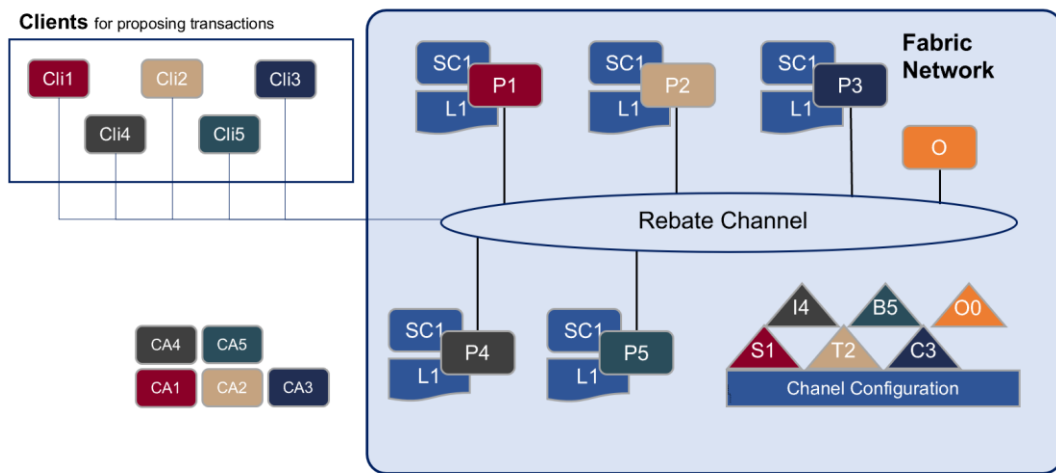


Figure 5 Channel Configuration

Each organization needs at least one peer(P1-P5) responsible for interacting with others within the blockchain in real time. Furthermore, each organization controls a client(Cli1-5) to interact with the network through the corresponding peers. All organizations share the ledger(L1) in the same channel(Rebate Channel), which can be used as an effective mechanism for data isolation between projects. For example, if the network has another project, it can have another channel with a different ledger. Furthermore, in the fabric network, smart contract controls the read and write access to their shared ledger(SC1). The smart contract is installed on each of the peers to

automate the business process. Each organization is designed to share valuable information about the construction supply chain in the network. In this study, the relevant parties are set to endorse the authenticity of each uploaded information using the certificate authority(CA1-5) before committing to the blockchain.

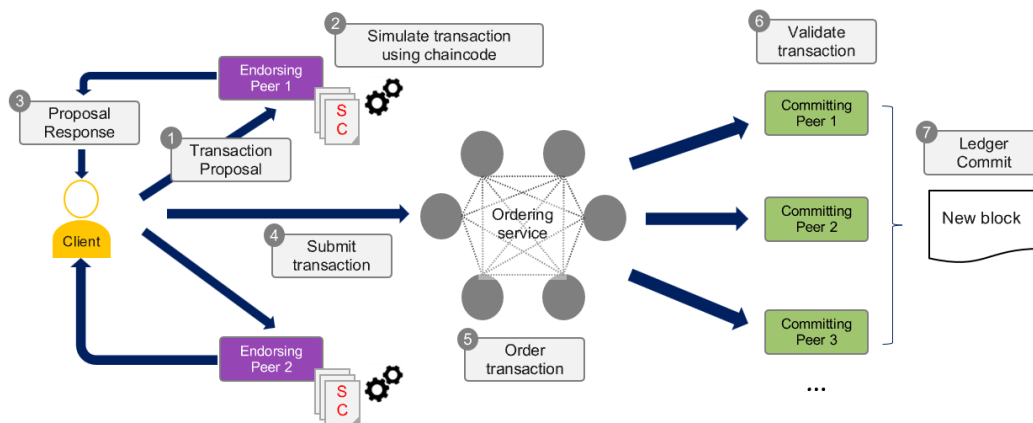


Figure 6 Transaction flow of Hyperledger Fabric network

Figure 6 is a simulation of the execution process of the smart contract. When an organization in the blockchain wants to add information to the shared ledger, it will file a transaction proposal to the endorsing peers(related parties). Those assigned endorsing parties will check the request through the smart contract(chaincode) and endorse the transaction through their certificate authority(CA). Once enough endorsements have been collected, this transaction will be submitted to the ordering peer(O) for ordering and updated to each peer in the channel. Immediately afterward, each peer will double-

check that there is no problem with the transaction, after which it updates its shared ledger.

4.4 Smart Contract Design

The smart contract is a core part of blockchain 3.0(Di Francesco Maesa and Mori 2020), serving as a medium for the outside world to interact with the blockchain network and responsible for the process of data. Smart contracts usually act as a specification and have a mandatory role in regulating the data interaction of individual blockchain members.

Smart contracts allow for the automation of contract execution and enforcement, reducing the need for intermediaries and increasing the speed and efficiency of executing and enforcing agreements. Because they are stored on a blockchain, smart contracts are also transparent and immutable, providing a record of the agreement that is verifiable and cannot be altered.

4.4.1 Streamlined Business Workflow

For the industry to adopt our proposed prototypes without too much extra conversion cost, the prototype needs to be as close as possible to real-world business processes. Therefore, real-world supply chain scenarios have been analyzed, and an 11-step pull-based workflow has been proposed after brainstorming(Figure 7). The proposed workflow is organized around individual orders, with each order treated as a separate unit.

When each order starts, the contractor will submit their order request and share their information about the installation plan(Step 1). Once the supplier has received the order request from the contractor, they will start the resource planning for that order accordingly(Step 2). At the same time, the prototype will automatically give the price of the rebate and the recommended date for the contractor to view. The contractor will make a rational decision based on the prototype's suggestion and set a delivery plan (Step 3). After this delivery plan is uploaded and sent to the supplier, the supplier will start the order preparation(Step 4). When the installation date is approaching, the contractor will reconfirm that the initial delivery plan is in line with their weekly construction plan and upload the revised delivery plan(Step 5). After the contractor has uploaded its revised delivery decision, the order will proceed to logistics and inspection. The order will go through pre-inspection(Step 6), transportation(Step 7), and post-inspection(Step 9) after being delivered to the construction site(Step 8). The inspector will be responsible for conducting inspections of the products during this period, and the two inspections(Step 6&9) will help to prevent liability issues during transportation. When the product is delivered to the contractor (prior to post-inspection), the contractor will sign off on this delivery(Step 7). After completing the post-inspection(Step 9), the contractor will install the product at the site. At this point, the contractor will also share their information about the actual time of installation (Step 10). After the contractor completes the information sharing, the bank will

proceed with the order payment transaction about this order(Step 11). After step 11, the order lifecycle is finished and ready to move on to the following order.

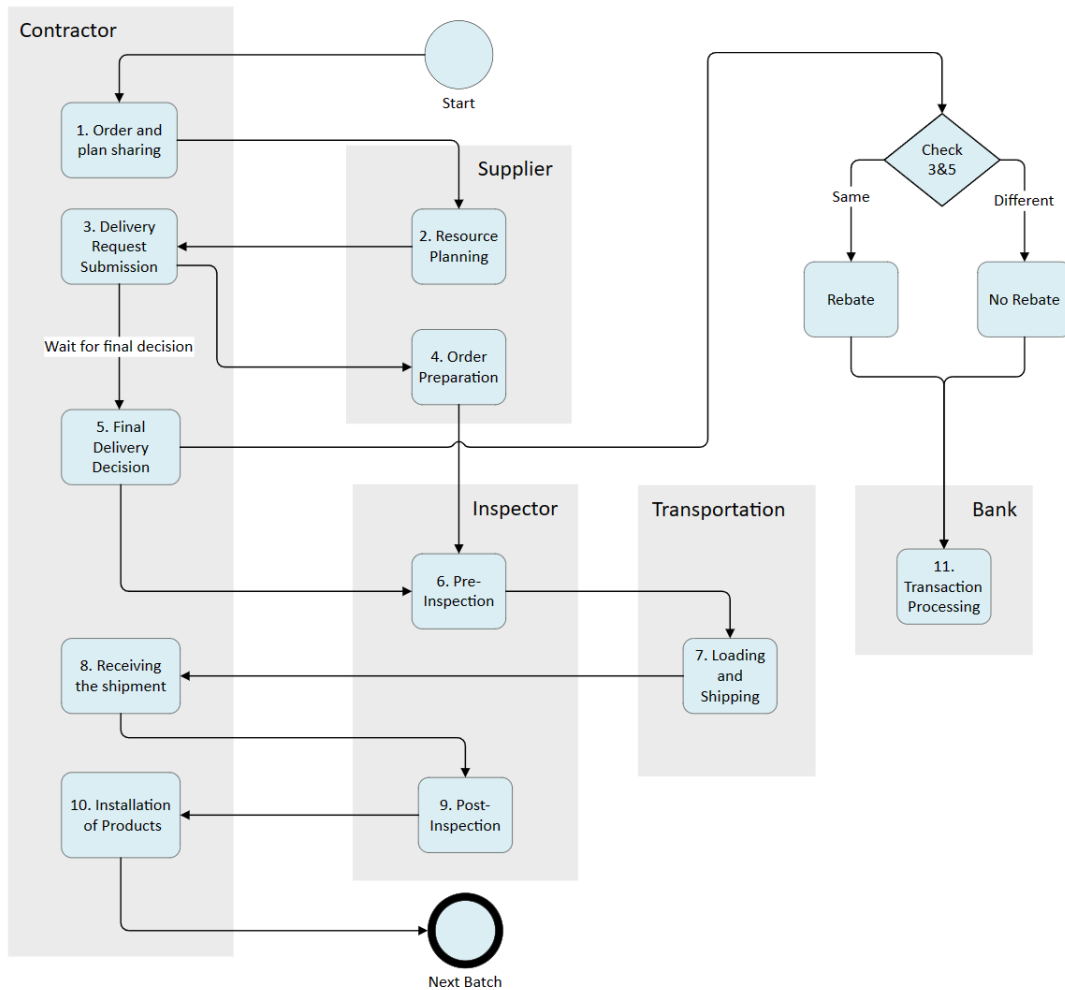


Figure 7 Business workflow

A streamlined, interrelated business process is proposed based on the modeling and optimization of real-world processes. This workflow involves the operations of the various parties involved in sharing information about an order, which needs to be further discussed.

4.4.2 Information sharing structure

The data size of the supply chain coordination platform proposed in this study is “relatively” portable. Therefore, the platform proposed by this study stores all the data on the blockchain. Compared to using off-chain data storage, this approach will protect the information when some organizations experience data corruption and loss by restoring from other organizations. Moreover, all historical versions of the data are stored on the blockchain so that users can quickly review them. As a prototype for sharing and coordinating data in the supply chain, it is essential to determine what kind of data will be provided by the participating parties in this consortium.

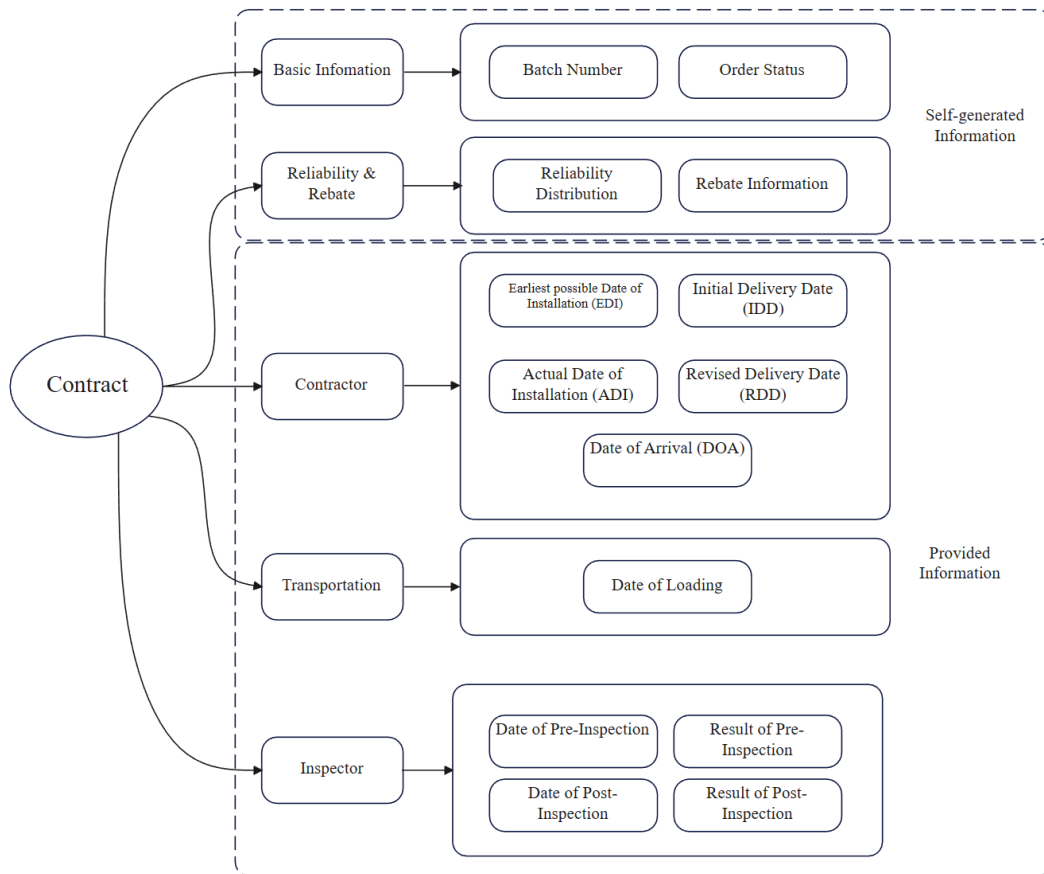


Figure 8 Data structure of the prototype

Figure 8 listed above shows the data structure designed for five participating stakeholders. As shown in Figure 8, there are two types of information in the blockchain platform: supply chain self-generated information and valuable information provided by each supply chain participant. Because this study focuses on schedule coordination, most of the shared information pertains to dates.

The most important participant is the contractor, who is required to provide five types of information regarding schedule coordination. Among this information, the earliest possible date of installation(EDI) and the actual date of installation(ADI) are the most critical ones, which constitute the source for calculating and updating the rebate incentive. The initial delivery date(IDD) is the reference for the delivery plan offered by the contractor, and the revised delivery date(RDD) will be used as a basis for the fixed decision on the order delivery request. All the above information will generate several subsequent details, like intermediate variables related to the rebate mechanism(reliability distribution and rebate information). That information will be recorded in the blockchain and provide a reference for the bank to do the transaction. The date of arrival(DOA) and other information provided by the transportation and inspector will be used to increase the transparency of the logistic procedure and quality control. In addition, the system will also provide two data points: the order batch number and the order status, to facilitate the operation of the smart contract.

4.4.3 Smart Contract Design

In this section, the study analyzes and proposes a smart contract flow to the business processes presented in the previous section (4.4.1). This analysis requires further deconstructing the business operations into operations on the data structure (4.4.2). Here, a total of 16 steps have been proposed to explore the role that smart contracts can play, including data manipulations and business operations.

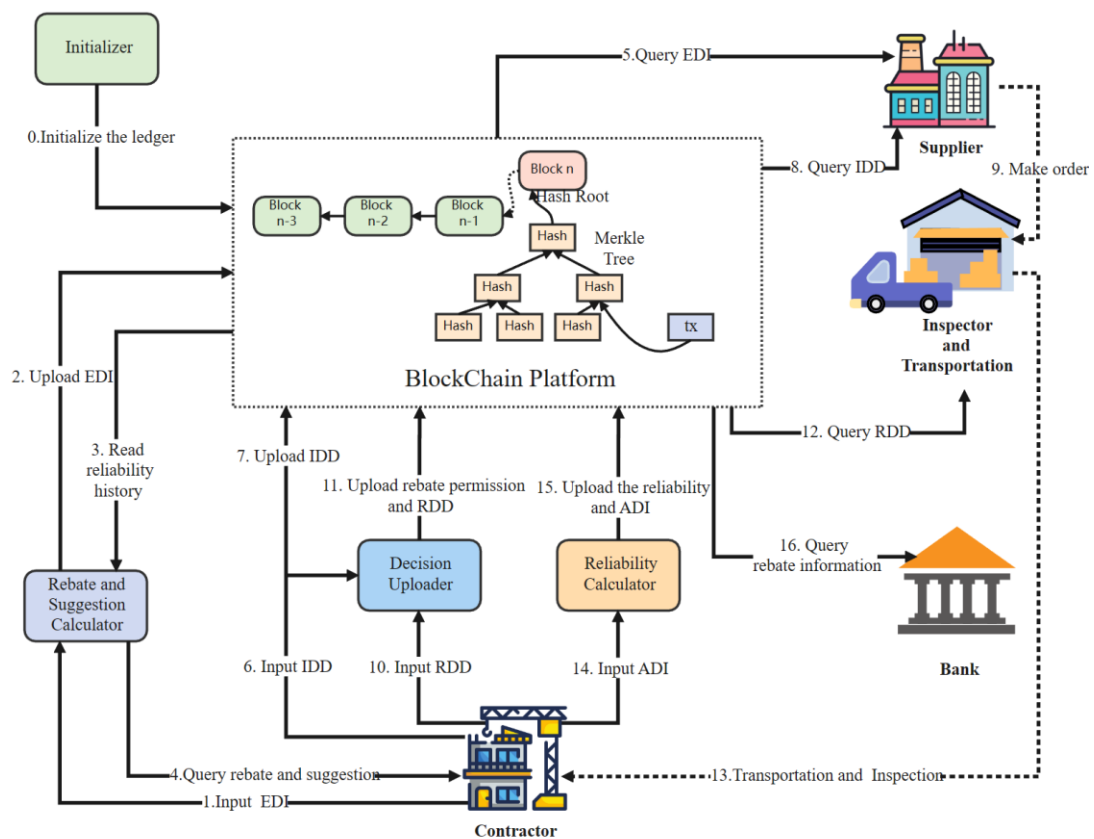


Figure 9 Smart contract flow integrated with business flow

From the initialization of the system, the initialization process will set certain system rules and serve as a reference for subsequent steps. Afterward, contractors will upload

the EDI, and the smart contract will calculate suggestions (optimal installation dates) and rebates based on the EDI and historical data. At the same time, suppliers will arrange production resources based on the EDI in the system. Afterward, contractors will upload their IDD as a reference for the supplier's production. When the RDD for the order's corresponding contractor's construction week plan is determined, this information is used as a basis for evaluating the contractor's reliability and stores rebate permission information in the system.

Meanwhile, transportation will receive the information and go to the supplier to pick up the order and start the transportation and inspection process. After the post-inspection and installation are completed, the contractor will upload the ADI to the system to update the reliability distribution. The entire process will finally be completed by the bank, which will query rebate information in the system and issue the financial incentive.

After proposing this smart contract flow, this study proposed a smart contract with several functions to realize this workflow.

4.4.4 Realization of Smart Contract (Function Design)

The smart contract in this research follows the modular programming concept. All the functions and struct(an aggregate data type) are being designed into different packages.

There are 4 different packages in this smart contract: Main, utils, model, and fabric.

The main package is the entry to the smart contract and serves as the “counter” of the whole coordination service. As a support to the “counter” in the main package, the fabric package contains the “clerks” who mainly handle the business. However, just like the clerks in the real world, those functions in the fabric package also need some tools to boost their productivity, which is the utils package. Furthermore, several structs are standardized and ready for invocation in the model package. Those structs are designed to aggregate different data types into one entity and streamline the smart contract process.

In the following, this article will introduce the contents of each package in turn, starting with the main package.

Main package

The main package is the overall entry point to the smart contract of this study, and it is designed into one single file that contains several different functions. As shown in Figure 10, most of the functions in this package are entries to the functions in the fabric package.

```
// ---Contractor ---Submit
func (s *SmartContract) ContractorSubmit(ctx contractapi.TransactionContextInterface, BatchNo int, operator string, args string) (string, error) {
    return fabric.SubmitByContractor(ctx, BatchNo, operator, args)
}

// ---Inspector ---Submit
func (s *SmartContract) InspectorSubmit(ctx contractapi.TransactionContextInterface, BatchNo int, operator string, args0 string, args1 string) (string, error) {
    return fabric.SubmitByInspector(ctx, BatchNo, operator, args0, args1)
}

// ---Transportation ---Submit
func (s *SmartContract) TransportationSubmit(ctx contractapi.TransactionContextInterface, BatchNo int, operator string, args string) (string, error) {
    return fabric.SubmitByTransportation(ctx, BatchNo, operator, args)
}

// Query the ledger (Bank)
func (s *SmartContract) QueryByBank(ctx contractapi.TransactionContextInterface, BatchNo int) (*model.BankResult, error) {
    result, err := fabric.QueryByBank(ctx, BatchNo)
    return result, err
}
```

Figure 10 Entries for the other functions

Despite the entries in the main package, the main function is the start of the chaincode life cycle. When our chaincode is installed, the main function will be called first, and the statements inside the function will be executed in the order they are written. In the main function, we will first create a new chain code through the NewChaincode function of the contractapi package, which is the first step of the smart contract on the blockchain.

```
// Main function for the Chaincode
func main() {
    // Define a new Chaincode
    chaincode, err := contractapi.NewChaincode(new(SmartContract))
}
```

Figure 11 Main function

In addition to the main function, there is another function written in the main package, which is the InitLedger function. The InitLedger function will be called on the first invocation of the chaincode. This function will read the initialization data from the model package and write it as the initial state of the ledger. The initialization of the coordination system can only be performed by the supplier. The identity of the proposer is checked in this function by the GetMSPID function.

```
// Initialize the supply chain coordination system
func (s *SmartContract) InitLedger(ctx contractapi.TransactionContextInterface) error {

    // Check the user ID
    mspid, _ := cid.GetMSPID(ctx.GetStub())
    if mspid != "SupplierMSP" {
        return fmt.Errorf("you are not the supplier, you have no permission to initialize the ledger")
    }

    // Load the init data from the smart contract
    orders := model.GetInitStateData()
    // Define the starting batch number
    FirstBatch := 1

    // Upload to the Fabric Blockchain
    for _, order := range orders {
        orderAsBytes, _ := json.Marshal(order)
        // Upload the init data
        err := ctx.GetStub().PutState(order.BatchNo, orderAsBytes)

        // Check idiomatic Error
        if err != nil {
            return fmt.Errorf("failed to upload to the world state")
        }
    }
}
```

Figure 12 InitLedger function

utils package

The utils package has 7 different functions, as shown in Figure 13. In this package, usually, the functions are designed for utility usage. One example is shown in the following Figure 14.

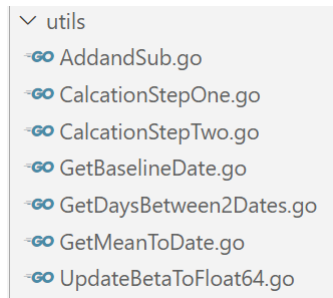


Figure 13 Functions in the utils package

```
// Addition and Subtraction
func GetDateAddInt(format, srcData string, day int) (string, error) {
    date, err := time.ParseInLocation(format, srcData, time.Local)
    if err != nil {
        return "", err
    }
    date = date.AddDate(0, 0, day)
    // [:19] represent the format of the date, in order to keep the date only
    // 2015-09-15 00:00:00 +08000CST => 2015-09-15 00:00:00
    return date.String()[:19], nil
}
```

Figure 14 The function used to add a number to the string date

These functions are the assistant functions. In the rebate calculation, many interactions will be needed to be done in the context of dates. To the computer, the date in a string data type and the days in an int type cannot do mathematical calculations. There are other similar functions that lie in the utils package.

Another critical function of the utils package is GetBaselineDate. This function can do the search in the distributed ledger and give you the baseline date of the rebate calculation.

```
// Get the baseline date
func GetBaselineDate(ctx contractapi.TransactionContextInterface) (string, error) {

    // Baseline date should get from the EDI of the 1st batch
    BatchNo := strconv.Itoa(1)
    order := new(model.Order)
    orderAsBytes, err := ctx.GetStub().GetState(BatchNo)

    if err != nil {
        return "", err
    }

    _ = json.Unmarshal(orderAsBytes, order)

    return order.EDI, nil
}
```

Figure 15 GetBaselineDate function

Apart from these “tool” functions, CalcationStepOne and CalcationStepTwo are the essential functions in the utils package. These two functions automate the process of updating rebates and reliability distributions.

rebate calculation and reliability distribution updating

The most significant part of the utils package is the rebate calculation and updating the reliability distribution. These two functions will be called when the contractor shares the corresponding valuable information. The following table (Table 2) shows the process and timing of the rebate calculation and updating process.

Pseudocode: Rebate coordination & distribution updating (*EDI*, *ADI*)

Input: *EDI*: contractor's earliest possible date of installation

ADI: contractor's actual date of installation

```
1 // Start the order lifecycle
2 EDI = getFromContractor ()
3
4 // Get the reliability history and rebate parameters
5 dist, cum_diff,  $\xi$ ,  $\eta$  = getFromBlockchain ()
6
7 // show the incentive to the contractor
8 optimalDeliveryDue = suggestionCalculator (EDI, dist)
9
10 rebate = calculateRebate (dist,  $\xi$ ,  $\eta$ )
11 uploadRebateInformationToBlockchain(rebate, optimalDeliveryDue)
12
13 // give the information to the Contractor
14 rebateInformationSendContractor(rebate, optimalDeliveryDue)
15
16 // upload the IDD and RDD
17 if IDD = RDD
18     rebate_permit = true
19 else
20     rebate_permit = false
21 updateRebatePermissionToBlockchain (rebate_permitted)
22
23 waitForInstallation () // this includes transportation and inspection
24
25 // refresh the distribution and cumulative difference of EDI
26 ADI = getFromContractor ()
27 dist = updateDistByBayesian (dist, cum_diff, EDI, ADI)
28 cum_diff = adi - edi + cum_diff
29 uploadToBlockchain (dist, cum_diff)
30
31 // order finished
```

Table 2 The pseudocode for the rebate calculation and reliability updating

As shown in the pseudocode(Table 2), this study designed CalculationStepOne for the rebate price calculation and CalculationStepTwo for the distribution updating. The part of the code is shown in the following figures(Figure 16 and Figure 17).

```
// If this is not the 1st batch, there is a regular calculation step here.
diff_edi, _ = GetDaysBetween2Dates("2006-01-02 15:04:05", order.EDI, order_last.EDI)
cum_diff_edi = diff_edi + order_last.CumDiffEDI
kappa = order_last.Kappa + 1
alpha = order_last.Alpha + 0.5
standard_deviation_last := order_last.StandardDeviation

mean_last_int, _ := GetDaysBetween2Dates("2006-01-02 15:04:05", order_last.Mean, BaselineDate)
mean_last_float64 := float64(mean_last_int)

dist := distuv.Normal{
    Mu:    mean_last_float64,
    Sigma: standard_deviation_last,
    Src:   nil,
}
DifferenceInDates := distuv.Normal(dist).Quantile(TheRate)
// We need to convert the float64 to int, rounded
DifferenceInDates_int := int(math.Round(DifferenceInDates))
unadj_optimal_delivery_date_string, _ = GetDateAddInt("2006-01-02 15:04:05", BaselineDate, DifferenceInDates_int)
unadj_optimal_delivery_date_int, _ := GetDaysBetween2Dates("2006-01-02 15:04:05", unadj_optimal_delivery_date_string, BaselineDate)
unadj_optimal_delivery_date_float := float64(unadj_optimal_delivery_date_int)
optimal_delivery_date, _ = GetDateAddInt("2006-01-02 15:04:05", unadj_optimal_delivery_date_string, cum_diff_edi)
probability_density_function = distuv.Normal(dist).Prob(unadj_optimal_delivery_date_float)
optimal_rebate = ContractorCostRate * TheRate / probability_density_function
```

Figure 16 Part of the function CalculationStepOne

```
if order.BatchNo == "1" {
    // If this is the 1st batch, we treat the calculation specially
    adj_adi, _ = GetDateAddInt("2006-01-02 15:04:05", order.ADI, -1*cum_diff_edi)
    mean, _ = GetMeanToDate(order_last.Kappa, order.EDI, adj_adi, kappa, BaselineDate)
    beta, _ = UpdateBetaToFloat64(order_last.Kappa, adj_adi, order.EDI, order_last.Beta, BaselineDate)
    variance = beta / alpha
    standard_deviation = math.Sqrt(variance)
} else {
    // If this is not the 1st batch, there is a regular calculation step here.
    adj_adi, _ = GetDateAddInt("2006-01-02 15:04:05", order.ADI, -1*cum_diff_edi)
    mean, _ = GetMeanToDate(order_last.Kappa, order_last.Mean, adj_adi, kappa, BaselineDate)
    beta, _ = UpdateBetaToFloat64(order_last.Kappa, adj_adi, order_last.Mean, order_last.Beta, BaselineDate)
    variance = beta / alpha
    standard_deviation = math.Sqrt(variance)
}
```

Figure 17 Part of the function CalculationStepTwo

The calculationStepOne will be called right after the contractor submits the EDI in the fabric package to calculate the optimal date and rebate price. The CalculationStepTwo will be called following the contractor submitting the ADI to update the reliability distribution.

model package

The model package is the most intuitive named package that contains various pre-defined aggregate data models. The content of the model package is shown in the following figure(Figure 18).

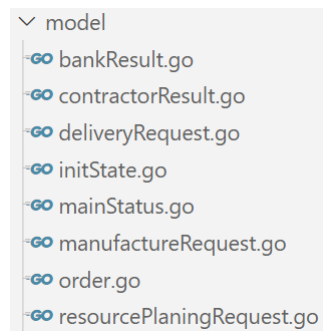


Figure 18 Different structs in the model package

In this package, the most important thing is the struct named order. This struct will be uploaded to the blockchain and used as a basis for information sharing and interaction. The order struct looks like the following figure.

```
// Basic Information
BatchNo string `json:"batch_no"`

// number 1-8 to represent status
OrderStatus string `json:"order_status"`

// Input of rebate
EDI string `json:"edi"`
ADI string `json:"adi"`

// Contractor
InitialDeliveryDue string `json:"estimate_delivery_due"`
RevisedDeliveryDue string `json:"revised_delivery_due"`
RebatePermission bool `json:"rebate_permission"`

// Inspetor(Pre)
DateOfPreInspection string `json:"date_of_pre_inspection"`
ResultOfPreInspection string `json:"result_of_pre_inspection"``
```

Figure 19 Part of the order struct

The other structs in the model package are the subset of this order struct. They were designed to give limited information to the relevant participant.

fabric package

The fabric package is the package that is primarily used to enable our interaction with the distributed ledger. There are 12 functions lie in the fabric package.

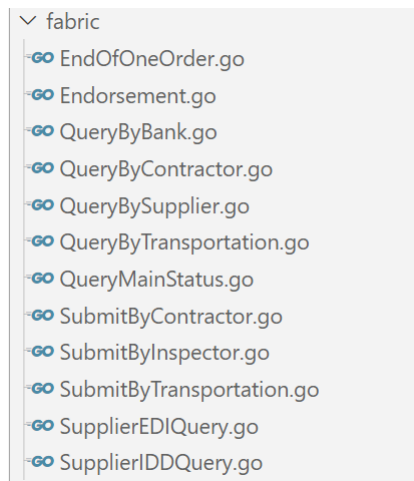


Figure 20 Functions in the fabric package

According to the data structure in section 4.4.2, the functions related to submitting the information are ContractorSubmit, InspectorSubmit, and TransportationSubmit.

The realization of one function will be shown here, and the rest are similar in idea.

ContractorSubmit

The contractor interacts with the blockchain platform most frequently. Before entering its function, the identity will be examined by the GetMSPID function.

At the same time, the information from the last order and the current order will be fetched from the blockchain and ready for further processing.

```

if mspid == "ContractorMSP" {
    /*
     1. Input EDI
     2. Input Initial Delivery date
     3. Input Revisied Delivery date
     4. Input Date of Arrival
     5. Input ADI
    */
    switch operator { ...
    }
} else {
    return "invoke failure", fmt.Errorf("you are not contractor")
}

```

Figure 21 Access control

In the **ContractorSubmit** function, there are 5 different cases, earliest possible date of installation(EDI), initial date of delivery (IDD), revised date of delivery (RDD), date of arrival(DOA), and actual date of installation(ADI). An operator is set in the function input arguments to navigate the function to the correct case.

Inside each case, specific logic is written, and several functions in the utils package are called. For example, in the case of EDI(Figure 22), the CalculationStepOne function is called to do the calculation for the optimal delivery date and rebate price right after the input.

```
case "EDI":
    order.EDI = args

    var BaselineDate string
    // if it is the 1st batch the the baselinedate = EDI
    // else call the GetBaselineDate function
    if order.BatchNo == "1" {
        BaselineDate = args
    } else {
        BaselineDate, _ = utils.GetBaselineDate(ctx)
    }

    *order = utils.CalcationStepOne(*order, *order_last, BaselineDate)
    order.OrderStatus = "1"
```

Figure 22 Case of EDI

After completing all operations, the processed order will be uploaded to the blockchain for information sharing. This coding part is shown in the following figure(Figure 23).

```

// Convert "order" to JSON (Encoding)
orderNewsAsBytes, _ := json.Marshal(order)
// Check idiomatic Error
err := ctx.GetStub().PutState(order.BatchNo, orderNewsAsBytes)
if err != nil {
|   return "putState failure", err
}
return "Successfully upload the information into the blockchain", nil

```

Figure 23 Uploading the data on the blockchain

It is important to note that in the blockchain network of fabric protocol, each submission of information needs to be endorsed by the relevant parties to guarantee its authenticity, so this endorsement policy also needs to be designed.

Order Status	Shared Information	Endorser
1	EDI	Contractor
2	IDD	Contractor
3	RDD	Contractor
4	Pre-inspection date and result	Supplier, Inspector
5	Date of loading	Supplier, Transportation
6	Date of Arrival	Contractor, Transportation
7	Post-inspection date and result	Contractor, Inspector, and Transportation
8	ADI	Contractor

Table 3 The endorsement policy design

Chapter 5 Prototype Validation

5.1 Validation Environment Setup

5.1.1 Validation Environment

This section will validate the functionality of the designed prototype. The validation process will use Hyperledger Fabric version 2.2.7 as the blockchain protocol. At the same time, Fabric CA version 1.5.3 is used to create the certificate authority files for each participant within the Hypothetical case study. This study uses docker version 4.15.0 to create containers for the different fabric peers and clients to conduct the network simulation.

5.1.2 Network Setup

First, the cryptogen tool is needed to create the network configuration. In the prototype validation process, this research used the most straightforward underpinning network architecture to simulate the scenario. Therefore, a network with five peers and one orderer has been configured in the crypto-config file. Part of the configuration is shown below(Figure 24). After that, the identity files were generated in the crypto-config folder. Those files will be used as the certificate of identity, which uses asymmetric cryptography(Ma et al. 2019).

```

1 OrdererOrgs:
2   - Name: Rebate
3     Domain: rebate.com
4     EnableNodeOUs: true
5     Specs:
6       - Hostname: orderer
7 PeerOrgs:
8   - Name: Supplier
9     Domain: supplier.rebate.com
10    EnableNodeOUs: true
11    Template:
12      Count: 1
13    Users:
14      Count: 1
15   - Name: Transportation
16     Domain: transportation.rebate.com
17     EnableNodeOUs: true
18     Template:
19       Count: 1
20     Users:
21       Count: 1

```

```

  crypto-config
  ordererOrganizations
  peerOrganizations
    bank.rebate.com
    contractor.rebate.com
      ca
      msp
      peers
      tlsca
      users
    inspector.rebate.com
    supplier.rebate.com
    transportation.rebate.com

```

Figure 24 (left) configuration of the crypto-config

(right) identity files used for the blockchain system

Second, the configtxgen tool is needed to generate the configuration transaction in our prototype. This step will set up the configuration of the channel configuration. Some of the settings are shown in the following figure.

```

RebateMainChannel:
  Consortium: SampleConsortium
  <<: *ChannelDefaults
  Application:
    <<: *ApplicationDefaults
    Organizations:
      - *Supplier
      - *Transportation
      - *Contractor
      - *Inspector
      - *Bank
    Capabilities:
      <<: *ApplicationCapabilities

```

Figure 25 The configuration file of the channel

All configuration for the channel is generated by the configtx file, and the configtx file needs the identity files generated from the first step. After setting up the configtx file, the genesis and the channel configuration block were generated.

Third, with the help of the configtxgen tool, we can also generate the configuration blocks for anchor peer setup. Anchor peers are those responsible for interactions with peers from other organizations. All those blocks from step 2 and step 3 are shown in the following figure(Figure 26).

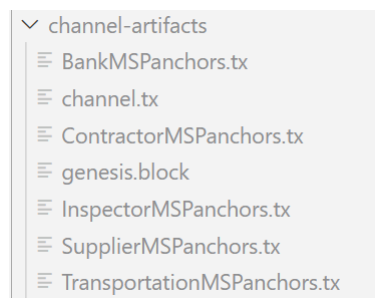


Figure 26 Blocks from step2 and step3

Forth, with the help of the docker, we can build a test network locally to validate the prototype. Docker enables the possibility to simulate the network with a configuration file called docker-compose written in yaml format. Some of the configurations are shown in the following figure(Figure 27).

```

358 bank_cli: #changed
359 container_name: bank_cli #changed
360 image: hyperledger/fabric-tools:2.2.7
361 tty: true
362 stdin_open: true
363 > environment: ...
375 working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
376 command: /bin/bash
377 volumes:
378 - /var/run:/host/var/run/
379 - ../chaincode/go:/opt/gopath/src/github.com/hyperledger/fabric-cluster/chaincode/go
380 - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
381 - ./channel-artifacts:/opt/gopath/src/github.com/hyperledger/fabric/peer/channel-artifacts
382 networks:
383 - test

```

Figure 27 Test network configuration

After settling the configurations, this study used the docker to build the test network.

As shown in the figure(Figure 28), there are 11 containers in the test network.

```

[+] Running 18/18
  :: Network UW_LCR_01_test                Created
  :: Volume "network_peer0.supplier.rebate.com"    Created
  :: Volume "network_orderer.rebate.com"          Created
  :: Volume "network_peer0.contractor.rebate.com"  Created
  :: Volume "network_peer0.transportation.rebate.com" Created
  :: Volume "network_peer0.bank.rebate.com"        Created
  :: Volume "network_peer0.inspector.rebate.com"   Created
  :: Container contractor_cli                 Started
  :: Container peer0.inspector.rebate.com        Started
  :: Container inspector_cli                  Started
  :: Container peer0.transportation.rebate.com    Started
  :: Container orderer.rebate.com              Started
  :: Container supplier_cli                   Started
  :: Container peer0.contractor.rebate.com        Started
  :: Container peer0.supplier.rebate.com         Started
  :: Container peer0.bank.rebate.com            Started
  :: Container transportation_cli              Started
  :: Container bank_cli                       Started

```

Figure 28 Building the networks

While the test network has been established, those configuration blocks created in the previous steps were used to build the channel and update the anchor peer information.

```
Step6 : Join the Channel
2023-01-03 01:15:20.670 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:20.711 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
2023-01-03 01:15:20.903 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:20.943 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
2023-01-03 01:15:21.128 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:21.166 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
2023-01-03 01:15:21.390 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:21.431 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
2023-01-03 01:15:21.627 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:21.663 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
Step7: Update the Anchor Peer
2023-01-03 01:15:21.904 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:21.924 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
2023-01-03 01:15:22.130 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:22.146 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
2023-01-03 01:15:22.452 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:22.473 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
2023-01-03 01:15:22.702 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:22.721 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
2023-01-03 01:15:22.986 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2023-01-03 01:15:23.003 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
```

Figure 29 Join the channel and update the anchor peer

With all those steps above, a test network environment was set up and ready for a hypothesis case study.

5.2 Hypothetical Case Study

Compared with the traditional software testing process, the study will simulate a closer implementation of the actual situation. At the same time, the Hyperledger explorer will be used in this section to visualize the process. In this section, the study uses a hypothetical case study to test our prototype to determine if it meets the expectation. The used experimental data is from a set of actual construction data about a curtain wall project collected in 2015. In this thesis, three rounds of the order lifecycle will be simulated.

5.2.1 Initial steps

The first step in the test environment would be for all participants “read” and “agree” with the “terms” written in the smart contract. After they agree on the streamlined workflow, they will “sign” the smart contract with their CA files and commit it. This step is shown in the following figure(Figure 30).

```
Check the chaincode commit is ready
{
    "approvals": {
        "BankMSP": true,
        "ContractorMSP": true,
        "InspectorMSP": true,
        "SupplierMSP": true,
        "TransportationMSP": true
    }
}
Commit the chaincode
```

Figure 30 Approval from each participant

With the Hyperledger explorer, we can see that the network is correctly settled, and the chaincode(an alternative name for the smart contract) is successfully installed on each peer(Figure 31 and Figure 32).

Peer Name	Request Url	Peer Type	MSPID
peer0.contractor.rebate.com...	peer0.contractor.rebate.com...	PEER	ContractorMSP
peer0.inspector.rebate.com:...	peer0.inspector.rebate.com:...	PEER	InspectorMSP
peer0.transportation.rebate....	peer0.transportation.rebate....	PEER	TransportationMSP
peer0.bank.rebate.com:10500	peer0.bank.rebate.com:10500	PEER	BankMSP
peer0.supplier.rebate.com:1...	peer0.supplier.rebate.com:1...	PEER	SupplierMSP
orderer.rebate.com:9050	orderer.rebate.com:9050	ORDERER	RebateMSP

Figure 31 The participants of the test network

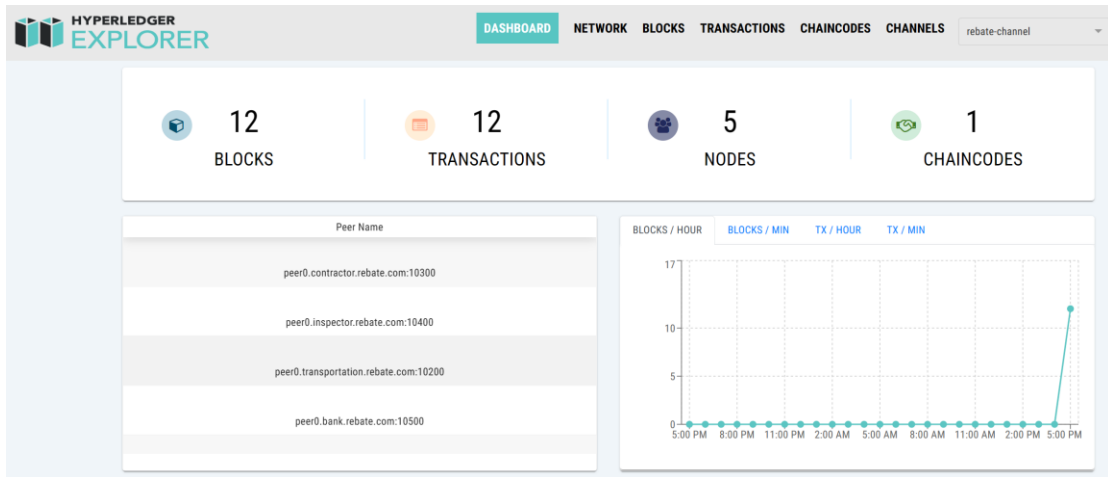


Figure 32 The dashboard of Hyperledger Explorer

After the successful installation of the smart contract, the supplier will perform the very first action on the network, which is the “initLedger” function. This function will write the pre-defined reliability information before the order of the first batch.

```

o seanzhao@DESKTOP-SEAN:~/LM_LCR_01$ docker exec -it supplier_cli bash
bash-5.1# peer chaincode invoke -o orderer.rebate.com:9050 --isInit --ordererTLSHostnameOverride orderer.rebate.com --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/rebate.com/orderers/orderer.rebate.com/msp/tlscacerts/tlsca.rebate.com-cert.pem -C rebate-channel -n rebate --peerAddresses peer0.supplier.rebate.com:10100 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/supplier.rebate.com/peers/peer0.supplier.rebate.com/tls/ca.crt --peerAddresses peer0.transportation.rebate.com:10200 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/transportation.rebate.com/peers/peer0.transportation.rebate.com/tls/ca.crt --peerAddresses peer0.contractor.rebate.com:10300 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/contractor.rebate.com/peers/peer0.contractor.rebate.com/tls/ca.crt --peerAddresses peer0.inspector.rebate.com:10400 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/inspector.rebate.com/peers/peer0.inspector.rebate.com/tls/ca.crt --peerAddresses peer0.bank.rebate.com:10500 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/bank.rebate.com/peers/peer0.bank.rebate.com/tls/ca.crt -c '{"function": "initLedger", "Args": []}'
2023-01-03 01:50:15.302 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
bash-5.1# █

```

Figure 33 InitLedger function process

The screenshot shows a 'Transaction Details' window with the following information:

- Transaction ID:** c5b28d6694d5d6f001c54f1677070ede37047b267e467316ca44411d70b20ebb
- Validation Code:** VALID
- Payload Proposal Hash:** 1da26dce6c637de4a689f77fd6cdb6e756877a0bc33486ccdbb4838cb72fe12
- Creator MSP:** SupplierMSP
- Endorser:** {"SupplierMSP","TransportationMSP","BankMSP","ContractorMSP","InspectorMSP"}
- Chaincode Name:** rebate
- Type:** ENDORSER_TRANSACTION
- Time:** 2023-01-03T01:50:15.280Z
- Direct Link:** <http://127.0.0.1:8080/?tab=transactions&transId=c5b28d6694d5d6f001c54f1677070ede37047b267e467316ca44411d70b20ebb>
- Reads:**
 - root: [] 2 items
 - 0: {} 2 keys
 - 1: {} 2 keys
- Writes:**
 - root: [] 2 items
 - 0: {} 2 keys
 - 1: {} 2 keys

Figure 34 The Prior Information in the 0th order

With the initialization of the smart contract, the business simulation process is ready to be conducted using the data derived from real-world cases.

5.2.2 Order simulation

Three rounds of order simulations were conducted in this study, and this section will examine the functionality of the prototype using the third round as a context.

Opportunistic attempt

In the Hypothetical case, the third order started with the contractor sharing its earliest possible date of installation(EDI). The smart contract is coded with access control, which means the wrong client cannot upload the information. For example, if the supplier wants to fake an EDI for its benefit, the smart contract will not allow that operation(Figure 35). This is a critical feature for realizing tamper-proof in the blockchain system.

```
○ seanzhao@DESKTOP-SEAN:~/UN_LCR_01$ docker exec -it supplier_c11 bash
bash-5.1# peer chaincode invoke -o orderer.rebate.com:9050 --ordererTLSHostnameOverride orderer.rebate.com --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/rebate.com/orderers/orderer.rebate.com/msp/tlscacerts/tlsca.rebate.com-cert.pem -C rebate-channel -n rebate --peerAddresses peer0.contractor.rebate.com:10300 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/contractor.rebate.com/peers/peer0.contractor.rebate.com/tls/ca.crt -c '{"function": "ContractorSubmit", "Args": ["3", "EDI", "2015-10-05 00:00:00"]}'
Error: endorsement failure during invoke. response: status:500 message:"you are not contractor"
bash-5.1#
bash-5.1#
```

Figure 35 Failure from the client side

At the same time, the smart contract also designed a specific endorsement policy for each information-sharing operation. If the collected endorsements do not correspond with the smart contract, an endorsement policy failure will occur, and the untrustworthy information will not be uploaded to the blockchain. This liability-based confirmation mechanism example is shown in the following figure(Figure 36). The EDI should be

provided by the contractor and endorsed by the contractor. The endorsement policy failure occurred when the supplier endorsed the information, not the contractor.

The screenshot shows a 'Transaction Details' window with the following information:

- Transaction ID: ad3e9161f9bb3f4c04ec96a812ee54c81b30509090eea280adbd02e6aeb712c3
- Validation Code: ENDORSEMENT_POLICY_FAILURE (highlighted with a red box)
- Payload Proposal Hash: 942e23201019c4c2c1d63f478469fea3340b162dbf136eb5d5806416736fdec1
- Creator MSP: ContractorMSP
- Endorser: {"SupplierMSP"} (highlighted with a red box)
- Chaincode Name: rebate
- Type: ENDORSER_TRANSACTION
- Time: 2023-01-03T03:08:04.650Z
- Direct Link: <http://127.0.0.1:8080/?tab=transactions&transId=ad3e9161f9bb3f4c04ec96a812ee54c81b30509090eea280adbd02e6aeb712c3>
- Reads:
 - root: [] 2 items
 - 0: {} 2 keys
 - 1: {} 2 keys
- Writes:
 - root: [] 2 items
 - 0: {} 2 keys
 - 1: {} 2 keys

Figure 36 Endorsement policy failure example – providing EDI

The failure of attempted opportunistic behavior demonstrates the potential of designed smart contracts to guarantee the source of information and the participants in charge.

Not only providing the information but also querying the information has access control.

If the inspector wants to query the information stored for the contractor, the returned result will not contain anything valuable(Figure 37).

```
o seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it inspector_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":["QueryByContractor","2"]}'
{"batch_no":"","optimal_delivery_date":"","rebate":0}
bash-5.1# █
```

Figure 37 The returned value will be empty

Regulated procedure

The regular third order starts with the information sharing from the contractor. The contractor shares its EDI using the contractor client with its own endorsement. After the validation by the smart contract, this information was successfully uploaded to the blockchain.

```
seanzhao@DESKTOP-SEAN:~/UM_LCR_01$ docker exec -it contractor_cli bash
bash-5.1# peer chaincode invoke -o orderer.rebate.com:9050 --ordererTLSHostnameOverride orderer.rebate.com --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/rebate.com/orderers/orderer.rebate.com/msp/tlscacerts/tlsca.rebate.com-cert.pem -C rebate-channel -n rebate --peerAddresses peer0.contractor.rebate.com:10300 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/contractor.rebate.com/peers/peer0.contractor.rebate.com/tls/ca.crt -c '{"function": "ContractorSubmit", "Args": [{"3", "EDI", "2015-10-05 00:00:00"}]}'
2023-01-03 03:13:02.222 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"Successfully upload the information into the blockchain"
bash-5.1#
```

The screenshot shows a 'Transaction Details' window with the following data:

Transaction ID:	14ac2485eea49946c68487598c883bf926c5c82288ea4c89651e0fe5032c52a3
Validation Code:	VALID
Payload Proposal Hash:	18c77360890c100babf039650ded52fd7ed0ef842f81c3cd79a5604099f3dd3b
Creator MSP:	ContractorMSP
Endorser:	{"ContractorMSP"}
Chaincode Name:	rebate
Type:	ENDORSER_TRANSACTION
Time:	2023-01-03T03:13:02.214Z
Direct Link:	http://127.0.0.1:8080/?tab=transactions&transId=14ac2485eea49946c68487598c883bf926c5c82288ea4c89651e0fe5032c52a3

Reads:

- root: [] 2 items
 - 0: {} 2 keys
 - 1: {} 2 keys

Writes:

- root: [] 2 items
 - 0: {} 2 keys
 - 1: {} 2 keys

Figure 38 Contractor uploads the EDI

At this time, the rebate price and optimal delivery date are automatically calculated and ready for being queried by the contractor. Followed by the uploading of EDI, the

contractor will query the information(Figure 39) and make a rational decision on the following initial and revised delivery dates.

```

seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it contractor_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":["QueryByContractor","3"]}'
{"batch_no":"3","optimal_delivery_date":"2015-10-07 00:00:00","rebate":88.07393994680024}
bash-5.1# █

```

Figure 39 Contractor query the optimal delivery date and rebate for rational decision

At the same time, the EDI should also be the supplier's reference for its production resource planning. Therefore, the supplier will log in to the system and query the EDI(Figure 40).

```

seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it supplier_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":["QueryEDIBySupplier","3"]}'
{"batch_no":"3","edi":"2015-10-05 00:00:00"}
bash-5.1# █

```

Figure 40 Supplier query the EDI for resource planning

After the contractor makes its rational decision, the initial delivery date(IDD) is uploaded to the shared ledger(Figure 41).

```

seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it contractor_cli bash
bash-5.1# peer chaincode invoke -o orderer.rebate.com:9050 --ordererTLSHostnameOverride orderer.rebate.com --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/rebate.com/orderers/orderer.rebate.com/msp/tlscacerts/tlsca.rebate.com-cert.pem -C rebate-channel -n rebate --peerAddresses peer0.contractor.rebate.com:10300 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/contractor.rebate.com/peers/peer0.contractor.rebate.com/tls/ca.crt -c '{"function":"ContractorSubmit","Args":["3","IDD","2015-10-07 00:00:00"]}'
2023-01-03 03:27:30.643 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"Successfully upload the information into the blockchain"
bash-5.1# █

```

Figure 41 Contractor uploads its IDD

The IDD is the reference for the supplier to start the production of the corresponding order. Therefore, the supplier would query this information for its production schedule(Figure 42).

```

o seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it supplier_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":["QueryIDDBySupplier","3"]}'
{"batch_no":"3","initial_delivery_due":"2015-10-07 00:00:00"}
bash-5.1# █

```

Figure 42 Supplier queries the IDD for production schedule

Approaching the installation date, the contractor will upload its revised delivery date as determined by the weekly construction plan(Figure 43).

```

o seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it contractor_cli bash
bash-5.1# peer chaincode invoke -o orderer.rebate.com:9050 --ordererTLSHostnameOverride orderer.rebate.com --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/rebate.com/orderers/orderer.rebate.com/msp/tlscacerts/tlsca.rebate.com-cert.pem -C rebate-channel -n rebate --peerAddresses peer0.contractor.rebate.com:10300 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/contractor.rebate.com/peers/peer0.contractor.rebate.com/tls/ca.crt -c '{"function":"ContractorSubmit","Args":["3", "RDD", "2015-10-08 00:00:00"]}'
2023-01-03 03:32:20.492 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"Successfully upload the information into the blockchain"
bash-5.1# █

```

Figure 43 Contractor uploads the revised delivery date

As in the simulated third order, the revised delivery date(10-08) is inconsistent with the initial delivery date(10-07), which is considered unreliable by the smart contract design. Therefore, the rebate permission will not be given to the contractor. The bank will later know this information for the rebate release transaction.

The transportation will be informed of the revised delivery date for the delivery schedule arrangement(Figure 44).

```

o seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it transportation_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":["QueryByTransportation","3"]}'
{"batch_no":"3","revised_delivery_due":"2015-10-08 00:00:00"}
bash-5.1# █

```

Figure 44 Transportation queries the RDD

During the following workflow, the order will go through the logistics process of pre-inspection, loading, arrival, and post-inspection. Due to the enforceability of the smart contract, opportunistic behavior will also be prohibited during this period. For example, Figure 45 shows the transportation tried to upload the date of loading before the pre-inspection was finished, but the smart contract will not permit updating the information.

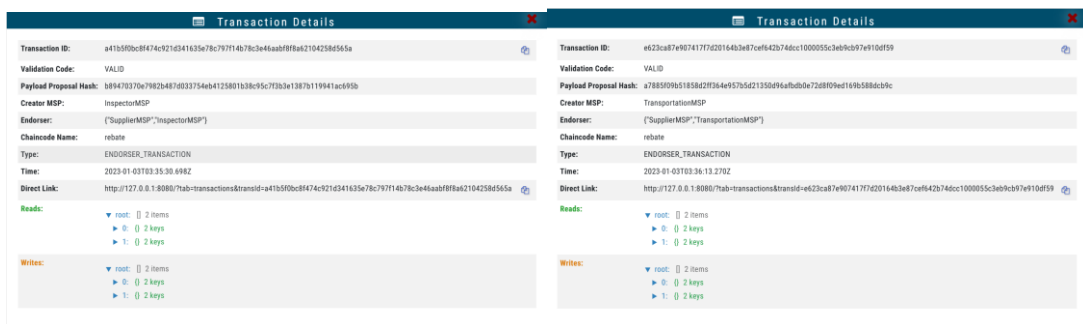
```

o seanzhao@DESKTOP-SEAN:~/UM_LCR_01$ docker exec -it transportation_cli bash
bash-5.1# peer chaincode invoke -o orderer.rebate.com:9050 --ordererTLSHostnameOverride orderer.rebate.com --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/rebate.com/orderers/orderer.rebate.com/msp/tlscacerts/tlsca.rebate.com-cert.pem -C rebate-channel -n rebate --peerAddresses peer0.supplier.rebate.com:10100 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/supplier.rebate.com/peers/peer0.supplier.rebate.com/tls/ca.crt --peerAddresses peer0.transportation.rebate.com:10200 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/transportation.rebate.com/peers/peer0.transportation.rebate.com/tls/ca.crt -c '{"function": "TransportationSubmit", "Args": ["3", "DOI", "2015-10-07 00:00:00"]}'
Error: endorsement failure during invoke. response: status:500 message:"the pre inspection is not finished"
bash-5.1#

```

Figure 45 Transportation is trying to skip the pre-inspection process.

Information that is essential for tracking the status of each order delivery from the perspective of the entire supply chain, but irrelevant to the rebate negotiation, is imported by each stakeholder (Figure 46). These uploaded pieces of information require endorsement from the relevant stakeholders and must be strictly processed in sequence.



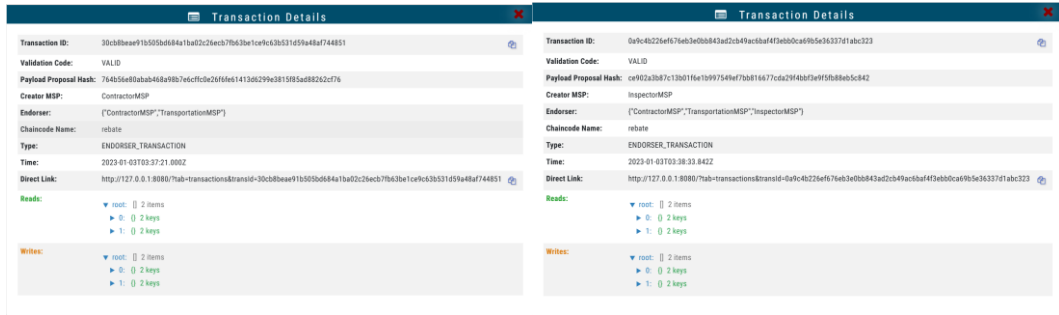


Figure 46 The information about pre-inspection, loading, arrival, and post-inspection

After the logistic procedure is finished, the contractor will share the actual date of installation(ADI).

```

seanzhao@DESKTOP-SEAN:~/UM_LCR_01$ docker exec -it contractor_cli bash
bash-5.1# peer chaincode invoke -o orderer.rebate.com:9050 --ordererTLSHostnameOverride orderer.rebate.com --tls true --cafile /opt/gopath/
src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/rebate.com/orderers/orderer.rebate.com/msp/tlscacerts/tlscra.rebate.com-c
ert.pem -C rebate-channel -n rebate --peerAddresses peer0.contractor.rebate.com:10300 --tlsRootCertFiles /opt/gopath/src/github.com/hyperle
dger/fabric/peer/crypto/peerOrganizations/contractor.rebate.com/peers/peer0.contractor.rebate.com/tls/ca.crt -c '{"function": "ContractorSub
mit", "Args": [{"3", "ADI", "2015-10-10 00:00:00"}]}'
2023-01-03 03:39:36.689 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"Succ
essfully upload the information into the blockchain"
bash-5.1#

```

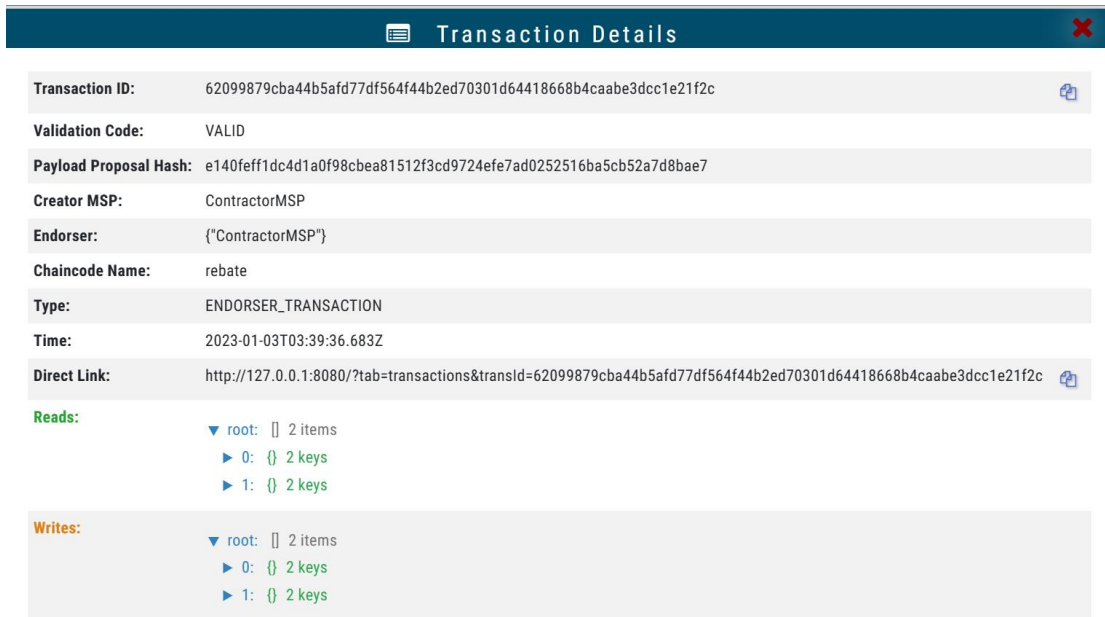


Figure 47 The ADI uploaded by the contractor for the reliability calculation

After the installation, the bank can query the information for the rebate price and permission, giving the bank the reference for the rebate transaction.

```
○ seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it bank_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":
["QueryByBank", "3"]}'
{"batch_no": "3", "optimal_rebate": 88.07393994211064, "rebate_permission": false}
bash-5.1# █
```

Figure 48 The rebate information queried by the bank

The QueryMainStatus function shows meaningful information about the third order in the client(Figure 49).

```
○ seanzhao@DESKTOP-SEAN:~/UW_LCR_01$ docker exec -it supplier_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":
["QueryMainStatus", "3"]}'
{"batch_no": "3", "order_status": "8", "edi": "2015-10-05 00:00:00", "adi": "2015-10-10 00:00:00", "optimal_delivery_date": "2015-10-07 00:00:00", "r
ebate": 88.07393994680024, "estimate_delivery_due": "2015-10-07 00:00:00", "revised_delivery_due": "2015-10-08 00:00:00", "rebate_permission": fal
se, "result_of_pre_inspection": "Passed the Pre-Inspection", "result_of_post_inspection": "Passed the Post-Inspection"}
bash-5.1# █
```

Figure 49 Main information of the third order

The complete information, including those intermediate variables, is stored in the blockchain but will not be available to a normal user, only to the administrator. The complete information of the third order is shown in Figure 50.

```
Writes:
  ▼ root: [] 2 items
    ► 0: {} 2 keys
    ▼ 1: {} 2 keys
      chaincode: "rebate"
      ▼ set: [] 1 item
        ▼ 0: {} 3 keys
          key: "3"
          is_delete: false
          value: "{\"batch_no\":\"3\",\"order_status\":\"8\",\"edi\":\"2015-10-05 00:00:00\",\"adi\":\"2015-10-10 00:00:00\",\"diff_edi\":0,\"cum_diff_edi\":11,\"adj_adi\":\"2015-09-29 00:00:00\",\"kappa\":3,\"mean\":\"\",\"alpha\":2.5,\"beta\":0,\"variance\":0,\"standard_deviation\":0,\"un_adj_optimal_delivery_date\":\"\",\"probability_density_function\":0.2523132522154507,\"optimal_delivery_date\":\"\",\"rebate\":88.07393994211064,\"estimate_delivery_due\":\"2015-10-07 00:00:00\",\"revised_delivery_due\":\"2015-10-08 00:00:00\",\"rebate_permission\":false,\"date_of_pre_inspection\":\"2015-10-07 00:00:00\",\"result_of_pre_inspection\":\"Passed the Pre-Inspection\",\"date_of_loading\":\"2015-10-07 00:00:00\",\"date_of_post_inspection\":\"2015-10-09 00:00:00\",\"result_of_post_inspection\":\"Passed the Post-Inspection\",\"date_of_arrival\":\"2015-10-09 00:00:00\"}"
```

Figure 50 Complete information of the third order

Rebate information comparison

Although the rebate mechanism is involved in the first three orders, the rebate results are affected by the contractors' decisions and other factors. The following results show those three orders' rebate prices and permission.

- seanzhao@DESKTOP-SEAN:~/UW_LCR_01\$ docker exec -it bank_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":["QueryByBank","1"]}'
{ "batch_no": "1", "optimal_rebate": 0, "rebate_permission": true }
bash-5.1# █
- seanzhao@DESKTOP-SEAN:~/UW_LCR_01\$ docker exec -it bank_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":["QueryByBank","2"]}'
{ "batch_no": "2", "optimal_rebate": 45.48118701984393, "rebate_permission": true }
bash-5.1# █
- seanzhao@DESKTOP-SEAN:~/UW_LCR_01\$ docker exec -it bank_cli bash
bash-5.1# peer chaincode query -C rebate-channel -n rebate -c '{"Args":["QueryByBank","3"]}'
{ "batch_no": "3", "optimal_rebate": 88.07393994680024, "rebate_permission": false }
bash-5.1# █

Figure 51 Comparison of 3 different scenarios of the rebate mechanism

Since the rebate price is calculated from the information from the order before the current one, there will not be a rebate price in the first order. The IDD is the same as the RDD in the second order, so the rebate permission is true. In the third order, the contractor did not keep its promise on the delivery date, so the rebate permission is false. These three cases demonstrate different situations of kickbacks and show that our prototype has reached the designed calculation and logic judgment capability.

Demonstration of blockchain features

Through a three-order hypothetical case study, this study verified that the ledger stored on each peer node remained consistent upon examination through the Hyperledger Explorer. The consistent ledger height corroborates the characteristic of blockchain technology that promotes transparency by sharing reliable information.

Peer Name	Request Url	Peer Type	MSPID	Ledger Height		
				High	Low	Unsigned
peer0.contractor.rebate.co...	peer0.contractor.rebate.co...	PEER	ContractorMSP	0	40	true
peer0.inspector.rebate.com:...	peer0.inspector.rebate.com:...	PEER	InspectorMSP	0	40	true
peer0.transportation.rebate....	peer0.transportation.rebate....	PEER	TransportationMSP	0	40	true
peer0.bank.rebate.com:105...	peer0.bank.rebate.com:105...	PEER	BankMSP	0	40	true
peer0.supplier.rebate.com:1...	peer0.supplier.rebate.com:1...	PEER	SupplierMSP	0	40	true
orderer.rebate.com:9050	orderer.rebate.com:9050	ORDERER	RebateMSP	-	-	-

Previous Page 1 of 1 10 rows Next

Figure 52 The consistent ledger height

At the same time, the blockchain serves as a secure and immutable record of the information manipulation on the ledger, linked through hash cryptography to prevent tampering. The following figure shows that each block is linked to its previous block. If someone wants to change the record in the middle, that requires changing all following blocks, which is impossible with the current technology.

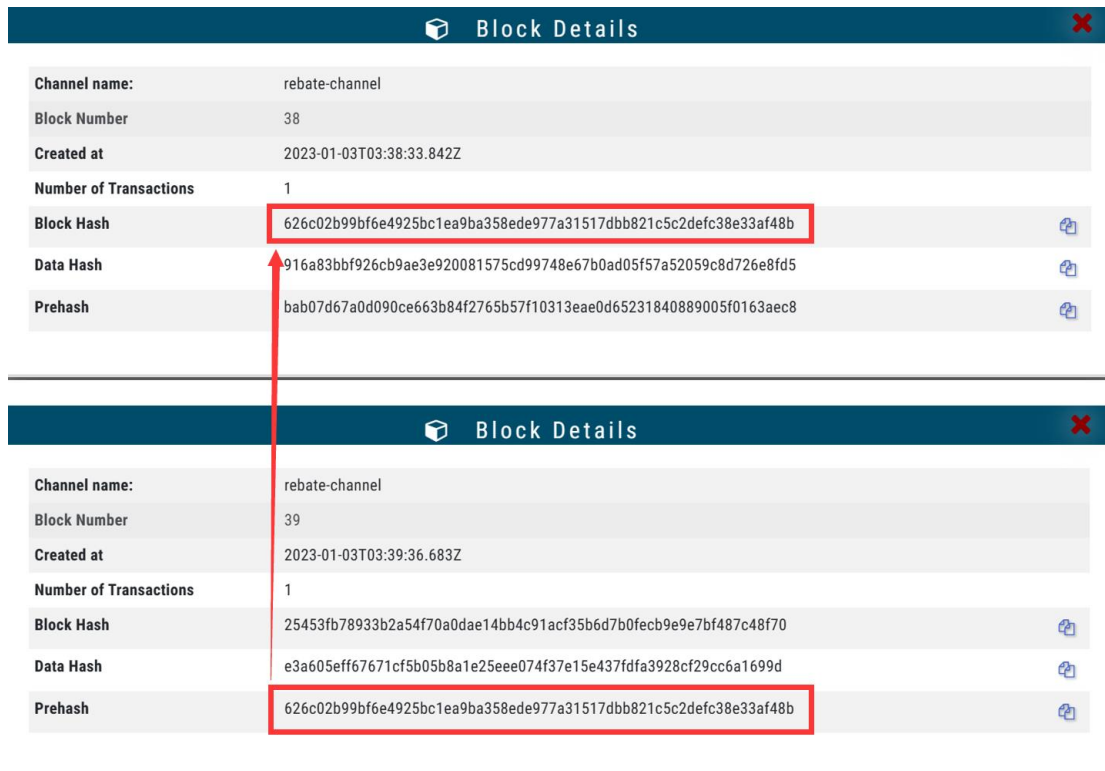


Figure 53 Block relationship

Despite the fact that multiple peers maintain a logically-single database, a database is not thoroughly controlled by a single entity, as each participant independently contributes to the ledger's updates without the need for a central institution. This characteristic gives blockchain-based prototypes a decentralized nature(Figure 54).



Figure 54 The transaction proposed by different organizations

5.2.3 Comparism with the traditional fixed rebate

Although the rebate mechanism can be implemented with traditional fixed rate, a comparism table is made to analyze the benefit of the blockchain-based system.

Aspect	Traditional Fixed Rebate System	Blockchain-Based Rebate System
Communication	Rebate information is usually communicated via email or other non-real-time channels	Updates are communicated in real-time
Calculation	Rebates are calculated using historical data with traditional methods, such as spreadsheet programs	Rebates are automatically calculated and updated in real-time using a decentralized ledger
Transparency	Lack of transparency because it is difficult for contractors to verify if the rebates they are receiving are accurate	Offers complete transparency as buyers can track the entire process and see the rebates they are receiving in real-time
Security	More prone to errors and fraud because it relies on manual processes	Highly secure because it is decentralized and cannot be tampered with
Efficiency	Generally slower and less efficient because it relies on manual processes	Automated and can update rebates in real-time, making it much more efficient

Table 4 Comparism analysis between traditional and Block-chain based system

In the context of off-site construction supply chain coordination, a blockchain-based rebate system can offer greater efficiency, transparency, and security than a traditional fixed rate rebate system. By automating the calculation and updating of rebates using a decentralized ledger, stakeholders can have greater control and visibility into the rebate process, and the system is less prone to errors and fraud. The table above provides a detailed comparison of these two approaches, highlighting the benefits of a blockchain-based rebate system in the context of off-site construction.

Chapter 6 Research Discussion

This research investigated a potential solution to align the interests and decisions of stakeholders in an off-site construction project. After comparing various methods, the study used a contractual approach and implemented that approach into a smart contract on a blockchain platform. Chapter 5 has shown the feasibility of this approach and tested its functionality. However, there are still some limitations in the scope of this research needed to be discussed.

6.1 Research Limitation and future research

First, this research adopted the rebate mechanism developed by Kim and Rhee (Kim and Rhee 2021), which assumes the contractor is unreliable. From the observation of the construction industry, the supplier's production schedule may also be unreliable under certain circumstances. The variability from the supplier side can be caused by different aspects, such as job shop scheduling and prioritizing the order from someone familiar. The rebate mechanism is based on the implicit assumption that the production from the supplier is unlimited. At the same time, the current unit cost from both supplier and contractor is set to a fixed rate, which can be improved in the future. The next step in developing a more advanced mechanism should be focused on introducing the uncertainty from the supplier's side and other characteristics.

Second, the current blockchain system contains a smart contract to enforce the business flow. However, from the hypothetical case study, the limitation of the smart contract is

brought to light. The blockchain platform, accompanied by smart contracts, serves as a passive system that cannot actively access outside information. While blockchain has proven to be effective in ensuring that the data entered is tamper-proof(Wu et al. 2022a), further research is needed to ensure that the data captured by the platform is reliable. Such questions are investigated in the computer science domain to improve incoming information's reliability and the automation level of the whole system. In the construction discipline, Lu investigated blockchain oracles to give a possible solution in 2021(Lu et al. 2021). Future research on the organic integration of business smart contracts and oracles(connected to BIM) needs to be added to the existing body of knowledge.

6.2 Research Contribution

This research presents a blockchain-enabled supply chain coordination platform that, despite certain limitations, is feasible and applicable, making several contributions to the current body of knowledge in construction management.

First, the whole process of the construction prefabrication supply chain is standardized in the smart contract, and every step of an order is recorded on the blockchain, which greatly increases transparency. This transparency can help to increase trust and confidence in the system, as all participants can see that the information is being recorded accurately and cannot be altered without the consensus of the network.

Concomitantly, this trust will increase the accountability of participants for the information they provide.

Second, the rebate mechanism is based on the assumption of a sequential game following the Stackelberg leadership model (Von Stackelberg 1934). However, opportunistic behavior in the construction industry can undermine the expected subgame perfect Nash equilibrium (Lu et al. 2016). With its transparent, tamper-proof features, a blockchain-based platform can give this game theory-based rebate mechanism an ideal practicing playground. With the help of a smart contract, the calculation steps are automated and streamlined for both the rebate price and the reliability distribution, providing an efficient coordination system.

6.3 Future of construction supply chain management

In the future, supply chain coordination in construction will likely continue to evolve and become more efficient and effective through advanced technologies such as machine learning, the Internet of Things (IoT), and blockchain. These technologies will enable real-time tracking and monitoring of materials, equipment, and other resources, allowing for more accurate forecasting and better decision-making. In addition, using digital platforms and collaboration tools will facilitate better communication and coordination among all stakeholders in the supply chain, including contractors, suppliers, and logistics providers. This study believes that as prefabricated construction

becomes more popular in the construction industry, more advanced technologies will bring a brighter and more promising future to the construction supply chain.

Chapter 7 Conclusion

With prefabrication and modular construction becoming popular in the construction industry, supply chain coordination becomes a crucial factor in the success of projects. Many past studies have examined different approaches to building trust among stakeholders and improving project performance. However, in the construction industry, where the silo mindset is prevalent, it is not practical to persuade the stakeholders to change their subjective decision-making habits overnight.

Therefore, this study proposed a blockchain-based system based on the premise that stakeholders do not have to trust each other. This study attempted to transform the relationship-based trust between construction parties into technology-based trust in the blockchain. In the meantime, in order to be able to improve the unreliable delivery requests of contractors, this study adopts a rebate incentive proposed by Kim and Rhee as a contractual scheme. This approach takes advantage of the reliability of smart contracts on the blockchain to remove the influence of possible opportunistic human behavior. With the help of smart contracts, a blockchain prototype for prudent supply chain management is proposed. This prototype is more in line with the philosophy of lean construction to streamline the workflow.

In the hypothetical case study conducted, this study demonstrated that the proposed blockchain prototype could effectively update the valuable information related to the

participants in real-time. In addition, all participants involved only have access to their relevant information, ensuring data security.

This research provided a suggested workflow and blockchain prototype as a departure to more actively investigating the potential of integrating new technology in supply chain management. After verifying that blockchain technology can reliably serve as an infrastructure for information sharing, further research should be done in the future.

In the future, research on the technology integration can embark on enriching the proposed prototype, including but not limited to proposing a more user-friendly operation process that incorporates IoT technology to improve automation level further.

Another future research can be placed on the improvement of coordination mechanisms, where the current incentives approach derived from a rule-based model can be improved with emerging methods, such as machine learning, to estimate and predict the supply chain situation. Likewise, in the future, the actual benefits of solutions with new technology call for further empirical analysis. After developing a matured solution that can be put into production, the corresponding research should be carried out.

Bibliography

- Barnes-Schuster, D., Y. Bassok, and R. Anupindi. 2002. "Coordination and Flexibility in Supply Contracts with Options." *Manuf. Serv. Oper. Manag.*, 4 (3): 171–207. <https://doi.org/10.1287/msom.4.3.171.7754>.
- Bider, I., P. Johannesson, E. Perjons, and L. Johansson. 2012. "Design Science in Action: Developing a Framework for Introducing IT Systems into Operational Practice." 20.
- Björnfot, A., and Y. Sardén. 2014. "Prefabrication: A Lean Strategy for Value Generation in Construction."
- Cachon, G. P. 2003. "Supply Chain Coordination with Contracts." *Handb. Oper. Res. Manag. Sci.*, Supply Chain Management: Design, Coordination and Operation, 227–339. Elsevier.
- Choi, J., and J. S. Russell. 2004. "Economic gains around mergers and acquisitions in the construction industry of the United States of America." *Can. J. Civ. Eng.*, 31 (3): 513–525. NRC Research Press. <https://doi.org/10.1139/104-004>.
- Di Francesco Maesa, D., and P. Mori. 2020. "Blockchain 3.0 applications survey." *J. Parallel Distrib. Comput.*, 138: 99–114. <https://doi.org/10.1016/j.jpdc.2019.12.019>.
- Dresch, A., D. P. Lacerda, and J. A. V. Antunes Jr. 2015. *Design Science Research: A Method for Science and Technology Advancement*. Cham: Springer International Publishing.
- Dubois, A., and L.-E. Gadde. 2001. "The Construction Industry as a Loosely Coupled System - Implications for productivity and innovativity." 20.
- Fawcett Stanley E. 1995. "Using strategic assessment to increase the value-added capabilities of manufacturing and logistics." *Prod. Inventory Manag. J.*
- Fulford, R., and C. Standing. 2014. "Construction industry productivity and the potential for collaborative practice." *Int. J. Proj. Manag.*, 32 (2): 315–326. <https://doi.org/10.1016/j.ijproman.2013.05.007>.
- Giannoccaro, I., and P. Pontrandolfo. 2004. "Supply chain coordination by revenue sharing contracts." *Int. J. Prod. Econ.*, 89 (2): 131–139. [https://doi.org/10.1016/S0925-5273\(03\)00047-1](https://doi.org/10.1016/S0925-5273(03)00047-1).
- Hofmann, F., S. Wurster, E. Ron, and M. Bohmecke-Schwafert. 2017. "The immutability concept of blockchains and benefits of early standardization." *2017 ITU Kaleidosc. Chall. Data-Driven Soc. ITU K*, 1–8. Nanjing: IEEE.
- Jansson, G. 2010. "Industrialised housing design efficiency."
- Jeuland, A. P., and S. M. Shugan. 1983. "Managing Channel Profits." *Mark. Sci.*, 2 (3): 239. INFORMS: Institute for Operations Research. <https://doi.org/10.1287/mksc.2.3.239>.
- Jiang, R., C. Mao, L. Hou, C. Wu, and J. Tan. 2018. "A SWOT analysis for promoting off-site construction under the backdrop of China's new urbanisation." *J. Clean. Prod.*, Sustainable urban transformations towards smarter, healthier cities:

- theories, agendas and pathways, 173: 225–234.
<https://doi.org/10.1016/j.jclepro.2017.06.147>.
- Kim, Y.-W., and B. Rhee. 2021. “Rebate for Scheduled Delivery, Planning Reliability, and Supply Chain Coordination.” SSRN Scholarly Paper. Rochester, NY.
- Larson, P. D., and D. S. Rogers. 1998. “Supply Chain Management: Definition, Growth and Approaches.” *J. Mark. Theory Pract.*, 6 (4): 1–5.
<https://doi.org/10.1080/10696679.1998.11501805>.
- Lee, C. H., and B.-D. Rhee. 2008. “Optimal Guaranteed Profit Margins for Both Vendors and Retailers in the Fashion Apparel Industry.” *J. Retail.*, 84 (3): 325–333. <https://doi.org/10.1016/j.jretai.2008.07.002>.
- Lee, H. L., V. Padmanabhan, T. A. Taylor, and S. Whang. 2000. “Price Protection in the Personal Computer Industry.” *Manag. Sci.*, 46 (4): 467–482.
<https://doi.org/10.1287/mnsc.46.4.467.12058>.
- Lu, P., L. Qian, Z. Chu, and X. Xu. 2016. “Role of Opportunism and Trust in Construction Projects: Empirical Evidence from China.” *J. Manag. Eng.*, 32 (2): 05015007. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000401](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000401).
- Lu, W., X. Li, F. Xue, R. Zhao, L. Wu, and A. G. O. Yeh. 2021. “Exploring smart construction objects as blockchain oracles in construction supply chain management.” *Autom. Constr.*, 129: 103816.
<https://doi.org/10.1016/j.autcon.2021.103816>.
- Ma, C., X. Kong, Q. Lan, and Z. Zhou. 2019. “The privacy protection mechanism of Hyperledger Fabric and its application in supply chain finance.” *Cybersecurity*, 2 (1): 5. <https://doi.org/10.1186/s42400-019-0022-2>.
- Min, J. U., and H. C. Bjornsson. 2008. “Agent-Based Construction Supply Chain Simulator (CS2) for Measuring the Value of Real-Time Information Sharing in Construction.” *J. Manag. Eng.*, 24 (4): 245–254. American Society of Civil Engineers. [https://doi.org/10.1061/\(ASCE\)0742-597X\(2008\)24:4\(245\)](https://doi.org/10.1061/(ASCE)0742-597X(2008)24:4(245)).
- Nakamoto, S. 2008. “Bitcoin: A Peer-to-Peer Electronic Cash System.” 9.
- Nguyen, D. T., and P. L. Le. 2022. “Twenty-year application of logistics and supply chain management in the construction industry.” *Constr. Manag. Econ.*, 40 (10): 796–834. <https://doi.org/10.1080/01446193.2022.2110273>.
- Pasternack, B. A. 1985. “Optimal Pricing and Return Policies for Perishable Commodities.” *Mark. Sci.*, 4 (2): 166–176.
<https://doi.org/10.1287/mksc.4.2.166>.
- Perry, M. K. 1989. “Chapter 4 Vertical integration: Determinants and effects.” *Handb. Ind. Organ.*, 183–255. Elsevier.
- Qian, X. (Alice), and E. Papadonikolaki. 2020. “Shifting trust in construction supply chains through blockchain technology.” *Eng. Constr. Archit. Manag.*, 28 (2): 584–602. <https://doi.org/10.1108/ECAM-12-2019-0676>.
- Schoenberg, R. 2006. “Measuring the Performance of Corporate Acquisitions: An Empirical Comparison of Alternative Metrics.” *Br. J. Manag.*, 17 (4): 361–370.
<https://doi.org/10.1111/j.1467-8551.2006.00488.x>.

- Sigalov, K., X. Ye, M. König, P. Hagedorn, F. Blum, B. Severin, M. Hettmer, P. Hückinghaus, J. Wölkerling, and D. Groß. 2021. “Automated Payment and Contract Management in the Construction Industry by Integrating Building Information Modeling and Blockchain-Based Smart Contracts.” *Appl. Sci.*, 11 (16): 7653. Multidisciplinary Digital Publishing Institute. <https://doi.org/10.3390/app11167653>.
- Simon, H. A. 1988. “The Science of Design: Creating the Artificial.” *Des. Issues*, 4 (1/2): 67. <https://doi.org/10.2307/1511391>.
- Sunyaev, A. 2020. *Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies*. Cham: Springer International Publishing.
- Taylor, T. A. 2002. “Supply Chain Coordination Under Channel Rebates with Sales Effort Effects.” *Manag. Sci.*, 48 (8): 992–1007. <https://doi.org/10.1287/mnsc.48.8.992.168>.
- The State Council. 2016. *Opinions on Further Strengthening Urban Planning and Construction Management*.
- Thomas, D. J., and P. M. Griffin. 1996. “Coordinated supply chain management.” *Eur. J. Oper. Res.*, 94 (1): 1–15. [https://doi.org/10.1016/0377-2217\(96\)00098-7](https://doi.org/10.1016/0377-2217(96)00098-7).
- Tsay, A. A. 1999. “The Quantity Flexibility Contract and Supplier-Customer Incentives.” *Manag. Sci.*, 45 (10): 1339–1358. <https://doi.org/10.1287/mnsc.45.10.1339>.
- Von Stackelberg, H. 1934. *Market Structure and Equilibrium*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Wang, Y., and Y. Gerchak. 2001. “Supply Chain Coordination when Demand Is Shelf-Space Dependent.” *Manuf. Serv. Oper. Manag.*, 3 (1): 82–87. <https://doi.org/10.1287/msom.3.1.82.9998>.
- Wang, Z., H. Hu, J. Gong, X. Ma, and W. Xiong. 2019. “Precast supply chain management in off-site construction: A critical literature review.” *J. Clean. Prod.*, 232: 1204–1217. <https://doi.org/10.1016/j.jclepro.2019.05.229>.
- Weng, Z. K. 1995. “Channel Coordination and Quantity Discounts.” *Manag. Sci.*, 41 (9): 1509–1523. INFORMS: Institute for Operations Research. <https://doi.org/10.1287/mnsc.41.9.1509>.
- W.L. Cheng, E., H. Li, P. E.D. Love, and Z. Irani. 2004. “Strategic alliances: a model for establishing long-term commitment to inter-organizational relations in construction.” *Build. Environ.*, 39 (4): 459–468. <https://doi.org/10.1016/j.buildenv.2003.09.014>.
- Wu, H., P. Zhang, H. Li, B. Zhong, I. W. H. Fung, and Y. Y. R. Lee. 2022a. “Blockchain Technology in the Construction Industry: Current Status, Challenges, and Future Directions.” *J. Constr. Eng. Manag.*, 148 (10): 03122007. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0002380](https://doi.org/10.1061/(ASCE)CO.1943-7862.0002380).
- Wu, L., S. AbouRizk, and K. Li. 2022b. “System Dynamics Modeling of the Construction Supply Chain in Industrial Modularized Construction Projects.” *2022 Winter Simul. Conf. WSC*, 2409–2420.

- Yang, R., R. Wakefield, S. Lyu, S. Jayasuriya, F. Han, X. Yi, X. Yang, G. Amarasinghe, and S. Chen. 2020. "Public and private blockchain in construction business process and information integration." *Autom. Constr.*, 118: 103276. <https://doi.org/10.1016/j.autcon.2020.103276>.
- Yoon, J. H., and P. Pishdad-Bozorgi. 2022. "State-of-the-Art Review of Blockchain-Enabled Construction Supply Chain." *J. Constr. Eng. Manag.*, 148 (2): 03121008. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0002235](https://doi.org/10.1061/(ASCE)CO.1943-7862.0002235).
- Zavolokina, L., N. Zani, and G. Schwabe. 2019. "Why Should I Trust a Blockchain Platform? Designing for Trust in the Digital Car Dossier." *Extending Boundaries Des. Sci. Theory Pract.*, Lecture Notes in Computer Science, B. Tulu, S. Djamasbi, and G. Leroy, eds., 269–283. Cham: Springer International Publishing.
- Zhang, Z., Z. Yuan, G. Ni, H. Lin, and Y. Lu. 2020. "The quality traceability system for prefabricated buildings using blockchain: An integrated framework." *Front. Eng. Manag.*, 7 (4): 528–546. <https://doi.org/10.1007/s42524-020-0127-z>.
- Zhong, R. Y., Y. Peng, F. Xue, J. Fang, W. Zou, H. Luo, S. Thomas Ng, W. Lu, G. Q. P. Shen, and G. Q. Huang. 2017. "Prefabricated construction enabled by the Internet-of-Things." *Autom. Constr.*, 76: 59–70. <https://doi.org/10.1016/j.autcon.2017.01.006>.