AN EXPERIMENTAL SMART CONTRACT-ENABLED BLOCKCHAIN FRAMEWORK FOR SUPPLY CHAIN

KANG PUNNRY

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR MASTER DEGREE OF SCIENCE
IN LOGISTICS AND SUPPLY CHAIN MANAGEMENT
FACULTY OF LOGISTICS
BURAPHA UNIVERSITY
2023

AN EXPERIMENTAL SMART CONTRACT-ENABLED BLOCKCHAIN FRAMEWORK FOR SUPPLY CHAIN

KANG PUNNRY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาการจัดการโลจิสติกส์และโซ่อุปทาน
คณะโลจิสติกส์ มหาวิทยาลัยบูรพา
2566

AN EXPERIMENTAL SMART CONTRACT-ENABLED BLOCKCHAIN FRAMEWORK FOR SUPPLY CHAIN

KANG PUNNRY

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR MASTER DEGREE OF SCIENCE
IN LOGISTICS AND SUPPLY CHAIN MANAGEMENT
FACULTY OF LOGISTICS
BURAPHA UNIVERSITY
2023

The Thesis of Kang Punnry has been approved by the examining committee to be partial fulfillment of the requirements for the Master Degree of Science in Logistics and Supply Chain Management of Burapha University

Advisory Committee

Principal advisor

....

(Associate Professor Dr. Nakorn Indra-payoong)

Examining Committee

Sarawut. .... Principal examiner

(Assistant Professor Dr. Sarawut Jansuwan)

.... Member

(Assistant Professor Dr. Pairoj Raothanachonkun)

.... Member

(Associate Professor Dr. Nakorn Indra-payoong)

.... Dean of the Faculty of Logistics

(Associate Professor Dr. Nakorn Indra-payoong)

5 July 2023

This Thesis has been approved by Graduate School Burapha University to be partial fulfillment of the requirements for the Master Degree of Science in Logistics and Supply Chain Management of Burapha University

Witawat Jangiam. .... Dean of Graduate School

(Associate Professor Dr. Witawat Jangiam)

6 July 2023.

61910003: MAJOR: LOGISTICS AND SUPPLY CHAIN MANAGEMENT;
M.Sc. (LOGISTICS AND SUPPLY CHAIN MANAGEMENT)
KEYWORDS: BLOCKCHAIN, SMART CONTRACT, SUPPLY CHAIN,
SCALABILITY, FLEXIBILITY, DECENTRALIZATION

KANG PUNNRY : AN EXPERIMENTAL SMART CONTRACT-ENABLED BLOCKCHAIN FRAMEWORK FOR SUPPLY CHAIN. ADVISORY COMMITTEE: NAKORN INDRA-PAYOONG, 2023.

The introduction of decentralized distributed systems such as blockchain, with support for smart contract, has created a new paradigm shift in how business networks can be managed. Case in point, supply chain has been plagued with issues like flexibility, scalability, and decentralization, all of which can potentially be resolved using alternative network management and design. This research attempts to solve these issues using an experimental blockchain network design, with support for smart contract, whose network architecture and configurations are designed for a local testing environment. The author also uses a series of different network configurations and a custom smart contract to further stress the network performance. This network, which is based on Hyperledger Fabric, achieves a significant usability and flexibility with feasible network scalability. We also see the network architecture and performance thoroughly analyzed for potential system bottleneck and key areas to improve. The result is a rising trend of average transaction response time, with increasing peer nodes in each experiment. This proves that blockchain-based networks have the potentials to revolutionize traditional forms of supply chain network design and management. Finally, a more advanced and customized form of this network design can be used to test a more complex supply chain network in future studies.

ACKNOWLEDGEMENTS

The author would like to express his deepest appreciation to his advisor Assoc. Prof. Dr. Nakorn Indra-Payoong, for his helpful advice, valuable recommendation, and patient support with encouragement throughout the duration of this work. The author would also like to extend his sincere gratitude to the examination committees, Asst. Prof. Dr. Sarawut Jansuwan (Principal examiner) and Asst. Prof. Dr. Pairoj Raothanachonkun (committee member) for playing a decisive role in constructive criticism and providing useful suggestions, as well as insightful comments and knowledge to make this research even more practical.

This work is financially supported by Royal Scholarship under His Royal Highness Princess Maha Chakri Sirindhorn Education Project to the Kingdom of Cambodia (2018).

The author also wishes to express his sincere thanks toward Assoc. Prof. Dr. Nakorn Indra-Payoong (Dean of Faculty of Logistics) and Asst. Prof. Dr Sarawut Luksanato for giving him a great opportunity to pursue this master's degree. Great appreciation is also given to all professional lecturers for teaching and sharing the practical knowledge and experience in this field. The author also wants to express his gratefulness to all faculty and university staffs for their kindness, support and helpful service during the author's study of Master of Science in Logistics and Supply Chain Management at the Faculty of Logistics, Burapha University, Chon buri campus, Thailand.

Many thanks to all Cambodian and Thai seniors, friends, and classmates who has always supported his with profound belief in the research. Especially, the author would like to express his sincere thanks to his beloved family who mentally and financially supported his on this journey. Without all the precious contribution and encouragement of all those people, the completion of this work would not have been possible.

Kang Punnry

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1
INTRODUCTION

Background

Agricultural revolution brought about a drastic change in humanity's survivability as a whole and is also responsible for the then future industrial revolutions, of which undeniably fostered technological dependency seen in today's societies. However, it is also one among the least digitized industry on earth. Even by employing hundreds of millions of people around the world, agricultural products are still mostly treated as traditionally as possible, from the way it is planted and farmed, down to the way it is distributed and sold along the supply chain to the end customers.

Agri-Food Supply Chain (AFSC) is by far facing as many uncertainties as it has ever seen before, ranging from demand, to consumer preference and environmental impact. One factor not given enough attention is its dedication to fairness to all stakeholders, especially food producers. One such volatility is the price—such that even if they produce for domestic consumption, of which they almost always do, global price will still profoundly affect their incomes (Segal & Le Nguyet, 2019, p. 06). This means it is not always considered to be equivalent to minimum wage compared to other industries since most farmers earn only somewhere between $2-$6 per day. As a result, hundreds of millions of people worldwide end up in substandard living conditions, lacking many of the basic necessities. The volatility of price emanates from the lack of transparency in the supply chain, particularly regarding both the information and physical flow of product. Coupled this with the shift in consumer preference to a more sustainable and organic purchases, it is a clear telltale sign that there has to be changes to the traditional model.

Blockchain, on the other hand, has been gaining a lot of traction as a new technological revolution in decentralized distributing system, by allowing transparency, anonymity, and security to coexist together. It has the potential to shape the world in almost every discipline, ranging from economics to politics, and even business (Kang & Indra-Payoong, 2019, p. A2; Swan, 2015, p. 30).

Statement of problem

Supply Chain is one of the most challenging industries in terms of complexity. During the 2020 pandemic, issues like traceability, logistics, manufacturing, increasing operational costs, uncertainties, shifting consumer preferences, and more have entirely changed some industries and seriously transformed others into sorts of hybrid business models. For instance, many retailers witnessed first-hand brutal decline in sales of non-essential items like entertainment, clothing, and such; while sales in food and household items are rising due to work-from-home policy implemented in many countries. Furthermore, agrifood industry has absorbed these issues in addition to its many existing supply chain issues like material scarcity, lack of technological integration, inflation, and more. As a result, many stakeholders are experiencing extra pressure and adversity on top of market competition. Reasons such as buyer's power and barrier to entry have consistently undermine producers' ability to obtain information, and consequentially, to compete effectively. Food producers are typically taken advantaged of in terms of strategic business decisions. They are known to be discriminated against and treated poorly both by their business partners and employers (Mark, 2006; Segal & Le Nguyet, 2019, p. 02); whereby the rule of law also does not necessarily favor their wellbeing and competitiveness in the industry (Polack, Cotula, Blackmore, & Guttal, 2014, p. 07). Additionally, a study made by (Mazoyer, 2001, p. 02) describes the imbalance of opportunities received by farmers around the world, especially those living in Least Developed Countries (LDCs). Approximately 2% of the world farmers had access to modern motorization of farming tool and equipment, meaning they could utilize fertilizers, special seeds and plants, and other supports. Moreover, only two thirds of the world's farmer were supported by the green revolution which allows them to obtain specially bred seeds and plants, fertilizers, as well as livestock, but at the expense of modern motorization and mechanization. This results in about one third of the farmers not receiving any support in the form of animal utilization, special crops, fertilizers, and have to resort to using manual labor to grow their crops. This revelation, combined with global trade, exhibits the truth that an increase in productivity of some farmers continue to overshadow the hard work and lack of support for other farmers in LDCs. Furthermore, the same increase in productivity generates the adverse effect on prices of agricultural product on a worldwide scale, hurting the profitability and well-being of poor, rural farmers.

The lack of traceability presented in today's supply chain industry poses serious concerns regarding transparency and sustainability. The outbreak of Bovine Spongiform Encephalopathy (mad cow disease) and E. Coli in USA drew significant discussion in food industry about the safety of food products. There is an estimation that food-related illnesses amount to millions of recorded sickness and 9,000 deaths yearly in USA alone (Pouliot & Sumner, 2008, p. 17). Moreover, the lack of transparency in value shared across stakeholders may also lead to abuse of power by large corporations and middlemen, which in turn, dramatically reduce farmers' share in the value chain year by year (Bunte, 2006, p. 39).

This study aims to discover the potential usage of integrated blockchain technology in improving scalability and transparency in agri-food supply chain. This could help food producers gain a better living and profit out of current supply chain where abuse of power is indisputably encouraged by multinational corporations. If done correctly, Blockchain could be the future of fair and equal wealth distribution economy done using a decentralized distributed system.

Research questions

This research seeks to answer the following questions:

1. What is the current state of blockchain applications in supply chain?
2. To what extent can blockchain be scaled and utilized in a local testing environment?
3. How can blockchain be implemented in a supply chain configuration in order to sustain scalability and flexibility?

Research objectives

The research aims to verify that farmers/food producers are in fact being taken advantages of during their business undertakings with other actors in the supply chain. The main objectives, however, are:

1. To study and compare contemporary studies in literature with existing blockchain technology in supply chain.

2. To design a blockchain framework with various blockchain system configurations.

3. To experiment on a smart contract-enabled blockchain network based on Hyperledger Fabric in order to demonstrate its scalability and flexibility on a simplified supply chain network.

Limitation and scope of study

This research contains the following limitations:

1. This experiment is confined to a local testing environment where blockchain configurations are tested within a single machine.

2. This study is a demonstration of a blockchain framework utilization in a simplified supply chain network.

3. It utilizes a simplified supply chain network as opposed to a full-fledged operational blockchain network.

CHAPTER 2
LITERATURE REVIEW

Agri-food supply chain (AFSC)

Agri-food industry has been feeding people since 20th century while staying mostly non-progressive in term of modernization. Most of the produce are often made by uneducated, small-scale farmers in rural parts of the world, and they are usually taken advantage of by middlemen and distributors—who are looking into squeezing more profits out of those farmers, whether through ethical means or not. As reported by (Schutter, 2014, pp. 38-43), Unfair trading practices acted upon food producers could lead to unsustainable livelihood, child labor, and even environmental degradation. The traditional form of agri-food supply chain relies heavily upon uncontracted wholesale buy-outs from farmers which directly leads to supplies/output and quality to remain mostly inconsistent, susceptible to unnecessarily long lead times, and the impossibility of product traceability in times of crisis. In addition, Tsolakis, Keramydas, Toka, Aidonis, and Iakovou (2014, p. 48) state that one of the most critical setbacks in agri-food sector is the complexity and cost efficiency of the supply chain as it requires a multi-tier supply chain approach to solving the problems of unmatched flow of goods, both upstream and downstream the chain itself. Agri-food retail firms help accelerate this system by deploying the use of vertical and horizontal integration, market segmentation, product offerings, branding of product lineups and companies, as well as trade in a global context as a whole.

The progress made in Information and Communication Technologies (ICT) in Logistics, food quality, government policies on food regulations, the arrival of modern multinational food firms, vertical and horizontal integrations, and a plethora of other disciplines led to the adoption of Agri-Food Supply Chain (AFSC) by respective stakeholders (K Chen, 2006, pp. 02-04). Typically, an Agri-Food Supply Chain takes time from farming to the hands of a consumer via a long sequence consisting of: Farming (land preparation to harvesting), processing, testing, packaging, warehousing, transportation, distribution, marketing, and even Corporate Social Responsibility (CSR) (Iakovou, Vlachos, Achillas, & Anastasiadis, 2012, pp. 06-10).

Stakeholders in AFSC normally consist of government and international organizations and private firms, the latter of which is composed of farmers, middlemen, research firms, suppliers, traders, logistics firms, food shops, and others (Jaffee, Siegel, & Andrews, 2010, pp. 35-37). In addition, Tsolakis et al. (2014, pp. 50-56) also present the first generic hierarchical decision-making framework in the context of AFSC as an alternative. The framework introduces Strategic, and Tactical and Operational Decisions as the main components. Strategic decisions consist of: selection of farming technologies, developing an investment portfolio, fostering supply chain partnering relationships, configuration of supply chain networks, establishing a performance measurement system, ensuring sustainability, and adoption of quality management policies. Tactical and operational decisions are composed of: planning of harvesting operations, planning of logistics operations, and supporting food safety via transparency and traceability.

This literature review aims to point out the overlooked unfairness of trade practices along the agri-food supply chain and the studies made in promoting sustainability in food industry. The reports and studies are reviewed, filtered, and compiled based on relevance and the impacts they made for fair and transparent, as well as sustainable supply chain for food producers as a priority.

Agri-food supply chain model, as mentioned above, used to rely on informal contract and immediate buy-outs from major middlemen and distributors who, for the most part, exert immense market power and pressure, forcing food producers to adapt to fewer, less market share and profitability respectively. The farmers' market share had been declining over a 16 years period, from 1995 to 2011, while the remaining shares went to food industry and retail plus food services. Concurrently, farmers' profit margins are also being squeezed out further by newer sustainable farming methods and regulations imposed by their clients and the government (Healy, 2015). Farmers' well-being and roles in society have largely been ignored; the crops once planted to feed the local people have now been discouraged in favor of popular and in-demand seeds meant for export (Madeley, 2000, p. 55). Instead of helping small, vulnerable farmers thrive, globalized trade only serves to somehow worsen the situation indirectly by encouraging governments to use lands for export crops, further undermining the values of food with foreign currencies. The agri-food industry is plagued with unfair trade practices seen across most countries, from developed nations to, especially, least developing countries who rely on agriculture the most.

Fairness in agri-food supply chain

In table 1.1 below, significant studies regarding fair trading practices toward food producers are examined and reviewed to project the state of livelihood food producers are facing. The issues of unfair trading practices have always been present in agri-food supply chain since before fair-trade association was formed. The cases of supermarkets controlling more than half of the total food market share in America and Europe is bad enough that they may directly or indirectly influence the price of agricultural products. Their growths have come from cut-throat competition and their abilities to extract extra profits from food producers down in the supply chain. Kevin Chen, Shepherd, and Silva (2005, p. 05) also found that firms who source produce from producers tend to do so from individual, small farmers rather than community-based ones, particularly because those producers lack production information, intelligence, negotiating power, and general competitiveness to negotiate any contractual terms at all. Making the matters worse, food supply chain in Asia is also delineated by almost total involvement from big supermarket chains that possess enormous market power over small suppliers. In UK, (Hingley, 2005, p. 05) noticed that power in agri-food industry is significantly imbalanced and overlooked. The research regarding fairness and power gap is undermined and ignored by other researchers. In a research conducted by (Hellberg-Bahr & Spiller, 2012, p. 91) in Germany, they concluded that around 40% of farmers participated in the survey expressed they didn’t feel they were treated fairly by their supply chain partners. Furthermore, they found there are positive correlations between higher payments and the acknowledgment of being treated fairly as seen by farmers. Additionally, Reliability and relationship quality proves to be even more important from farmers’ perspective on the issue, mainly because they feel they can rely on their partners in the long run.

Needless to say, the reality concerning fairness from the market says otherwise. The whole food supply chain is driven by price resulted from unchecked competition. Another study done by (Blizkovsky & Berendes, 2016, p. 108) brilliantly illustrates the power imbalance between farmers and suppliers: “Asymmetric scopes of power to enforce self-centered profit distributions and/or possibilities to actively influence certain actors to conduct economic performances according to one’s own concepts and interests form a threat towards a fair functioning of bargaining practices within the food supply chain.” Strong actors obtain even more power because therein lies lack of competition in the industry. This lack of competition fuels the buyer’s bargaining power of suppliers and retailers, forming what is essentially oligopsony, and thus forcing farmers to sell at much lower prices than necessary. This, in turn, further fuels the tendency for other actors to engage in Unfair Trading Practices (UTPs). Likewise, (Fair World, 2018, pp. 08-09) reports similar occurrences for farmers in rural developing countries. 84% of worldwide farmers rely on 2 hectares of land or less to feed their communities and environment. However, the issue of global export of agricultural product threatens their well-beings and communities. By allowing for corporations to exploit these vital food producers, the communities face numerous obstacles, ranging from land grabbing, unfair trade practices, uneven wealth distribution, low and volatile prices, and most importantly, the corporations’ control of the food supply chain itself.

Sustainability in agri-food supply chain

In Table 1.2, the topic of sustainability in agri-food industry is brought up along with recent studies. Unsurprisingly, one study states price changes at the consumer level may not be transmitted to the suppliers after all. In the long run, through acquiring market power, price changes and its risks are shifted to food producers through various means (price transmission not transmitted, asymmetrical price changes, and the lag of time between price changes), while leaving some rooms for suppliers to still make a profit (Bunte, 2006, p. 41). Meanwhile, (Seuring & Müller, 2008, p. 460) identifies economic aspect among the three aspects of sustainability as the most important one, arguing that without long-term profit an enterprise will not survive the competition. The study reveals that lack of customer demand and government regulation threatens the producers’ businesses. Furthermore, it is known that competitiveness of the supply chain lies deeper than the pricing and economic aspect of sustainability. Turns out that there should be a case-by-case study that reflects on the multi-objectiveness of the whole supply chain, preferably focusing more on food producers rather than the suppliers up the stream.

Another important part of sustainability, namely traceability, is seen as a key solution to supply chain’s adaptiveness to modern-day fast-moving markets. Improving food safety and traceability does encourage customers to pay more, and thereby has a potential to open up a new type of market to increase both safety and profitability of food producers (Pouliot & Sumner, 2008, p. 19). On top of this, (Thompson et al., 2007, p. 13) provides an excellent questions as to how can poor rural farmers negotiate their way out of unfairness in the face of numerous challenges in agrifood industry, from market failures to public intervention? The study unearthed vital issues, from ineffective and irresponsive agricultural system, to changes of the modern supply chain and customer trend. The study proposes some keys aspects to solving sustainability issue such as ecological care, modernized agricultural technology, government regulation, and the dynamics of modern supply chain production. What is missing the most from today’s cluttered food supply chain is the dynamics of a creative, technology-driven supply chain that is both fair and transparent for all stakeholders. In a similar fashion, (Fritz & Matopoulos, 2008, pp. 08-10) argue that corporations basically neglect the economic aspect of sustainability of its own suppliers and instead focus more on social and environmental aspects which help them sell more products. It is further argued that food producers face immense risks and pressure from its own buyers who can do whatever it wants and command the farmers to adopt any sustainable production methods, mostly at their own expense. The system simply lacks the modernized, fair, and sustainable model in which all stakeholders could participate. Leaving only government regulation and the implementation of sustainability in the hands of transnational supermarket chains only exacerbate the circumstances up to the point where unfair trading practices are prevalent--and the imminency of poor rural farmers’ economic collapse is not far ahead.

In 2050, there will be a total of 9 billion people in the world waiting to be fed, while land use couldn’t be expanded anymore due to devastating environmental impact it could pose to all life on earth. Thus, the only logical way is to increase productivity and efficiency with utmost focus on sustainability. As proposed by

(Chiengkul, 2017, pp. 10-14; Hendrickson, Howard, & Constance, 2017, p. 39), modern-day agrifood industry is capitalistic in nature and should not represent such a vital aspect of human lives like rights to food. Take international trade as an example, we see price dumping happening a lot on an alarming scale that it challenges the livelihood of all farmers in developing nations. The flawed assumption about the existence of perfect competitions has proved to be a huge mistake as negative externalities and lack of sustainable production threaten the long-term wellbeing and livelihood of millions of people around the globe. The study recommends sustainability food sovereignty as potential solutions to sustain the global food supply chain in 21st century. In addition, not only sustainability has proven to be save costs and environment in the long run, but it is also shown to deeply correlate to performance and circular supply chain as well (Lai & Wong, 2012, p. 278).

With that being said, the study focuses on creating a fair, transparent, and sustainable model in a low-risk-high-reward manner based on Blockchain and Smart Contract to leverage stakeholder power for those of food producers and level the playing field in food supply. This research could potentially adopt a platform-based system to form a community based on fairness, respect, and transparency in a cutting edge secured environment.

Blockchain applications in supply chain

Distributed computing technology has been around for decades, and only over the past ten years has blockchain been utilized as a form of trustless distributed computing. The early form of Blockchain usage is in Bitcoin as a form of distributed ledger, of which Blockchain's application is extremely limited to recording tamper-proof messages and transactions. Later on, Ethereum (Tikhomirov, 2017, p. 01) emerged as an evolution of distributed computing by introducing the concept of smart contract as a foundation on which blockchain can be leveraged as a platform for many critical purposes. Eventually, along with combination of distributed computing system and encryption, Blockchain has evolved into a multi-purpose tool, making it suitable for various industrial and business applications. Specifically, Blockchain has a special inherent quality of being a decentralized system, allowing for transparent data distribution without necessarily exposing sensitive information to unauthorized parties. As reported by (Lin et al., 2020, p. 143922), Blockchain’s decentralized feature is one of the most effective solutions for data integrity and distribution of information across all stakeholders as it allows for an unprecedented information exchange and collaboration across the whole supply chain. On top of that, de Carvalho, Naoum-Sawaya, and Elhedhli (2022, p. 856) concludes that Blockchain offers considerable potential to redefine supply chains due to its data immutability and transparency shared between its network members. The way Blockchain works is by distributing exact copies of ledgers containing information about transactions and its metadata to its members. Then additional future transactions and metadata will be further distributed to members through its consensus algorithm of which network participants have already agreed upon. Smart Contract enables the Blockchain network to support various level of interactions between members and assets/objects. Members can send, receive, create, update, and execute other functions as per the required complexity of the business interactions. Smart contract can also be programmed with legality in mind, further decentralizing the network from any centralized legal entity. Due to its tamper-resistant and auditable nature, smart contract is a fitting match for distributed system like Blockchain (Mohanta, Panda, & Jena, 2018, p. 01).

It has been found that food producers and small manufacturers of supply chain experience unfair treatment because of centralized authoritative pressure. The pressure exerted on those producers can be categorized as a form of practice from various parties up the stream (Kang & Indra-Payoong, 2021, p. 03). Similarly, Supply Chain as a whole is also facing unfair trading practices, as well as various other issues such as: lack of stakeholders’ collaboration, operational inefficiency, especially concentration of power (centralization) as well as scalability issue in a huge supply chain information sharing network. These problems are remedied using various advanced technology to enhance data sharing, security, and scalability. Supply Chain has evolved and grown significantly since its inception; however, due to the increased complexity, it can no longer be feasibly solved using conventional and established technological frameworks. For instance, transparency issue in Supply Chain requires a system so focused on information sharing and collaboration—the sort of issues for which Blockchain and smart contract is created. A study conducted by Montecchi, Plangger, and West (2021, p. 11) affirms that there needs to be more researches done into transparency issues of supply chain due to "…the growing complexity of supply chains and unpredictable continuous changes in the external environment…". The same study also reiterates that future researches should explore how various frameworks can add additional value and competitive edge to the supply chain stakeholders. Blockchain network helps facilitate transactions from peer-to-peer, member-to-member to be transparent when required, especially without the need for middlemen in Supply Chain—which greatly assists in enabling all stakeholders to participate and voice their opinions in challenging the established norms of trading practices (Xiong, Dalhaus, Wang, & Huang, 2020, p. 02). This eventually creates trust which is derived from a system that doesn't require its members to trust each other at all. In addition, Pournader, Shi, Seuring, and Koh (2020, pp. 15-29) examined various researches on Blockchain and smart contract applications, many of which take aim at solving trust and transparency issues in supply chain. It also concludes that Blockchain technology enables transparency, and in turn, creates trust as it decentralizes the flow of information sharing, and allows for stakeholders to exercise their powers in a fairer and more transparent environment.

Because Supply Chain is a complicated set of processes, often involving multiple business parties, it requires a lot of collaboration, information sharing, management, and ultimately the integration of all vital processes in order to be treated as a complete and optimized set of system. Therefore, information flow is an indispensable part of collaboration in the whole supply chain. It is well-known that Information Technology (IT) plays a significant role in various Supply Chain activities since it allows for an increased amount and variety of information to be shared between business allies (Vanpoucke, Boyer, & Vereecke, 2009). Moreover, the integration of technology tremendously help in the context of information flow as inter-firm relationships and costs are optimized versus the traditional means of information sharing. As such, accurate, on-time, and visible communication of information between business partners is essential to ensure supply chain consistency, competency, and effectiveness (Singh, 1996, p. 30). As businesses strive to not only develop long-term relation-based value creation in Supply Chain, but also by utilizing information shared as effectively as possible firms can foresee a need to adopt experimental technology—by which its mere presence is sufficient to enable a better integrated Supply Chain (Patnayakuni, Rai, & Seth, 2006, pp. 39-40). As Blockchain and Smart Contract provides unparalleled level of security and information sharing in various experimental case studies, it is obvious to firstly utilize its properties in small-scale designs that resemble working supply chain networks. Blockchain-enabled network has the capacity to enrich the variety and amount of information exchanged between business partners to such an extent that foreign entities cannot tamper with the integrity of the data shared. Thus, blockchain's potential in many areas of supply chain information flow management is considered as the solution to many existing information flow managements in optimizing supply chain operation. Better information management also fosters transparency and auditability of supply chain operation, which in turn improves supply chain efficiency and allows for a more integrated operation for all stakeholders (Kersten, Seiter, von See, Hackius, & Maurer, 2017, pp. 27-28).

On the other hand, there exists many researches made in supply chain industry using Blockchain technology, most of which focus on either identifying new frameworks or echoing the existing Blockchain applications in Supply Chain. Case in point, a study done by Dutta, Choi, Somani, and Butala (2020, p. 22) analyzed a total of 178 articles related to Blockchain technology in supply chain, with applications on various parts of supply chain and logistics. As said, with most of the studies exploring the concept of integrating Blockchain and smart contract into supply chain, the minority remaining researches seek to apply certain designs and concept to cases studies; there have been an absent in studies done on experimenting or analyzing of information flow of such concepts in Supply Chain. Another example by Kakarlapudi and Mahmoud (2021, p. 12) corroborates that even if there exist many conceptual papers with prototypes, there are still some of them with "…no implementation or evaluation details…". Most conceptual studies, with or without actual applications of such designs, do not prioritize network design testing and analysis, as well as their implications on Supply Chain. It can be summarized as due to the relative infancy of the technology, whereby it is still in its early stages of emergence, too premature to be examined in terms of long-term usability. This occurs primarily because of the gap in researches, implementations, and long-term viability of such concepts (Hackius &

Petersen, 2017, p. 07). Furthermore, Antonucci et al. (2019, p. 6136) states that Blockchain has been gaining grounds as the platform for solving traceability and transparency issue of Agri-Food Supply Chain, while also reaffirming that only a handful of practical applications are present. Besides, there has been no mention of design vulnerabilities in the reviewed studies. Also, according to (Wamba & Queiroz, 2020), top 20 most globally cited publications related to Blockchain technology in Supply Chain were mostly works on privacy, security, smart contracts, and Blockchain architecture. The researches referenced in this study were further explained to be experimental in nature, with each focusing on different aspects of supply chain traceability, security, operation, and information management—with a noticeable lack of studies made on designing and testing of such designs. Based on these studies we see that Blockchain technology is an emerging technology where its utility and scope of applications in Supply Chain are still being determined by scholars and professionals alike.

From Table 1, here we have some current implementations of Blockchain technology in Supply Chain. These applications in literature attempt to uncover the usability of Blockchain in various specific scenarios. For instance, a study done by (Cocco et al., 2021) utilizes NFC and RFID with Blockchain and integrate them into a traditional bakery supply chain. This allows for a more transparent and auditable supply chain where customers can see the journey of the product, from raw material to the end product they are consuming, while enabling certain supply chain authority to monitor the product quality and working practices. However, this Blockchain network is permissionless, meaning everyone can join the network and see the information being passed along the transactions, which is not ideal in considering all of the information in Supply Chain is made transparent with no customization options. The approach to Blockchain integration mentioned here works best for the transparency and traceability issues in food Supply Chain where safety and quality are of utmost importance. Another study by Reddy et al. (2021) also experimented with the use of IoT and blockchain to combat transparency issue in Supply Chain Management. The proposed framework aims to decentralize information sharing and to eliminate single point of failure as with traditional centralized server-based systems. This design also leverages the usage of Blockchain with IoT and smart contracts in a local setting. Sensors are placed at various points within the Supply Chain so that it can be used with smart contracts to generate relevant info for customers and other stakeholders. Farmers are able to see info relating to soil quality, herbicides, and other factors impacting the end quality of product. Customers are also able to use a web interface to track product info and overall transparency from farming to their tables. The authors found that ultimately this blockchain network, with the use of smart contract, is able to achieve transparency, immutability, and traceability for the whole supply chain.

Studies made by (Adamashvili, State, Tricase, & Fiore, 2021; Chiranjeevi, Tripathi, & Maktedar, 2021) emphasize on traceability and security in Supply Chain. The former focuses more on Supply Chain mapping and simulation with Blockchain integration into auditability and tracking of Supply Chain product in case of quality issues. It found that Blockchain helps decrease inefficiencies and disorganizations along the Supply Chain, as well as simplifying information sharing among stakeholders as they are given the capability to track and audit the product's quality. The latter study concentrates on a generic framework, upon which can be used with Blockchain and smart contract as a solution to tracking prices and overall traceability. Combined with ERP, the study focuses on fast information sharing and tracking that empowers stakeholders to be able to participate in a more transparent Supply Chain. On the other hand, Lau, Liu, and Au (2021) proposes a generic blockchain system for Supply Chain traceability that adopts a hybrid Blockchain architecture. The design allows for traceability via smart contract, immutability and data integrity by Blockchain architecture, and increased efficiency and transparency with immutable records of transactions and other records. Similarly, Sund and Lööf (2019) conducted an extensive research on Blockchain application in Supply Chain traceability in which a Blockchain network is designed and implemented on IKEA Supply Chain. The system design includes smart contract compatibility and off-chain data storage to more effectively trace and track thousands of products across many different categories. Combined with smart contract, users can transfer ownership of products along the Supply Chain and have those records be safely and immutably stored.

On top of this, Sathya, Nithyaroopa, Jagadeesan, and Jacob (2021) implemented Blockchain and smart contract in an experimental study to determine its security and decentralization capabilities. They found the network to have increased efficiency, transparency, and faster transaction. The study integrates Blockchain and smart contract into Food Supply Chain where information relating to products are recorded along the Supply Chain using Ethereum network. However, it is still limited in nature and has not proved to be useful beyond experimental stage. Also, as mentioned above, de Carvalho et al. (2022) found that by modeling Blockchain as a key component into the Supply Chain, certain information can be leveraged as features, leading to monetization of products into premium categories. This leads to increased profitability, transparency, traceability, and efficiency in Supply Chain as a whole. Moreover, by strategically deploying Blockchain in particular key areas of Supply Chain results in better product differentiation due to traceability and transparency, as well as more profitability and higher surplus for consumer. The study also states that the adoption of Blockchain in partial areas of Supply Chain is the key to optimize profitability, efficiency, transparency, quality, and cost effectiveness. Similarly, Vo, Nguyen-Thi, and Nguyen-Hoang (2021) experimented on Blockchain-enabled Supply Chain with sustainability and traceability as the main objectives. The authors create a Blockchain architecture network capable of tracking food in Supply Chain, from production to retail whereby the network can be used in tandem with smart contract to track products and improve efficiency of Supply Chain as a whole—and subsequently build consumer trust. Another study by Surjandari, Yusuf, Laoh, and Maulida (2021) also proposes a Blockchain network based on Hyperledger Fabric as a platform for a more decentralized and transparent Supply Chain. This research mainly focuses on the HALAL Supply Chain where product origin and processing are highly sensitive and vital for end consumers. The authors made use of an older version of Hyperledger Fabric (version 1.4.3) to design a Blockchain-enabled HALAL Supply Chain, completed with smart contract to demonstrate its potential and capability as a solution for increased decentralization and transparency in Supply Chain.

Inspired by the issues in conventional centralized traceability solutions in Supply Chain, Sunny, Undralla, and Pillai (2020) set out to offer an overview of the state of Supply Chain Traceability based on Blockchain. By reviewing 27 articles on Blockchain Traceability system for Supply Chain, the authors found that Blockchain traceability has a variety of usage within Supply Chain as it can be implemented to counteract counterfeit products, monitoring business processes, and much more. Also, the designs analyzed in this study help achieve visibility issues in conventional Supply Chain systems. However, even with Smart Contract integration, the current state of Blockchain is still far too novel and incomplete to be reliably used; thus, the frameworks reviewed are mostly used with the added integration IoTs, as well as conventional systems. This allows for an ample space for experimentation without disrupting the supply chain itself. Additionally, Blockchain's potential is still yet to be determined, and that makes designing and implementing such systems to be extremely dynamic and disrupting for conventional systems. Issues like scalability, consensus, and lack of a standard framework deter many potential enterprises and stakeholders from experimenting. The authors also specify that most of the applications reported in literature are mainly conceptual in nature. Part of the reason why, argued the authors, actual applications are so rare is primarily because of Blockchain's infancy and lack of new experiments in this field.

Table 1 Literature review of fair-trade practice in agri-food supply chain

Nº	Author(s) (Year)	Title	Findings	Challenges
1	Hingley (2005)	Power Imbalance in UK Agri-Food Supply Channels: Learning to Live with the Supermarkets?	- Power is noticeably imbalanced in Agri-food industry, favoring the buyers. - Inequality among stakeholders, especially food producers are often overlooked.	- Legislative actions are historically ineffective against essentially monopoly powers. - Horizontal and Vertical integration require immense effort and coordination.
2	Kevin Chen et al. (2005)	Changes in Food Retailing in Asia	- Agribusiness firms have a tendency to avoid using wholesalers and directly contact individual farmers for supplies. - Lack of access to Production technology, limited information, intelligence on buyers and prices, negotiating power, and financing heavily prohibit farmers to take advantages of modern supply chain. - Produce supply chain in Asia is delineated by heavy involvement of urban supermarket chains.	- Capitalism-based economic system poorly leverages the livelihood of farmers. - New legislations targeted Supermarket chains could hurt the livelihood of consumers who benefit from fierce competition. - The need to modernize and enforce public services and regulations in rural areas remain substantial in developing countries.
3	Padel and Zander (2010)	Regional production and ‘Fairness’ in organic farming: Evidence from a CORE Organic project	- Fair and equitable financial returns for farmers are expressed by both the producers and consumers, while the European organic regulation only partly covers this issue. - Organic and transparent food system is demanded by vital stakeholders to be implemented based on local supply chain structure. - Local and regional supply chain structure could reinvigorate the well-being and fairness of local farmers.	- The term "Fairness" can be misleading without specifying criteria and framework upon which an agreement is made. - Misguided usage of the term can result in loss of trust from consumers. - The use of Information-Display-Matrix (IDM) has some limitations in terms of variables used.

Table 1 (Continued)

4	Hellberg-Bahr and Spiller (2012)	How to treat farmers fairly? Results of a farmer survey	- Around 40% of farmers taken part in the survey argued they didn’t feel treated fairly. - Key success factors influencing fairness in supply chain are: price satisfaction, reliability, and relationship quality. - Reliability and relationship quality are more important than price to achieve fairness from farmers' perspective.	- The study was conducted in Germany, which is not a representation of farmers, especially those in developing countries. - The study focuses on only one production sector and did not cover the entire supply chain. - Further studies into price volatility and its definition of fairness.
5	Schutter (2014)	Who's Got the Power? Tackling Imbalances in Agricultural Supply Chains	- The concentration of power among some stakeholders affects food producers badly. - Large buyers gain influence in governance patterns. - The abuses of power create unfair trade practices. - Agrifood liberalization creates price volatility for farmers.	- Need close government support. - Need huge investment over long period of time. - Require collaborative effort from all related entities.
6	Fałkowski et al. (2017)	Unfair trading practices in the food supply chain: A literature review on methodologies, impacts and regulatory aspects	- 96% of suppliers in EU food chain experienced at least one form of Unfair Trade Practices (UTPs). - Available data to analyze UTPs is imperfect as firms involved aren't willing to disclose sensitive information. - There is no consensus on the definition of being "unfair", and thus, any effort to leverage farmers' market power is not guaranteed to be efficient.	- A vital increase in agrifood supply chain transparency. - There is arguably little corroboration regarding the validity of the effect of UTPs and whether the anti-UTP regulation works as intended. - Lack of systematic evidence stems from problem of measurement and lack of systematic evidence regarding UTPs' impact.

Table 1 (Continued)

7	Blizkovsky and Berendes (2016)	Economics of Fairness Within the Food Supply Chain in Context of the EU	- Imbalance in market power means more powerful actors get to benefit more at the expense of farmers and small retailers. - Strong market power means the stronger and bigger the firm is, the more market power and tendency to take advantage of the market it can exert. Farmers suffer simply because there is lack of competition in the industry. - Consequently, producer share of the revenue has steadily been declining for decades, sparking inequalities in profit sharing among actors in the food supply chain.	- Although in the same region, producers in some countries do receive fair market share and profit distribution. - In depth study investigating correlations between price set by producers and legislative measurement needs to be conducted. - An investigative study incorporating non-EU regions would further validate the issues of UTPs.
8	Yoo and Won (2018)	A Study on the Transparent Price Tracing System in Supply Chain Management Based on Blockchain	- Give stakeholder easy access to retail price information. - Each transaction is valid and transparent. - Stakeholders can reduce operational costs.	- Unproven Concept. - Expensive to implement. - Require deep collaboration
9	(Fair World, 2018)	Fairness for Farmers: A Report Assessing the Fair Trade Movement and the Role of Certification	- Most food producers face marginalization. Fair trade helps them improve terms of trade with other actors in value chains. - Enforcing strong auditability and traceability help improve profitability and trust for food producers. - By setting up minimum prices and guaranteed equal treatment from other players, food producers' livelihood is drastically improved.	- Long term sustainability is still questionable. - Abuse of power is still present. - Need government intervention.
10	Segal and Le Nguyet (2019)	Unfair Harvest: The State of Rice in Asia	- Inequality in food value chains are worsening as some farmers receive 4% of total price. - Food producers lack the power to fairly negotiate with traders and other actors. - Climate change, unsustainable production, and increasing input costs pressure farmers to accept low incomes.	- Developing new, better regulated value chains. - Government support and subsidies. - Guaranteed fair treatment of food producers by other actors.

Table 2 Literature review of sustainability in agri-food supply chain

Nº	Authors (Year)	Title	Findings	Challenges
1	Bunte (2006)	Pricing and Performance in Agri-food Supply Chains	- There are abuse of power done on food suppliers. - Income risks are shifted to farmers. - There has been little study into sustainable agri-food supply chains.	- Creating a measurement for sustainable supply chains. - Leveraging food producers' roles and income in supply chains. - Fostering vertical and horizontal integration.
2	Thompson et al. (2007)	Agri-Food System Dynamics: Pathways to Sustainability in an Era of Uncertainty	- Majority of farm production do not have negative ecological impact in mind. - Traditional food producers are suffering from poverty and poor livelihood amidst technologically-driven farming system.	- Invoke the use of sustainability production requires government support. - Multinational corporations have to cooperate fairly with food producers.
3	Fritz and Matopoulos (2008)	Sustainability in the agri-food industry: a literature review and overview of current trends	- Consumer trends, and globalization have shaped unsustainability business practices. - Concentration of sector has influenced retailers to enforce unfair practices on food producers to retain profitability. - Most firms focus only on environmental and social aspects, while neglecting economic factor.	- Changing consumption pattern of customer requires massive collaboration from governments and corporations. - Huge investment in training employees and communities about sustainability.
4	Seuring and Müller (2008)	Core Issues in Sustainable Supply Chain Management – a Delphi Study	- The integration of sustainability in supply chain needs to be studied case by case. - There are four dimensions of which a sustainable supply chain discussion can be made: pressures and incentives, measuring impacts, supplier management, and supply chain management. - Pro-active companies are seen as the ones developing sustainable products and supply chains.	- Delphi study relies on open questions, developed into a set of issues. - Contrasting opinions between practitioners and researchers make the study somewhat inconclusive.

Table 2 (Continued)

5	Pouliot and Sumner (2008)	Traceability, Liability, and Incentives for Food Safety and Quality	- Improved food safety encourages consumers to pay more, thereby increasing profitability for food producers. - Traceability allows for potential increase in vertical integration and coordination.	- Traceability does not offer huge incentives to food producers. -Requires extensive information exchange.
6	Albino, Balice, and Dangelico (2009)	Environmental Strategies and Green Product Development: An Overview on Sustainability-Driven Companies	- The integration of Triple Bottom Line (TBL) and Supply Chain Management boosts the sustainability performance of the firm. - Enhanced communication based on long-term collaboration helps reduce costs and time. - Obtaining standards and certifications can drastically improve the quality and consistency of end products.	- The lack of government support and subsidies across many developed regions pose as a slowdown for innovation. - Most companies cannot afford to gamble with innovation and untested market opportunities. - Competitive pricing limits the scope of firm's innovation and responsibility. - TBL model focuses more on economic aspect rather than ecological responsibility.
7	Dangelico and Pujari (2010)	Mainstreaming Green Product Innovation: Why and How Companies Integrate Environmental Sustainability	- Companies go green to comply with regulations, but it also allows for risk minimization, revenue, and image protection. - Green product designs are influenced by: market growth, increasing profit, brand image, and ecological responsibility.	- Integrate environmental and conventional product attributes. - High development and manufacturing costs lead to highly competitive price. - Lack of customers' awareness of the benefits and price sensitivity.
8	Lai and Wong (2012)	Green Logistics Management And Performance: Some Empirical Evidence from Chinese Manufacturing Exporters	- Green Logistics Management (GLM) is characterized into four attributes: procedure-based, evaluation-based, partner-based, and general environment management practices. - Study supports environmental management or GLM does in fact facilitates environmental and operational performance in recycling, reducing carbon footprints, product quality, and shorter lead time.	- To verify the effectiveness of attributes, a long-term study should be made using consistent data over a period of time. - More credible insight and data could be extracted from specific fields of manufacturing, including telecommunication equipment and consumer electronics. - More research could be done focusing on

Table 2 (Continued)

			- Pressure from customers and governments compel producers to comply to environmentally friendly regulations.	how the ratio of both import and export sales volume could relate to environmental pressure and manufacturers' effort in implementing GLM.
9	Tseng, Tan, and Siriban-Manalang (2013)	Sustainable Consumption and Production For Asia: Sustainability Through Green Design and Practice	- Utilizing new policies to enable for a greener Public Private Partnerships (PPP). - Combining Green and Lean Supply Chain to help achieve true Triple Bottom Line (TBL) in accordance to the emergence of Green Consumerism. - Investment in technology for optimization and innovation of green supply chain design helps boost its deployment across various segments of industries.	- The shift to renewable energy and green supply chain is extremely challenging, especially in developing countries. - Call for collaboration from various stakeholders in every industry, including government bodies.
10	Hendrickson et al. (2017)	Power, Food and Agriculture: Implications for Farmers, Consumers and Communities	- Constrained choices force farmers into choosing predetermined choices of inputs. - Industrialized farming increase income inequality among stakeholders. - Environment has been severely affected by large-scale farming with no sustainable practices put into place.	- Establishing food sustainability policy worldwide. - Subsidize food producers using agro-ecological means.

Table 3 Literature review of existing blockchain solutions in supply chain

Nº	Authors (Year of publication)	Titles	Supply chain field
1	Cocco et al. (2021)	A Blockchain-Based Traceability System in Agri-Food SME: Case Study of a Traditional Bakery	Traceability
2	Reddy et al. (2021)	FarmersChain: A Decentralized Farmer Centric Supply Chain Management System Using Blockchain and IoT	Traceability and transparency
3	Adamashvili et al. (2021)	Blockchain-Based Wine Supply Chain for the Industry Advancement	Traceability
4	Chiranjeevi et al. (2021)	Block chain Technology in Agriculture Product Supply Chain	Traceability
5	Lau et al. (2021)	Blockchain-based Supply Chain System for Traceability, Regulation and Anti-counterfeiting	Traceability
6	Sund and Lööf (2019)	Performance Evaluation of a Blockchain-based Traceability System – A Case Study at IKEA	Traceability
7	Sathya et al. (2021)	Blockchain Technology for Food supply chains	Decentralization and security
8	de Carvalho et al. (2022)	Blockchain-Enabled supply chains: An application in fresh-cut flowers	Transparency and immutability
9	Vo et al. (2021)	Building Sustainable Food Supply Chain Management System Based on Hyperledger Fabric Blockchain	Sustainability and traceability

Table 3 (Continued)

10	Surjandari et al. (2021)	Designing a Permissioned Blockchain Network for the Halal Industry using Hyperledger Fabric with multiple channels and the raft consensus mechanism	Decentralization and transparency
11	Sunny et al. (2020)	Supply chain transparency through blockchain-based traceability: An overview with demonstration	Traceability and transparency
12	Di Vaio and Varriale (2020)	Blockchain technology in supply chain management for sustainable performance: Evidence from the airport industry	Sustainable performance
13	Sternberg, Hofmann, and Roeck (2021)	The Struggle is Real: Insights from a Supply Chain Blockchain Case	Supply chain management
14	Azzi, Chamoun, and Sokhn (2019)	The power of a blockchain-based supply chain	Traceability and IoT in supply chain
15	Elbashbishy, Ali, and El-adaway (2022)	Role of Transactional Blockchain in Facilitating Procurement in International Construction Projects	Supply chain procurement

As seen above, most of the researches reviewed pertain to mostly frameworks and concepts, with some applications that have not either been implemented or too complicated to be feasibly applied. Therefore, this study could very well be among the first of its kind to explore the blockchain network designs testing as well as the analysis of said network's scalability in literature. It also seeks to use blockchain technology and smart contract to remedy the issues of scalability and centralization clearly observed and documented in Supply Chain literature. We have learned that supply chain, particularly food supply chain is delineated with centralization of power, with big buyers such as retail chains, distributors, and such abuse their purchasing power to force producers and other stakeholders to accept unfair terms and business deals. We have also learned that traceability in supply chain is another big issue where a lot of products are not properly recorded along the supply chain. Technology adoption is another concern where producers and some other manufacturers are left without technological support, putting them into an unfair position comparing to competition. In addition, there have been some studies made on blockchain applications in supply chain. We have seen some studies focus on creating a blockchain network that establish and retain decentralization where abuse of power is reduced since network participants can access the same shared information. Not only do marginalized stakeholders are given the chance to compete fairly, but they also can access a more transparent network where business conducts can be better negotiated, products are better traced, and responsibilities can be evidently imposed on deserving parties.

On the other hand, this research is particularly pertained to experimenting on various configuration of blockchain network that is based on Hyperledger Fabric. The experimentation makes use of multiple network configurations in order to study the network performance, flexibility, and stability in manners where network members are gradually increased. The experiment is intended this way to concentrate on performance, particularly the average network transaction time. Due to the decentralization nature of the network where there is no central authority to create and share the information, the blockchain network has to make use of Peer-to-Peer (P2P) concept where members communicate in order to sync information. The next chapters will showcase the concept behind blockchain and smart contract, as well as the network topology behind the experimentation. The detailed specification of hardware and software is presented, along with Hyperledger Fabric and its components—the most vital part of this research. In chapter 4, the experimentation workflow is shown in details as to how the network is created step by step, as well as how each network configuration is tested, and how each of the network transaction speed is obtained. The explanation is provided behind what each figure means, and how it affects the network performance. Thus, the experimentation and analysis of this research on blockchain network scalability and flexibility serve as a starting point on which the framework can be studied and leveraged into any future form of practical application should there be any further studies.

CHAPTER 3
METHODOLOGY

This research methodology is to design a blockchain-based smart contract platform aiming to leverage the use of technology in a simplified supply chain network. It also incorporates network scalability, flexibility, as well as network performance analysis.

Blockchain and Hyperledger Fabric

Blockchain has emerged as the core technology to power Bitcoin, the first and, currently, the biggest cryptocurrency of its kind due to its “immutability, decentralization, and time-stamped record keeping” (Gausdal, Czachorowski, & Solesvik, 2018, p. 01), and its “integrity, resilience, and transparency” (Viriyasitavat & Hoonsopon, 2018, p. 01). First mentioned in pseudonymous author Satoshi Nakamoto’s well-known white paper titled: “Bitcoin: A Peer-to-Peer Electronic Cash System” (Nakamoto, 2008, p. 02), Blockchain has been at the core of Bitcoin’s innovation as it delivers “a trustless proof mechanism of all the transactions on the network, as well as existing “as the architecture for a new system of decentralized trustless transactions...” (Swan, 2015, p. X).

Shortly after the release of Bitcoin as an open source software in 2009, the focus was on Blockchain because of its unique solution to the double-spending problem, by verifying all transaction logs and its publication’s validity via cryptography hashes using Nakamoto’s Consensus (Clark, Edward, & Felten, 2015, pp. 106-107), and its introduction of a trustless decentralized system (Marr, 2018). According to various publications, Blockchain is given slightly different definitions. For instance, Blockchain is regarded as “a distributed, transactional database. Globally distributed nodes are linked by a Peer-to-Peer (P2P) communication network with its own layer of protocol messages for node communication and peer discovery” (Glaser, 2017, p. 1545), or “a public ledger and all committed transactions are stored in a list of blocks” (Zheng, Xie, Dai, Chen, & Wang, 2017, p. 557). In other publications, however, Blockchain is defined in a more technical manner, focusing on its decentralization and peer-to-peer validation via time-stamped ledger (Aste, Tasca,

& Di Matteo, 2017, p. 19; Francisco & Swanson, 2018, p. 02; Hawlitschek, Notheisen, & Teubner, 2018, p. 52; Seebacher & Schüritz, 2017, p. 14), a trustless approach of data system management and transparency (Bano et al., 2017, p. 01; Tribis, El Bouchti, & Bouayad, 2018, p. 01; Yli-Huumo, Ko, Choi, Park, & Smolander, 2016, p. 02), security (Cai et al., 2018, p. 02; Korpela, Hallikas, & Dahlberg, 2017, p. 4187; Li, Jiang, Chen, Luo, & Wen, 2017, p. 07; Watanabe et al., 2016, pp. 01-02), and the blockchain framework itself (Risius & Spohrer, 2017, p. 07).

In this research, we focus on a simpler and more basic approach meaning of Blockchain as *"decentralized distributed network with a shared ledger that is tamper-proof, time-stamped, encrypted, and nodes-verified to ensure security, scalability, and transparency."* Since its inception in 2008, Blockchain was made open-source compatible in 2009 following its deployment alongside Bitcoin. According to Swan (2015, p. IX), Blockchain is an extremely disruptive technology that "…could have the capacity for reconfiguring all aspects of society and its operations." Thus, its revolutions are categorized into 3 phases: Blockchain 1.0, Blockchain 2.0, and Blockchain 3.0. Blockchain 1.0 is the implementation of the cryptocurrencies via peer-to-peer digital payment systems; whereas Blockchain 2.0 is the extension of the technology, focusing on contracts and application that extend the usage of Blockchain into economic and business practices: stocks, bonds, loans, as well as smart contracts. Blockchain 3.0, on the other hand, goes beyond business, finance, and markets—to focus on government, health, literature, science, and art (G. Chen, Xu, Lu, & Chen, 2018, p. 02).

This research makes use of Hyperledger Fabric (Hyperledger, 2023), a state-of-the-art permissioned Blockchain Platform that offers versatility and modularity, as well as privacy and security for enterprise use. Hyperledger Project was founded by Linux Foundation in 2016 ("Blockchain Quick Reference by Brenn Hill, Samanyu Chopra, Paul Valencourt," 2023) to foster blockchain development in enterprise usage by preserving security, modularity, and privacy using Decentralized Ledger Technology (DLT), also known as Blockchain. Hyperledger Fabric is a sub-project which is built with collaboration from 30 founding members such as Digital Asset, Blockstream, and IBM.

Hyperledger Fabric prioritizes privacy and security by using modular components such as pluggable consensus model, multiple ledger formats, and Membership Service Provider (MSP). This allows Hyperledger Fabric to support a varying degree of networks in different industries such as in Supply Chain. In addition, members of Hyperledger Fabric network can be grouped into different channels where confidential information can be shared—with customization as to which members can view or participate with precision. For instance, some members might be competitors in the same field, and thus, they can be grouped into different channels where transactions and other information is not shared. As a result, only channel members can access certain information, and only those members possess the ledger copies containing the aforementioned information. Below are more key advantages of Hyperledger Fabric:

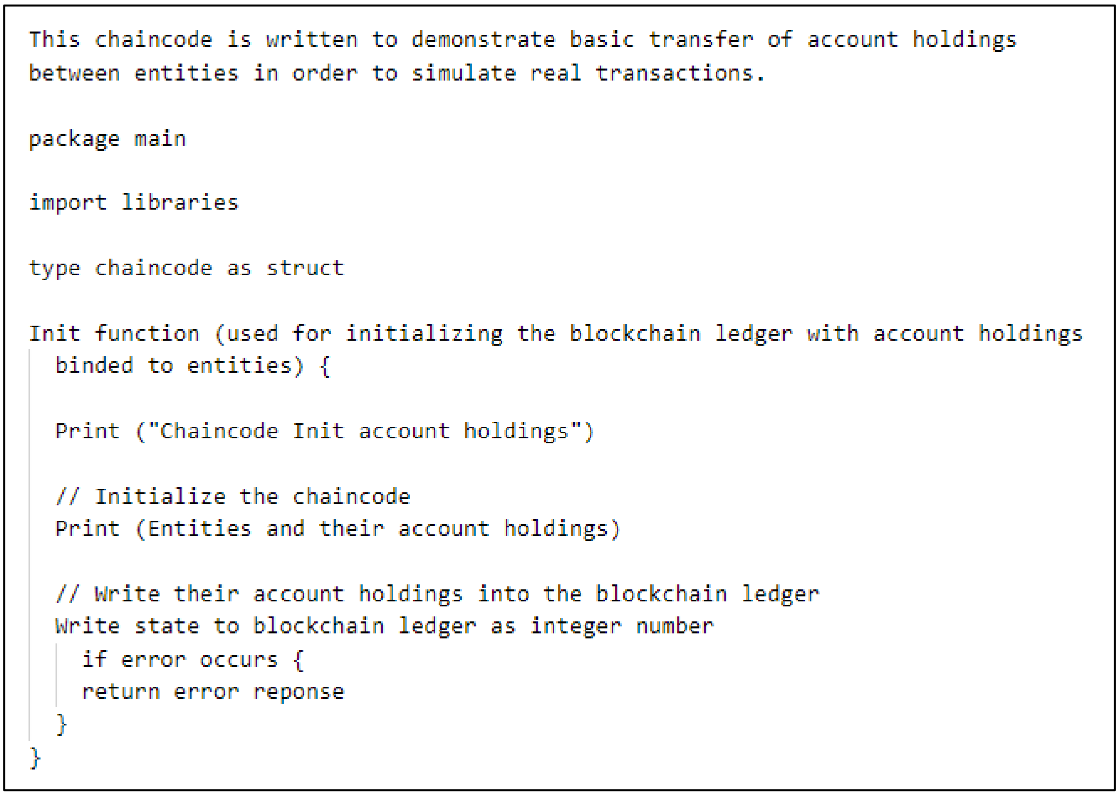
1. **Permissioned Architecture:** Only permissioned members are allowed access to the network.
2. **Modularity:** As stated before, components such as Certificate Authority and Consensus can be replaced with desired supported alternatives.
3. **Consensus:** Default consensus is RAFT, which is scalable and reliable; however, members can develop their own consensus model suited for specific use cases.
4. **Flexible data sharing approach:** Members can isolate vital and confidential data using channels or use private data collections.
5. **Easy to implement and govern smart contract model:** With multi-language support for smart contract (Go, Java, JavaScript), members can utilize existing resource to establish multiple smart contracts within the same network, while chaincode versioning support helps with maintaining and/or upgrading those smart contracts for future business logics.
6. **Customizable Endorsement Policy:** members can vote for a varying degree of rights and duties for all participants. Some members may be elected to endorse specific transactions, while others are needed to participate in certain network configuration updates.
7. **Rich database support:** Members can vote to select between key-value pair queries or JSON queries.

8. **Decentralized ledgers:** each peer node can store its own copy of ledgers across multiple channels depending on the network configuration. One peer node can also store multiple instances of chaincode.

Smart contract and functions

The basic idea behind smart contract was explored more than twenty years ago by Szabo (1997). It is essentially a form of autonomous digital software made to emulate contracts through the blockchain architecture and to also prevent any fraudulent alteration to the data (Lauslahti, Mattila, & Seppala, 2017, p. 11). According to Savelyev (2017, p. 05), smart contract is “an agreement whose performance is automated”; whereas Greenspan (2016) defines it as “a piece of code which is stored on an Blockchain, triggered by Blockchain transactions, and which reads and writes data in that Blockchain’s database.” Another definition sees smart contracts as “automated software program built on a blockchain protocol” and as “programmable contractual tools, they are contracts embedded in software code. Thus, a smart contract can include the contractual arrangement itself, governance of the preconditions necessary for the contractual obligations to take place and the actual execution of the contract.” (Koulu, 2016, p. 53). However, One of the more concrete and complete definition is: “Smart contracts are digital contracts allowing terms contingent on decentralized consensus that are tamper-proof and typically self-enforcing through automated execution” (Cong & He, 2019, pp. 1764-1765).

Smart contracts are based on code, and therefore, are immediate and can be securely executed without third party interventions like banks or courts. It has also been heralded as the next revolution in global business. (Levy, 2017, p. 02) As a consequence, it helps increase trust and transparency in a public or private blockchain since everyone is allowed to check the codes underlying behind the contracts themselves (Gatteschi, Lamberti, Demartini, Pranteda, & Santamaría, 2018, p. 05). Additionally, smart contract excels at managing heavy data-driven scenarios. It can efficiently and effectively automate transactions and other contractually-agreed terms despite the complexity and will always produce accurate result (Christidis & Devetsikiotis, 2016, pp. 2296-2297).



```
This chaincode is written to demonstrate basic transfer of account holdings
between entities in order to simulate real transactions.

package main

import libraries

type chaincode as struct

Init function (used for initializing the blockchain ledger with account holdings
  binded to entities) {

  Print ("Chaincode Init account holdings")

  // Initialize the chaincode
  Print (Entities and their account holdings)

  // Write their account holdings into the blockchain ledger
  Write state to blockchain ledger as integer number
    if error occurs {
    return error reponse
  }
}
```

Figure 1 A snippet of simple smart contract

The author has written a smart contract in Go for use in this blockchain network. The smart contract is written as a basic contract meant to establish entities, or organizations in this case, with account holdings so that we are able to transfer said holdings from one entity to another. It is also meant to simulate transactions between organizations. Doing so enables the researcher to measure the stability and scalability of the network, as well as the transparency and other benefits for supply chain. There are two necessary functions that are used for experimentation on this blockchain network:

1. Init: This function is used to populate the blockchain ledger with initial information about entities and their account holdings in integer number.

2. Invoke: This function is meant for transferring account holdings from one entity to another. Doing so will change the state of the blockchain ledger and allow the researcher to verify its data accuracy and immutability.

Blockchain network architecture

Hyperledger Fabric version 2.5 has been chosen as it is the latest available version during this research. Since the author uses this network only for local experimentation, some components such as Certificate Authority (CA) are not used because they would not otherwise have any impact on performance nor stability of the blockchain network. Specifically, Certificate Authority is used in production environment where blockchain networks, similar to the configurations used in this study, rely on security and authorization for every member and user before any network access is granted. CA allows organizations to issue certificates and keys to users based on their needs. For instance, client users such as customers are allowed certain actions such as issuing transactions and checking their remaining balance. Whereas users like organization administrators are permitted to access and initialize configuration changes within the network. Components such as CA is unnecessary since we are dealing with a preconfigured network in a local testing environment where security is not a concern at all. Moreover, custom ledger solution like CouchDB is not used in this experiment since it would have been a further bottleneck to the network performance as it requires roughly double the hardware performance in order to emulate. CouchDB commands extra docker containers to run in parallel to the peer nodes, with each peer node connecting to one CouchDB instance to store one ledger. This would effectively cut the network scalability in half as we continue to expand the network for experiments. To combat this issue, we use the default ledger database, LevelDB, since it does not require additional resource on the system and affect the network performance in any way. It is also fitting for this experiment because we are using a standard key-value pair custom chaincode that performs well with LevelDB. CouchDB, on the other hand, is only required if the network requires storing ledgers with complex asset properties and query requests.

Below are necessary components used for this blockchain network (Adhav, 2020; devendrasalunke, 2022; Krishnan, 2020; Maheshwari, 2018):

- **Peer nodes:** Arguable the most important component of any Hyperledger Fabric network. Peer is responsible for managing ledgers, smart contracts, as well as participating in various activities within the network. It may access the data in ledgers in order to execute smart contracts, or it may endorse transactions prior to committing any. There are many types of peers such as: endorsing peer, committing peer, anchor peer, leader peers, and orderer peers.

- **Organizations:** They own and control peer nodes within their own respective organizations. Organizations together form a blockchain network that abide by their own rules and conducts. These organizations also manage identities for peers and users, as well as authenticating all participants on the same network to avoid unauthorized access.

- Membership Service Provider (MSP): MSP manages identity and authentication of said identities within the network. This separates Hyperledger Fabric from the rest of the blockchain platform as each and every member and user of the network is deliberately recognized and authorized to participate in network activities. Participants are like client applications, users, peers and such are identified using certificates and cryptographic materials like private and public keys. The author utilizes local MSP for this setup as it is ideal for development and testing purposes of identities within the local network. Cryptogen tool is used to generate certificates and keys for this study as it is ideal for local testing.

- **Orderer:** It is a vital component in specifying the delivery of transactions to peer nodes for validation and endorsement processes before any transaction can be committed and appended into the blockchain ledger. Orderer maintains and sorts transactions, as well as communicating necessary information to peer nodes in order to reach consensus. Orderer also acts as the mediator for communicating between peer nodes by implementing the consensus protocol to order the transactions. Orderer nodes form an ordering service withing a certain blockchain network. The consensus protocol utilized in this research is RAFT protocol (Ongaro & Ousterhout, 2014, 2015). Raft provides accurate, secured, and low-latency consensus in a scalable distributed environment—which provides massive benefits for a supply chain network, whose members are constantly cooperate in dynamic environments of communication and data sharing.

- **Channel:** a private “subnet” of information sharing and communication for specific network members. Channels are defined by organizations, ledgers they share, ordering nodes, and smart contracts. Peers join channels using their certificates authorized by MSP.

- **Smart Contract:** Also known as Chaincode, smart contracts within a blockchain network carry out a series of executable business logic to be stored on the shared ledger. Chaincode helps define objects and assets for logic use that enables blockchain data to be written into the ledgers. Smart Contract also enables peers to simulate transactions before packaging them up into transaction proposals.

- **Ledger:** It is a digital book or journal of sort containing both factual transaction history and current value of business objects' attributes.

After installing the required base software layer, we can start installing Hyperledger Fabric and the sample images (Fabric, 2023d). For this, we have to clone Hyperledger Fabric samples repository, download latest Hyperledger Fabric Docker images, as well as Fabric CLI tool with configuration files. The following executables are used to create the Hyperledger Fabric blockchain network seen in this research:

- **Cryptogen:** It is a utility used in generating Hyperledger Fabric key materials. It is meant to be used for testing purposes.

- **Configtxgen:** It helps users create and inspect configurations of channels and their artifacts such as genesis block and configuration update files.

- **Configtxlator:** It allows for encoding and decoding between JSON and Protobuf versions of Hyperledger Fabric data structure, and to also create configuration updates.

- **Orderer:** It promotes transaction finality received from peers. Ordering service, which is comprised of ordering nodes, also helps eliminate bottlenecks and loss of performance as the network scales up.

- **Peer:** Administrators can start a peer node process or check its status. Peer also provides the point for access and management to channels.

- **Osnadmin:** Allows administrators to perform operations related to channel on an orderer. The activities can be: joining a channel, listing channels of which an orderer is a member, and removing it from a channel.

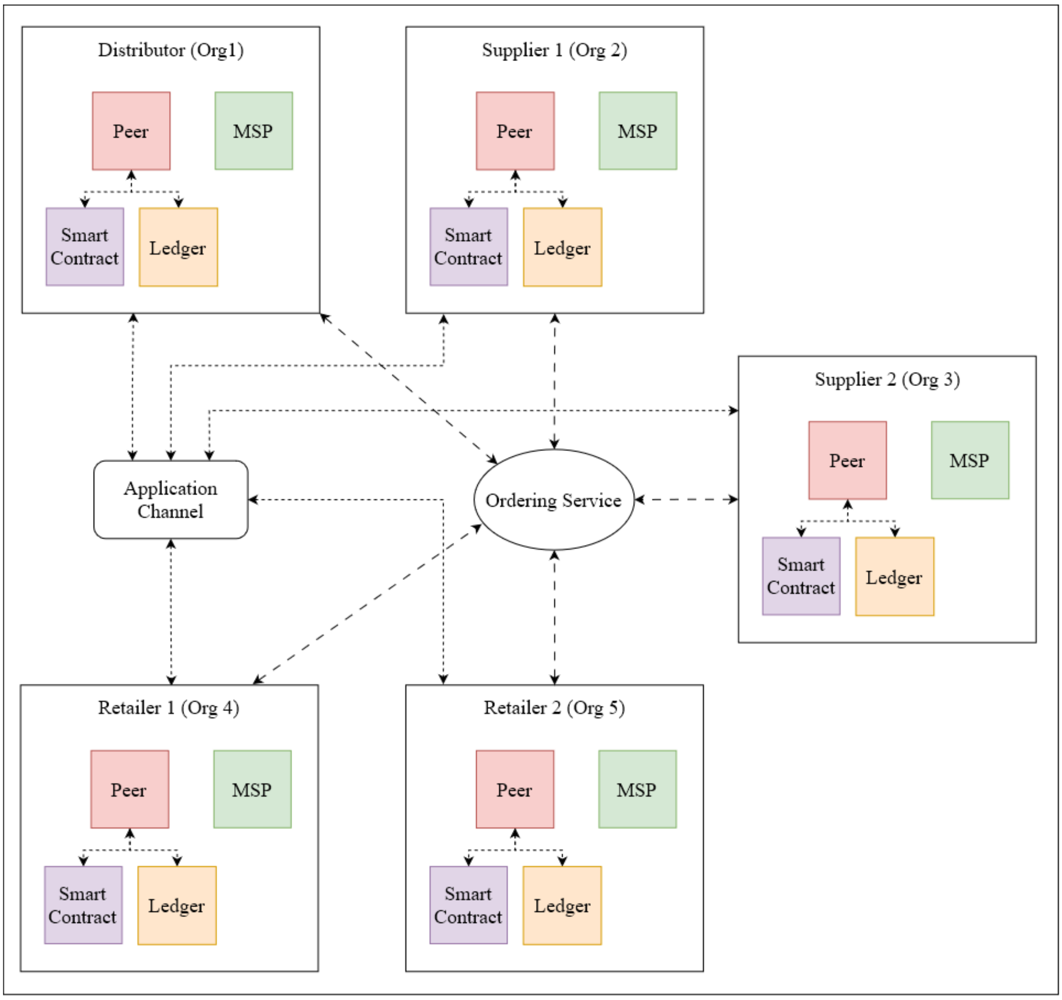


Figure 2 Hyperledger Fabric Network Topology

The figure 2 depicts the network topology used in this research. In this study, there are five actors or network members within our blockchain network:

- Distributor: The middleman who establishes and maintain the blockchain network.
- Supplier 1: A participant of the network.
- Supplier 2: A participant of the network.
- Retailer 1: A participant of the network.
- Retailer 2: A participant of the network.

Supplier 1, Supplier 2, Retailer 1, and Retailer 2 are participants of the network, who also host their own peer nodes that link to their own ledgers as well. Together, these 5 organizations form a blockchain network with an application channel for secured and private interactions. Each organization has their own MSP to manage and authenticate peers and users. Any unauthorized entry or attack is unlikely to happen as long as each organization has the capability to verify and control each member via certificates and keys. As seen in figure 2, This Hyperledger Fabric blockchain network has 5 members, all of which host their own instances of peer nodes, smart contracts, and blockchain ledgers. Each peer instance is holding an identical copy of the blockchain ledger. An ordering service comprising of one ordering node will be started in order to facilitate the transactions that will be conducted later. Also, Smart contract used in this research is named "simple" contract, whose role is to establish a mechanism for which peers can invoke transaction from one entity to another as defined by the logic within the smart contract itself. As explained in the preceding subsection of this chapter, the smart contract is meant to establish entities within the blockchain network, combined with functions that allow for initializing their account holdings, and another for sending a predetermined amount of said account holdings from one entity to another. The smart contract can be installed on any peer within the network, provided that specific peer needs to utilize any function within the smart contract.

In addition, peer ledger or blockchain ledger stores information relating to all transactions, including configuration updates in the blockchain network itself. In the event that one peer is temporarily offline, it will receive the up-to-date blocks containing transactions as well as configuration updates later on when it goes back online within an allowed period of time. The way this works is that all peers have their own blockchain ledger of which blocks are distributed from either orderer or other peers based on whichever can be done quicker. This ledger contains blocks linked together using hashes, providing almost impenetrable security where data tampering is almost impossible due to its many layers of security and distributed nature. Peers also host their own ledgers alongside the smart contract if required. Besides, the ledger that each peer hosts contains identical information about the business logic and its facts. This means that each blockchain ledger that is being hosted by each peer contains identical information that has been copied and distributed to each other. This information is the current world state of the transactions and facts about any assets and entities recorded. The facts and current world state are changed due to subsequent transactions made. The blockchain ledger, or ledger in short, cannot be changed or tampered with in any way; instead, it can only be appended—meaning that it can only be added, and it cannot be changed retroactively.

Moving on, for the orderer, since there are few transactions conducted in each experiment of this research, there is no real advantage or impact with the number of orderer nodes used—as such, only one orderer node is used. Moreover, there is an application channel named "testchannel" created to connect all five organizations and one orderer node together. This testchannel enables all members to participate and access confidential information such as transaction history or channel configuration updates. All peer, smart contract, ledger instances and one orderer node are hosted inside docker containers to ensure security and complete isolation from the operating system—and preventing data tampering.

On top of this, the researcher also made use of some specific preparations for the experiment. For instance, in order to maximize the local hardware resources needed, the local computing unit is configured to run Windows 11 with Ubuntu 22.04 LTS as the virtual machine residing inside this host Windows Operating System (OS). The Ubuntu OS has to install certain dependencies such as docker desktop, whose version is at least 4.18.0. Since docker desktop runs on another layer of virtual machine inside the Ubuntu OS, the resources are configured as following: CPU is set at 16, Memory is set at 7.8GB, Swap is set at 1GB, Virtual disk limit is set at 72GB.

As a notice, Docker Desktop can be downloaded from: "https://docs.docker.com/desktop/install/ubuntu/", and it has to be configured in a certain way.

1. Firstly, set up a docker repository within the Ubuntu OS using terminal. In a terminal window, type "sudo apt-get update" and "sudo apt-get update install ca-certificates curl gnupg".
2. Next, add the official key: "sudo install -m 0755 -d /etc/apt/keyrings",
3. then "curl -fsSL https:// //download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg".
4. After this, type: "sudo chmod a+r /etc/apt/keyrings/docker.gpg".

5. Now, the following command is used: “echo \ "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \ "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \ sudo tee /etc/apt/sources.list.d/docker.list > /dev/null” in order to setup repository for docker desktop.

6. Finally, the downloaded deb package can be installed using: “sudo apt-get update / sudo apt-get install ./docker-desktop-4.18.0-amd64.deb”.

Also, in order to use Hyperledger Fabric, the researcher has to setup the dependencies as following:

- Git is installed using the command: “sudo apt-get install git”
- cURL is setup by using: “sudo apt-get install curl”
- Docker is installed by: “sudo apt-get -y install docker-compose”, then installation is verified using: “docker –version” and “docker-compose –version”. Docker is also turned on by default using command: “sudo systemctl start docker” and “sudo usermod -a -G docker <username>”.
- Go is installed by downloading the latest executable using “https://go.dev/doc/install”.

As an additional note, Hyperledger Fabric sample network containing the essential Fabric components for this experiment can be installed as shown below:

1. Find a desired directory in which Hyperledger Fabric executables and configuration files are saved, then open a terminal window inside said directory.

2. To get the Hyperledger Fabric install script, use: “curl -sSLO https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh && chmod +x install-fabric.sh”.

3. This gives the user a script from which Hyperledger Fabric components can be chosen to be installed.

4. Type: “./install-fabric.sh docker samples binary” in order to install Hyperledger Fabric docker images, sample configurations, and binary files for Hyperledger Fabric components. Components such as: cryptogen, configtxlator, configtxgen, orderer, peer, osnadmin, and more will be installed. This gives the complete setup of Hyperledger Fabric sample network. For the experimentation done in this study, the researcher also utilizes the same Hyperledger Fabric installation

method as shown, with the exception of configuration files—which are specially preconfigured for testing, alongside some scripts written to automate many processes of the network creation workflow as shown in Chapter 4 of this research.

CHAPTER 4
EXPERIMENT RESULT

This chapter showcases the detailed explanation for blockchain network testing, from how it is created, the various configurations deployed, and how the data is obtained and presented. Firstly, the author outlines the configuration details which the tests take place—including the hardware and software used, as well as the blockchain platform on which the experiment is based. The total number of experiments are also explained along with details for each one of them. Next, we see the experimentation workflow for blockchain networks based on Hyperledger Fabric such as this one, specifically as to how the network is constructed. Furthermore, the transaction mechanism is shown to explain how the distributed network handles such procedure with security as a priority. Then the experimentation result is shown, alongside how each data point is obtained and how its average transaction time is calculated—as well as pointing to the gradual rise in delay of network performance in relation to the increasing number of peer nodes. After this, an explanation is provided as to why the network consumes a lot of resources, and to why it cannot be scaled up past a certain threshold in this local testing environment. Finally, it is shown the reason why peers are taking up most of computing resources in order to communicate to each other to keep the network secured and synchronized.

Blockchain network configurations and workflow

Here the author outlines the configurations used in this study as network scalability is demonstrated. Hyperledger Fabric is designed with modularity as a priority as components can be swapped out as desired. It also features fast transaction speed, smart contract technology, as well as refined data sharing. The particular network configuration used in this research is conducted on a Windows PC running on Windows 11 Pro 22H2 with OS Build 22621.1702. The actual Hyperledger Fabric network is then run on an Ubuntu (version 22.04 LTS) virtual machine located within the Windows Operating System. The following are the specifications for said virtual machine:

- CPU: AMD Ryzen 7 4800HS (8 cores/16 threads) 35 Watt @ 3GHz
- RAM: 16GB DDR4 @ 3200MHz
- GPU: Nvidia GeForce RTX 2060 6GB
- Storage: 150GB PCIe Gen 3 SSD

There are also a number of prerequisite software and executables needed in order to run Hyperledger Fabric and create a blockchain network for this study. As per outlined in Fabric's official documentation (Fabric, 2023b), there are a number of base layer software and dependencies needed. Since the study utilizes Ubuntu as the operating system, the author uses the following applications:

- Git: Version 2.34.1
- cURL: Version 7.81.0
- Docker: Version 23.0.5
- Docker-Compose: Version 2.17.2
- Docker Desktop: Version 4.18.0
- Go: Version 1.20.2
- JQ: Version 1.6

For Hyperledger Fabric executables, the author uses the latest available Hyperledger Fabric binaries at the time of writing to run the network. The details are as following:

- Cryptogen: Version 2.4.9
- Configtxgen: Version 2.4.9
- Configtxlator: Version 2.4.9
- Peer: Version 2.4.9
- Orderer: Version 2.4.9

The network architecture and the general framework upon which this network is based can be found in the preceding chapter of this study. What the author intends to do with this framework is to demonstrate the blockchain network potential in supply chain, including its scalability and flexibility. The underlying assumption of these blockchain network configurations and experimentation is to test the limit upon which the local environment-based simplified blockchain network can achieve per given computing resources. For instance, the first experiment, consisting of only 2 organizations and 2 peer nodes with an orderer is meant to display the minimum requirement for a blockchain network to operate. This blockchain configuration can give readers an expected minimum average transaction time for a blockchain network. It is also a benchmark on which further testing is based. Furthermore, the second experiment is meant to represent another supply chain network containing a total of 10 peer nodes across 5 organizations. This test is the template from which further experiment in this research is based, partly because the rest of the experiment will maintain the organization number as 5 members. It is expected that the number of 10 peer nodes is easily handled by the available hardware, and that further experiments from experiment number 3 to 5, or from 20 peer nodes to 40 peer nodes will not affect the hardware capability to handle the transaction invocation, as well as the constant communication between peer nodes. However, experiment number 6 and 7 are believed to consume the maximum capability of allocated hardware and its resources as will be explained later in this chapter.

Besides, this blockchain network is tested and further analyzed by increasing the number of peer nodes with increments of 10 per experiment. Essentially, the network will be started out as having 2 organizations, and with each organization hosting only one peer node in order to establish the minimum requirement of running such a blockchain network based on Hyperledger Fabric. Next, a custom smart contract based on Go will be installed on both peer nodes. Afterwards, chaincode initialization will take place using a command from one of the two peer nodes. This initialization populates the blockchain ledger with information about certain entities and their account holdings in a form of integer number. Then one of the peer nodes will send out a transaction command, of which will be approved and appended into the blockchain ledger. Afterwards, another experiment will take place by which the network will scale up to 5 organizations, with 2 peer nodes inside each organization. The same custom chaincode will be installed, and the transaction will take place. Thereafter, we scale the network up by increasing the peer nodes to 10 nodes across all 5 organizations. Experiment number 3 will keep the number of organizations at 5, while increasing the peer nodes to 20. This means that each organization contains 4 peer nodes. Consequently, it creates a consistent and equal distribution of the number of peer nodes across all 5 organizations that ensure the testing is kept as neutral as possible, with the performance of the network being tracked at all times. Experiment

4 through 7 will maintain the number of organizations at 5 and increase the number of peer nodes equally across all of them, up until there are 12 peer nodes for each of the organization—totally 60 peer nodes at the end. The exact details of how all the steps are executed can be found below.

Table 4 Blockchain system configurations used in this research

Blockchain System Configurations Used in This Research							
Experiment Number	1	2	3	4	5	6	7
Number of Organizations	2	5	5	5	5	5	5
Number of Peer Nodes	2	10	20	30	40	50	60

The table 4 describes the number of experiments, and how each of them is structured. As explained before, we start from the minimum requirement of a blockchain network—which has 2 peer nodes and 2 organizations. Moving on to experiment 2, the network is started again, but with 5 organizations and 10 peer nodes equally distributed. Experiment 3 is started with 5 organizations and 30 peer nodes, while experiment 4, 5, 6, and finally 7 increase the peer nodes by increments of 10. The peer nodes are equally distributed among all 5 organizations. The workflow of this entire research is optimized to save time and prevent previous experiment’s interference and unwanted effect on network performance and stability. The standardized workflow for this research can be found in figure 3.

In step 1, we start by generating the crypto materials needed for network creation. To do this, we utilize Cryptogen tool to generate key materials such as certificates and keys for each organization. These materials are credentials that peer nodes and users utilize whenever they participate in any activities within the network. Any user or peer instances that attempt to execute any command on the network will not be granted permission—that is, without the specific certificates and keys known to the organizations and network. After this, in step 2, we can start deploying docker containers for orderer and organization peers. By using docker, we can guarantee complete system isolation with the containers from system interference. Not only does docker consumer little resource, but it also provides additional layer of security. For instance, docker desktop that is used to manage and monitor docker containers in this research runs on a virtual machine state, effectively cutting it off from the system's interference and potential security risks.

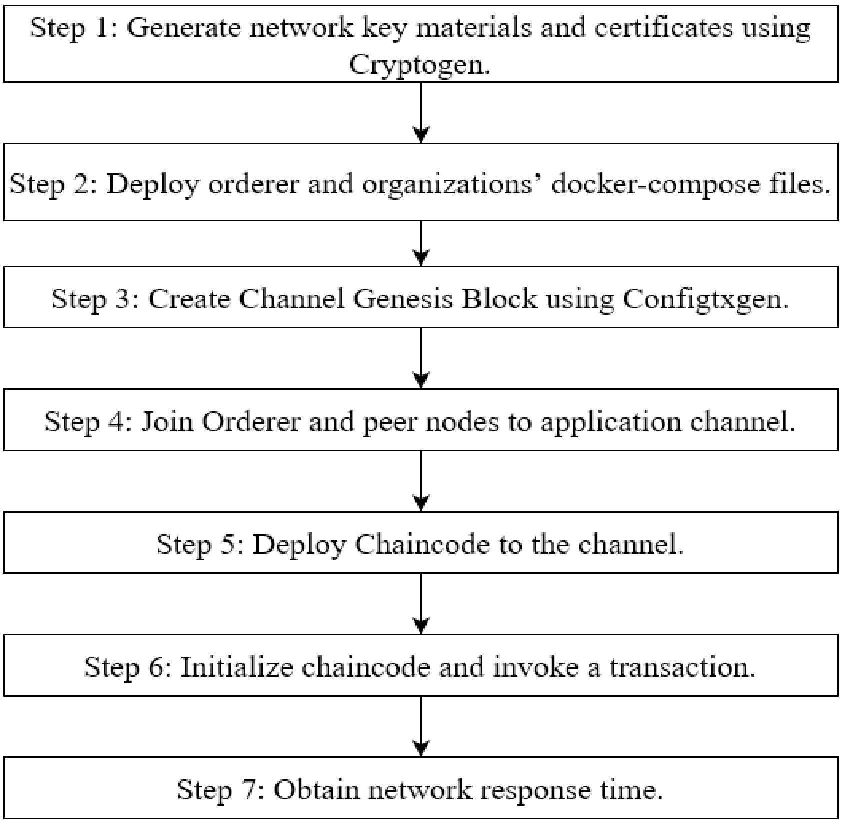

Step 1: Generate network key materials and certificates using Cryptogen.
Step 2: Deploy orderer and organizations' docker-compose files.
Step 3: Create Channel Genesis Block using Configtxgen.
Step 4: Join Orderer and peer nodes to application channel.
Step 5: Deploy Chaincode to the channel.
Step 6: Initialize chaincode and invoke a transaction.
Step 7: Obtain network response time.

Figure 3 Experimentation work flow

In step 3, Before a channel can be created, the administrator has to generate what is called "a genesis block". This genesis block is the primary method by which the network can be initialized, called "bootstrapping". Configtxgen tool is used in order to generate the block with information provided by a configuration file. This file contains details about each organization, including information about this channel and its policies as well. Next, in step 4, the orderer joins the network before peers, using osnadmin command tool. As explained by Fabric's documentation (Fabric, 2023a), the first orderer node that joins the channel is essentially activating the channel for peer nodes. Step 5 and 6 focus on chaincode packaging, installation, approving, and committing into the channel. These steps require administrator to setup proper channel and chaincode policies to allow for appropriate levels of participation from organization members, as well as their peer nodes. Lastly, after chaincode has been committed, we can initialize and invoke transactions. The invocation of smart contract allows the author to record and measure the network performance, all of which will be explained in detail in subsequent section of this chapter.

Experiment result

As mentioned in previous sections, Hyperledger Fabric version 2.5 was chosen for this research. Still, there are a number of processes to be complete prior to running Hyperledger fabric docker images with Fabric CLI tool binaries and create a blockchain network. After installing all the base layer software as outlined in previous chapter, we can start cloning the official Hyperledger Fabric samples configuration files, as well as platform-specific CLI tool binaries and configuration files. These configuration files and docker files enable us to design and test the various network configurations as desired. However, since Hyperledger Fabric is designed with pluggable components, it is recommended to design the network topology with selected components beforehand as it helps speed up the development and deployment processes.

The experiment procedure and configurations mentioned above are conducted one by one, each separated from the rest. After each experiment run has completed, the transaction time is measured using command feedback from the logs. Each transaction has to go through “transaction flow”, which is the mechanism behind asset exchange in Hyperledger Fabric. Transaction flow highlights the Execute-Order-Validate protocol of Fabric where achieving consensus is done through a process of voting. The protocol is explained below (Belchior, 2019):

- Execute: This process refers to when after blockchain client created a transaction proposal and sent it through to endorsing peers. The transaction contains the transaction info, payload, and transaction ID. The endorsement peers in this process simulate the transaction proposal against their own ledgers to make a read/write set, as well as checking the validity of the transaction itself. Endorsing peers then send the proposal response along with the transaction ID, read/write set, endorsers’ ID and their signatures back to blockchain client.

- Order: Blockchain client then sends this verified endorsed transaction received from all the endorsing peers to the orderer. The orderer will check whether the client has permission to propose a transaction in the channel. After this, Orderer makes a block of the endorsed transaction in an ordered manner for the channel. Orderer will then broadcasts this transaction to all peers inside the blockchain network.

- Validate: As a final check, each peer needs to validate this transaction by verifying it with endorsement policy configured beforehand. The read/write set is also checked for this transaction in the block. If anything doesn't match, the peer simply cancels the transaction. If everything proceeds without any error, the ledger is then updated with this new block, containing a transaction that passed all the checks.

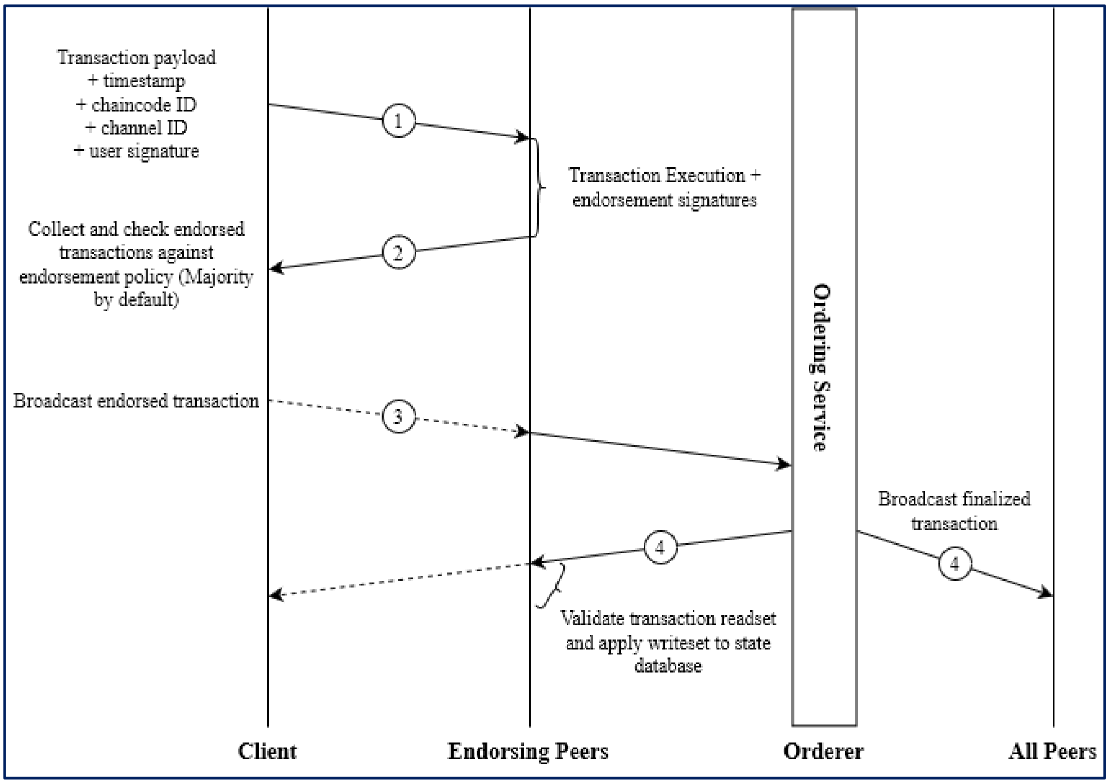

Transaction payload
+ timestamp
+ chaincode ID
+ channel ID
+ user signature
1
Transaction Execution + endorsement signatures
Collect and check endorsed transactions against endorsement policy (Majority by default)
2
Broadcast endorsed transaction
3
Ordering Service
Broadcast finalized transaction
4
4
Validate transaction readset and apply writeset to state database
Client
Endorsing Peers
Orderer
All Peers

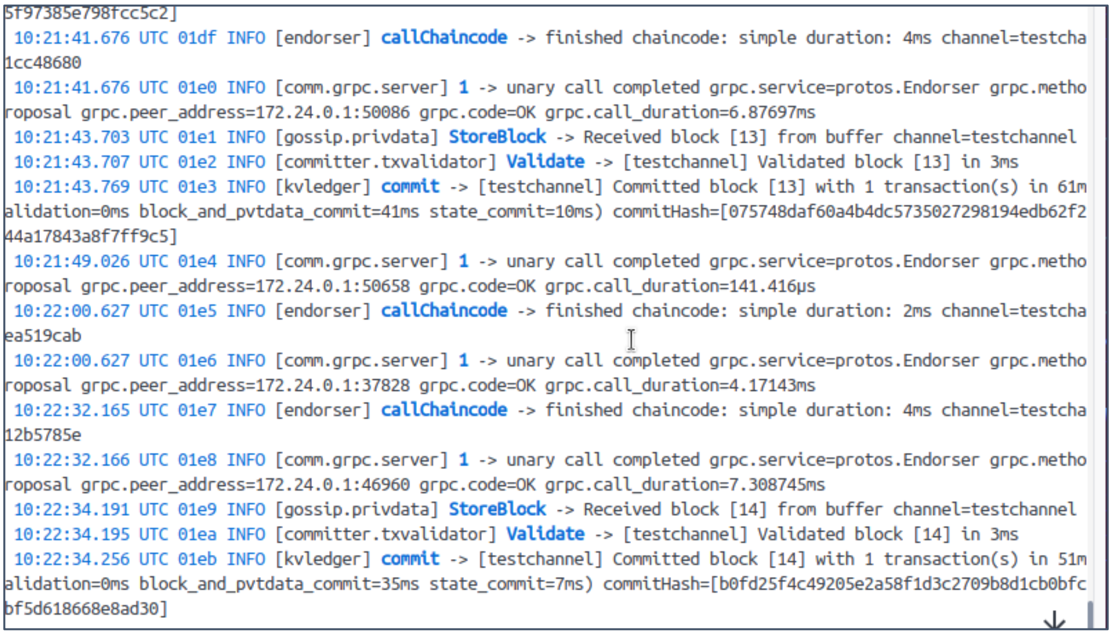
Figure 4 Transaction Mechanism

Figure 4 shows the mechanics behind every transaction taken place in this research. It utilizes Execute-Order-Validate protocol (Fabric, 2023e) used in Hyperledger Fabric Blockchain platforms. In step 1, the mechanism starts with client or submitter initiating the transaction with payload that is equipped with timestamp, channel ID, chaincode ID, as well the signature of submitter. This transaction proposal is then passed to endorsing peers who will then execute/simulate the transaction against their existing state database or blockchain ledger. Endorsing peers have to verify whether: transaction proposal is formed correctly, whether it has been submitted before to prevent double-spending, whether the transaction submitter's signature is valid, and whether the submitter is authorized to propose transactions in this channel in the first place.

In step 2, If it's valid and correct, endorsing peers will then endorse this transaction with their unique signatures and pass it back to the submitter, who has to check these signatures or approvals against the channel endorsement policy set by all peer members. During step 3, if enough approved votes are collected, this transaction is considered valid and endorsed, it will also be then broadcasted to the endorsing peers, who verify the consistency of transaction prior to step 2. Endorsing peers will then broadcast this transaction proposal as a transaction message to ordering service, containing read/write set, channel ID, and signatures from endorsing peers. In step 4, Ordering service orders the transaction and creates a block to send to all peers within the channel. Each peer validates the transaction again against their own blockchain ledger to make sure there is consistency to their respective existing ledger. Finally, each peer can approve the transaction as valid then append this block into their own ledger.

Moving on to the result, through a series of testing the author has accumulated groups of data points in order to establish the average duration for transactions based on the size of networks. The data helps create an exact picture of what has been achieved and how important they are to realize its mechanism alongside performance, as well as stability associated with such blockchain network. Moreover, these numbers also help cement the reliability and flexibility of the concept to be further used in future experiments—including stretching the limits of its practical scalability. The table below demonstrates the results of such nature as the author successful conducted a series of experiments focusing on the network scalability and transaction performance, with gradual increase on system load via peer numbers.

To begin, the table 4 contains the number of peers to be used in a series of isolated experiments. There are a total of 7 experiments all of which tested the network performance in terms of scalability, transactional speed, and network reliability. While 6 of them contains a total of 5 organizations as members inside a blockchain network, the first experiment only has 2 organizations. This is done to establish a baseline number of minimum requirement for a functional blockchain network so that we can see how the performance behaves over increasing numbers of peer nodes. Moreover, as mentioned multiple times in preceding sections, the network utilizes only one orderer node since there are few transactions needed to be invoked to test the performance; therefore, additional orderer nodes are considered redundant and unnecessary in this scenario. Furthermore, there are 5 peer nodes used in most of the experiments except the first one as it is the one meant for baseline observation. Peer 0 for each of the 5 orgs are used to measure the transaction speed every time a transaction is invoked. These peer 0s are essential to measure the speed since they are designated as endorsing peers (Andrade-Salinas, Salazar-Chacon, & Vintimilla, 2020, p. 98; Rilee, 2018).



```
5f97385e798fcc5c2]
 10:21:41.676 UTC 01df INFO [endorser] callChaincode -> finished chaincode: simple duration: 4ms channel=testcha
1cc48680
 10:21:41.676 UTC 01e0 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=protos.Endorser grpc.metho
roposal grpc.peer_address=172.24.0.1:50086 grpc.code=OK grpc.call_duration=6.87697ms
 10:21:43.703 UTC 01e1 INFO [gossip.privdata] StoreBlock -> Received block [13] from buffer channel=testchannel
 10:21:43.707 UTC 01e2 INFO [committer.txvalidator] Validate -> [testchannel] Validated block [13] in 3ms
 10:21:43.769 UTC 01e3 INFO [kvledger] commit -> [testchannel] Committed block [13] with 1 transaction(s) in 61m
alidation=0ms block_and_pvtdata_commit=41ms state_commit=10ms) commitHash=[075748daf60a4b4dc5735027298194edb62f2
44a17843a8f7ff9c5]
 10:21:49.026 UTC 01e4 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=protos.Endorser grpc.metho
roposal grpc.peer_address=172.24.0.1:50658 grpc.code=OK grpc.call_duration=141.416µs
 10:22:00.627 UTC 01e5 INFO [endorser] callChaincode -> finished chaincode: simple duration: 2ms channel=testcha
ea519cab
 10:22:00.627 UTC 01e6 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=protos.Endorser grpc.metho
roposal grpc.peer_address=172.24.0.1:37828 grpc.code=OK grpc.call_duration=4.17143ms
 10:22:32.165 UTC 01e7 INFO [endorser] callChaincode -> finished chaincode: simple duration: 4ms channel=testcha
12b5785e
 10:22:32.166 UTC 01e8 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=protos.Endorser grpc.metho
roposal grpc.peer_address=172.24.0.1:46960 grpc.code=OK grpc.call_duration=7.308745ms
 10:22:34.191 UTC 01e9 INFO [gossip.privdata] StoreBlock -> Received block [14] from buffer channel=testchannel
 10:22:34.195 UTC 01ea INFO [committer.txvalidator] Validate -> [testchannel] Validated block [14] in 3ms
 10:22:34.256 UTC 01eb INFO [kvledger] commit -> [testchannel] Committed block [14] with 1 transaction(s) in 51m
alidation=0ms block_and_pvtdata_commit=35ms state_commit=7ms) commitHash=[b0fd25f4c49205e2a58f1d3c2709b8d1cb0bfc
bf5d618668e8ad30]
```

Figure 5 Transaction log inside Docker Desktop

These endorsing peers are vital to the network since they practically simulate transactions against the chaincode, and then endorse those transactions through orderer to get it appended into the ledger. Because the author has preconfigured this series of blockchain network to use all 5 peer 0 from 5 organizations as endorsing peers, we are able to measure a more precise change in network performance and stability than a fewer number of endorsing peers in the network. In order to achieve a more uniform and easier way to understanding how the blockchain network behaves, the author has decided to add the average number of transaction speed as the deciding factor to look into the experiment itself. Through this, we are able to grasp a simpler and more definitive meaning to grading the transaction speed as the average number adds more clarity and consistency. To get transaction speed for each endorsing peer, the author conducted an experiment where a transaction is invoked using peer 0 from org 1. The transaction proposal is simulated against the chaincode and then passed to orderer and then back to all the peers. The duration between transaction endorsement and block commit, which is basically when the transaction is cemented as finalized and appended into the blockchain, is considered to be the time it takes to complete a block. Figure 5 shows an example of how each number is obtained.

Table 5 Blockchain experiment results

Peer names	**Number of Peers**						
	2 peers	**10 peers**	**20 peers**	**30 peers**	**40 peers**	**50 peers**	**60 peers**
Peer 0 Org 1	2039	2053	2087	2123	2175	2329	2341
Peer 0 Org 2	2039	2054	2086	2126	2161	2241	2337
Peer 0 Org 3	N/A	2050	2087	2126	2163	2190	2310
Peer 0 Org 4	N/A	2052	2081	2118	2170	2170	2328
Peer 0 Org 5	N/A	2050	2087	2115	2174	2242	2333
Average	2039	2052	2086	2122	2169	2234	2330
Results are based on duration between transaction endorsement to block commit							
All figures are in milliseconds (ms)							

From the figure 5, we know that there are steps by which the transaction must complete before it arrives to commit stage. Typically, it starts from endorsement as peers simulate transactions as seen at the time of 10:22:32.165. When it arrives at block commit exactly on 10:22:34.256, we can see the block is committed into the blockchain ledger. To get the transaction speed for this particular peer, we subtract the duration by which it takes from block endorsement to block commit which is 34.256 – 32.165, which results in 2.091 seconds. Next, in order to get an average number, each transaction speed on each particular experiment is added against the rest and then divided by the total number of endorsing peers. This same method is consistently used against all the other peers in the rest of experiments, creating a group of data by which the averages are calculated and show. These average numbers provide a clearer and more concise view into how the network behaves under different configurations—especially with network scalability and stability.

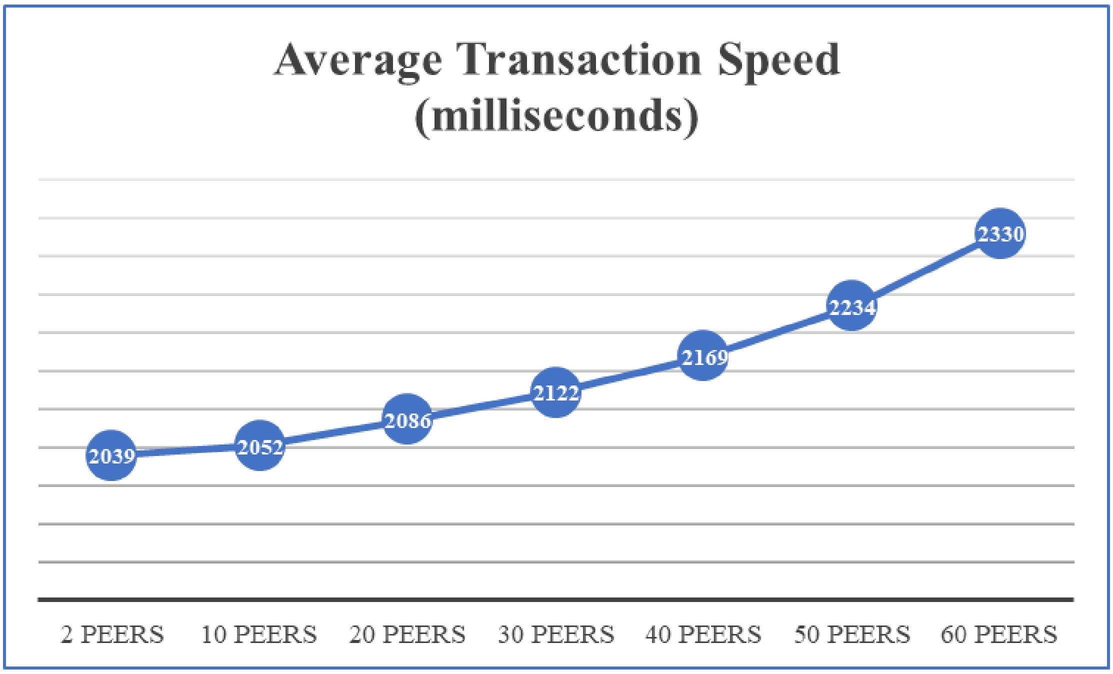


Figure 6 Average transaction speed (milliseconds*)*

For instance, in order to arrive at 2.3298 seconds as the average transaction speed with 60 peer nodes, the researcher added all 5 peer 0’s transaction speed and then divide it by 5 as shown below:

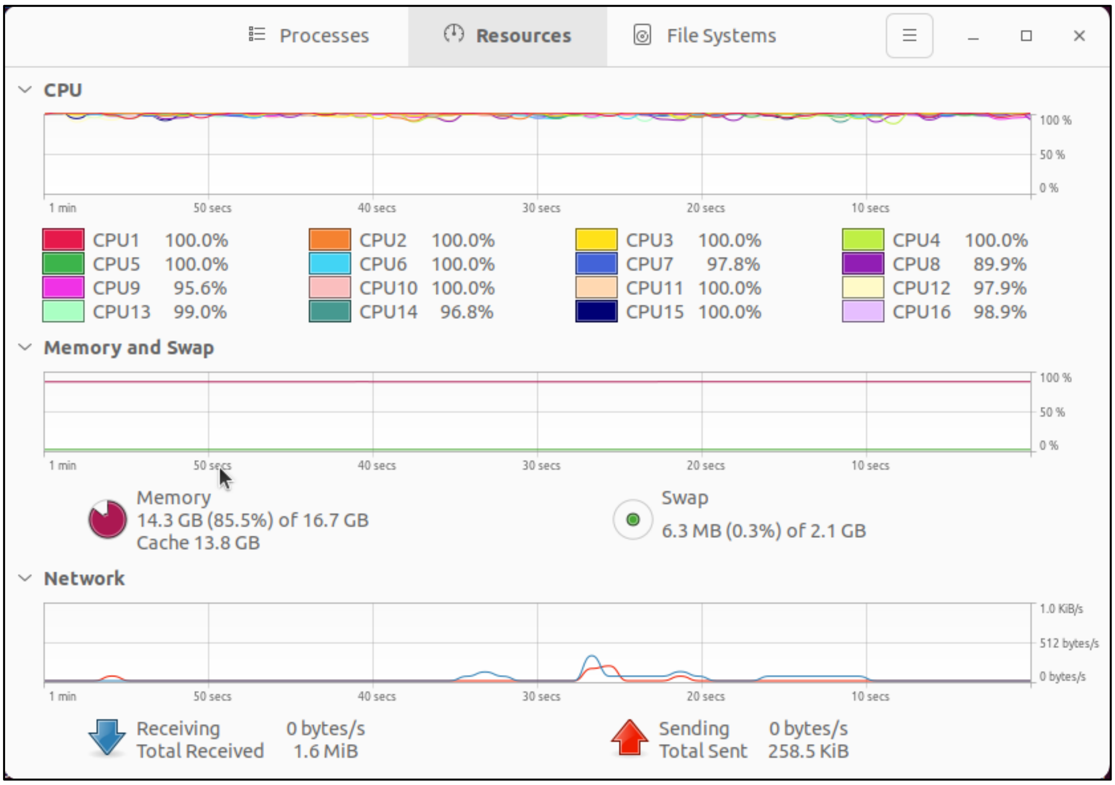
$$(2.341 + 2.447 + 2.31 + 2.328 + 2.333) / 5 = 2.3298 \text{ seconds}$$

This same process is applied to all the other scenarios as well. Finally, we get 2.0390, 2.0518, 2.0856, 2.1216, 2.234, and 2.3298 seconds as the averages for 2, 10, 20, 30, 40, 50, and 60 peer nodes respectively. In essence, average transaction speeds represent typical duration in which it takes to complete the transaction with a single orderer node in a Hyperledger Fabric Network. Figure 6 shows the average transaction speed represented as a line graph across all the experiments. One can see the obvious change in speed when the number of peer nodes are gradually increased in the network.

We start off at 2 peers with 2 organizations in a blockchain network, which has an average transaction time of 2 seconds 39 milliseconds. This establishes the typical time an absolute minimum size of blockchain network could achieve to process a transaction through a local test setup environment. Moving on, we establish a 5 organizations network, consisting of 10 peers, that is 2 peer nodes for every organization. The author found an average time of 2 seconds 52 milliseconds. This signifies a slow increase of transaction time compared to the previous run. However, it is just the beginning of a gradual decline of speed the more peer nodes are introduced into the network. For instance, we see a notable increase to 2 seconds and 86 milliseconds once we conduct the experiment using 20 peer nodes in total. Comparing to 10 peer nodes, we get an increase of 34 milliseconds, which is not entire significant yet. Likewise, moving to a total of 30 peer nodes in the network, the average transaction time increases to 2 seconds and 122 milliseconds. Again, the increase is only 36 milliseconds in this scenario.

The significant increase in transaction time commences when the total peer nodes are at 40. The figure stands at 2 seconds and 169 milliseconds, an arguably noticeable increase of 47 milliseconds compared to having 30 peer nodes total. The surprising turn comes at this next scenario when the experiment was conducted with 50 peer nodes total. The author obtains an average of 2 seconds and 234 milliseconds of transaction time in the local network, which is an increase of 66 milliseconds—a remarkable increase of almost doubled compared to that of experimenting with 10 peer nodes total. Finally, we see average transaction time rose to its highest yet, at 2 seconds 330 milliseconds when there are 60 peer nodes inside the network. This produces 95 milliseconds slower transaction speed compared to 50 peer nodes, or 278 milliseconds when compared with only 10 peer nodes total. This whole experiment reveals a lot about the potential of blockchain network based on Hyperledger Fabric, and particularly how it may be used in a supply chain network seeing how its performance can degrade over the number of total participants. As we increase the network size, the more time is needed to process a transaction across the whole network before it can be cemented into the blockchain ledgers. Due to its distributed nature, blockchain inevitably suffers from performance bottleneck as it entails the number of total participants with the consensus protocol, as well as the endorsement policy. With this experiment, the author uses rather simple network configurations where there is no extra load on the network performance such as Certificate Authority (CA) or even external database solution like CouchDB, aside from intended factors such as orderer and peer nodes.

Noticeably, the size of the network being kept at 5 organizations and 60 peer nodes is not a coincidence. Because the experiment is being conducted on a local virtual machine meant to emulate tens of computer nodes, the size of the network has to hit a hardware bottleneck somewhere. The author has found the limit to be at 60 peer nodes per available computing resources. This research keeps the size at these 60 peer nodes to account for performance loss and network stability. As higher the number of peer nodes go past this amount, the more unstable the network is—mainly due to inadequate CPU cycles being distributed for peer nodes, rendering the network unresponsive and leading to it inevitably failing. In more simple terms, the computing resources are unable to simulate more than 60 peer nodes in this form of experiment due to hardware limitation. Taking a look at figure 7 above, we see that the CPU is all used up simulating 60 peer nodes for this blockchain network. Also, this is the exact reason why we see a rather startling reveal of increasing network performance loss during experiments with 50 and 60 peer nodes total in the network. As CPUs are being bombarded with instructions, it can no longer simulate the interactions between orderer to peers and peers to peers instantly and reliably, thereby forcing the whole network to wait or in some cases freeze and quit entirely.



CPU1 100.0% CPU2 100.0% CPU3 100.0% CPU4 100.0%
CPU5 100.0% CPU6 100.0% CPU7 97.8% CPU8 89.9%
CPU9 95.6% CPU10 100.0% CPU11 100.0% CPU12 97.9%
CPU13 99.0% CPU14 96.8% CPU15 100.0% CPU16 98.9%

Memory
14.3 GB (85.5%) of 16.7 GB
Cache 13.8 GB

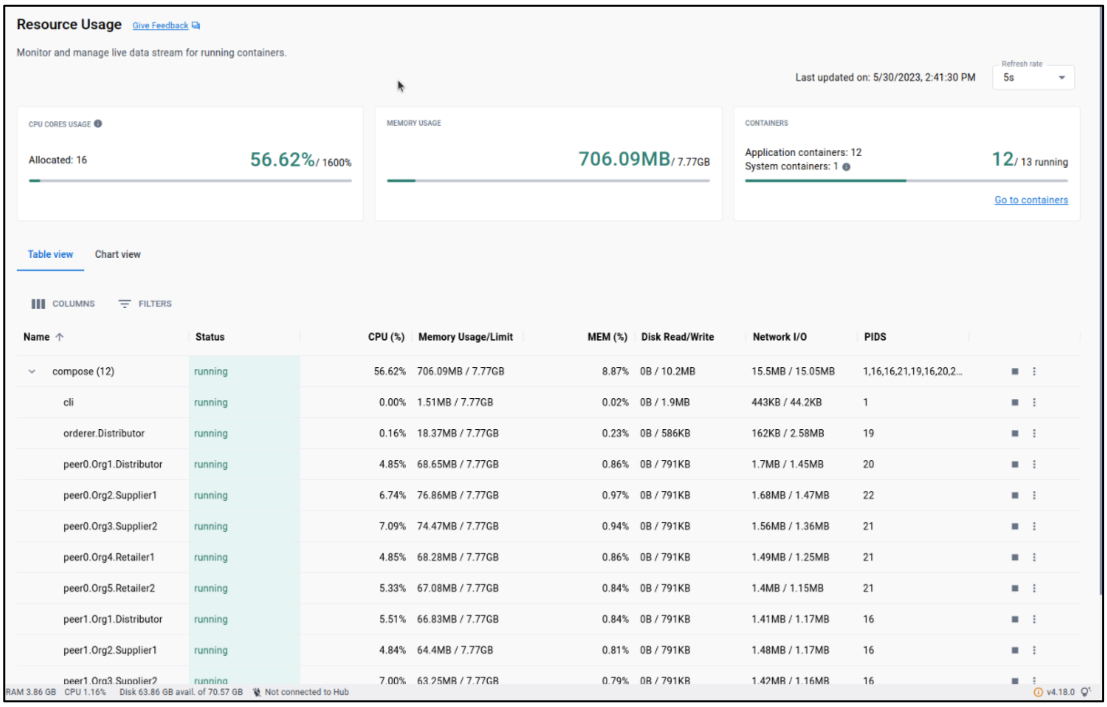
Swap
6.3 MB (0.3%) of 2.1 GB

Receiving 0 bytes/s
Total Received 1.6 MiB

Sending 0 bytes/s
Total Sent 258.5 KiB

Figure 7 Computing Resources Used

Besides, looking at figure 8 and 9, we can see that the orderer and each peer occupies some percentages of the computing resources as they are running. These tiny percentages add up to collectively and effectively bottleneck the system, making it totally unresponsive when reaching a certain threshold as per discussed. The main reason is as an increasing number of peers are being simulated on the network, the more interactions are made between them, primarily with a protocol known as Gossip Data Dissemination Protocol (Fabric, 2023c). As Blockchain networks are distributed and lack any sort of central authority by nature, Hyperledger Fabric peer nodes in this research need to communicate to send or receive data to and from each other using gRPC (Gęsior, 2020) and proto buffers for bi-directional communication about channel membership, peer discovery, as well as to pull missing data from other peers, and to send data to new peers on the network. All peer nodes in this experiment also send signed heartbeats to each other over an interval period of time to acknowledge and check for each other's presence on the network. As a result, these reasons encompass the limitation set on out local testing environment of a blockchain network such as this one.



Resource Usage Give Feedback

Monitor and manage live data stream for running containers.

Last updated on: 5/30/2023, 2:41:30 PM   Refresh rate 5s

CPU CORES USAGE
Allocated: 16   56.62%/ 1600%

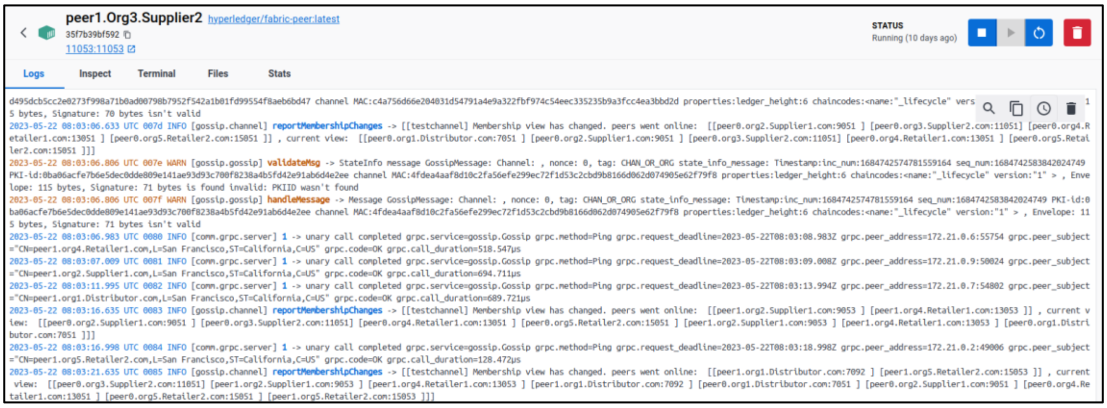
MEMORY USAGE
706.09MB/ 7.77GB

CONTAINERS
Application containers: 12
System containers: 1
12/ 13 running
Go to containers

Name	Status	CPU (%)	Memory Usage/Limit	MEM (%)	Disk Read/Write	Network I/O	PIDS
compose (12)	running	56.62%	706.09MB / 7.77GB	8.87%	0B / 10.2MB	15.5MB / 15.05MB	1,16,16,21,19,16,20,2...
cli	running	0.00%	1.51MB / 7.77GB	0.02%	0B / 1.9MB	443KB / 44.2KB	1
orderer.Distributor	running	0.16%	18.37MB / 7.77GB	0.23%	0B / 586KB	162KB / 2.58MB	19
peer0.Org1.Distributor	running	4.85%	68.65MB / 7.77GB	0.86%	0B / 791KB	1.7MB / 1.45MB	20
peer0.Org2.Supplier1	running	6.74%	76.86MB / 7.77GB	0.97%	0B / 791KB	1.68MB / 1.47MB	22
peer0.Org3.Supplier2	running	7.09%	74.47MB / 7.77GB	0.94%	0B / 791KB	1.56MB / 1.36MB	21
peer0.Org4.Retailer1	running	4.85%	68.28MB / 7.77GB	0.86%	0B / 791KB	1.49MB / 1.25MB	21
peer0.Org5.Retailer2	running	5.33%	67.08MB / 7.77GB	0.84%	0B / 791KB	1.4MB / 1.15MB	21
peer1.Org1.Distributor	running	5.51%	66.83MB / 7.77GB	0.84%	0B / 791KB	1.41MB / 1.17MB	16
peer1.Org2.Supplier1	running	4.84%	64.4MB / 7.77GB	0.81%	0B / 791KB	1.48MB / 1.17MB	16
peer1.Org3.Supplier2	running	7.00%	63.25MB / 7.77GB	0.79%	0B / 791KB	1.42MB / 1.16MB	16

RAM 3.86 GB   CPU 1.16%   Disk 63.86 GB avail. of 70.57 GB   Not connected to Hub   v4.18.0

Figure 8 Resources used in a test



peer1.Org3.Supplier2 hyperledger/fabric-peer:latest
35f7b39bf592
11053:11053

STATUS
Running (10 days ago)

```
d495dcb5cc2e0273f998a71b0ad00798b7952f542a1b01fd99554f8aeb6bd47 channel MAC:c4a756d66e204031d54791a4e9a322fbf974c54eec335235b9a3fcc4ea3bbd2d properties:ledger_height:6 chaincodes:<name:"_lifecycle" vers          1
5 bytes, Signature: 70 bytes isn't valid
2023-05-22 08:03:06.633 UTC 007d INFO [gossip.channel] reportMembershipChanges -> [[testchannel] Membership view has changed. peers went online:  [[peer0.org2.Supplier1.com:9051 ] [peer0.org3.Supplier2.com:11051] [peer0.org4.R
etailer1.com:13051 ] [peer0.org5.Retailer2.com:15051 ]] , current view:  [[peer0.org1.Distributor.com:7051 ] [peer0.org2.Supplier1.com:9051 ] [peer0.org3.Supplier2.com:11051] [peer0.org4.Retailer1.com:13051 ] [peer0.org5.Retai
ler2.com:15051 ]]]
2023-05-22 08:03:06.806 UTC 007e WARN [gossip.gossip] validateMsg -> StateInfo message GossipMessage: Channel: , nonce: 0, tag: CHAN_OR_ORG state_info_message: Timestamp:inc_num:1684742574781559164 seq_num:1684742583842024749
PKI-id:0ba06acfe7b6e5dec0dde809e141ae93d93c700f8238a4b5fd42e91ab6d4e2ee channel MAC:4fdea4aaf8d10c2fa56efe299ec72f1d53c2cbd9b8166d062d074905e62f79f8 properties:ledger_height:6 chaincodes:<name:"_lifecycle" version:"1" > , Enve
lope: 115 bytes, Signature: 71 bytes is found invalid: PKIID wasn't found
2023-05-22 08:03:06.806 UTC 007f WARN [gossip.gossip] handleMessage -> Message GossipMessage: Channel: , nonce: 0, tag: CHAN_OR_ORG state_info_message: Timestamp:inc_num:1684742574781559164 seq_num:1684742583842024749 PKI-id:0
ba06acfe7b6e5dec0dde809e141ae93d93c700f8238a4b5fd42e91ab6d4e2ee channel MAC:4fdea4aaf8d10c2fa56efe299ec72f1d53c2cbd9b8166d062d074905e62f79f8 properties:ledger_height:6 chaincodes:<name:"_lifecycle" version:"1" > , Envelope: 11
5 bytes, Signature: 71 bytes isn't valid
2023-05-22 08:03:06.983 UTC 0080 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=gossip.Gossip grpc.method=Ping grpc.request_deadline=2023-05-22T08:03:08.983Z grpc.peer_address=172.21.0.6:55754 grpc.peer_subject
="CN=peer1.org4.Retailer1.com,L=San Francisco,ST=California,C=US" grpc.code=OK grpc.call_duration=518.547µs
2023-05-22 08:03:07.009 UTC 0081 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=gossip.Gossip grpc.method=Ping grpc.request_deadline=2023-05-22T08:03:09.008Z grpc.peer_address=172.21.0.9:50024 grpc.peer_subject
="CN=peer1.org2.Supplier1.com,L=San Francisco,ST=California,C=US" grpc.code=OK grpc.call_duration=694.711µs
2023-05-22 08:03:11.995 UTC 0082 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=gossip.Gossip grpc.method=Ping grpc.request_deadline=2023-05-22T08:03:13.994Z grpc.peer_address=172.21.0.7:54802 grpc.peer_subject
="CN=peer1.org1.Distributor.com,L=San Francisco,ST=California,C=US" grpc.code=OK grpc.call_duration=689.721µs
2023-05-22 08:03:16.635 UTC 0083 INFO [gossip.channel] reportMembershipChanges -> [[testchannel] Membership view has changed. peers went online:  [[peer1.org2.Supplier1.com:9053 ] [peer1.org4.Retailer1.com:13053 ]] , current v
iew:  [[peer0.org2.Supplier1.com:9051 ] [peer0.org3.Supplier2.com:11051] [peer0.org4.Retailer1.com:13051 ] [peer0.org5.Retailer2.com:15051 ] [peer1.org2.Supplier1.com:9053 ] [peer1.org4.Retailer1.com:13053 ] [peer0.org1.Distri
butor.com:7051 ]]]
2023-05-22 08:03:16.998 UTC 0084 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=gossip.Gossip grpc.method=Ping grpc.request_deadline=2023-05-22T08:03:18.998Z grpc.peer_address=172.21.0.2:49006 grpc.peer_subject
="CN=peer1.org5.Retailer2.com,L=San Francisco,ST=California,C=US" grpc.code=OK grpc.call_duration=128.472µs
2023-05-22 08:03:21.635 UTC 0085 INFO [gossip.channel] reportMembershipChanges -> [[testchannel] Membership view has changed. peers went online:  [[peer1.org1.Distributor.com:7092 ] [peer1.org5.Retailer2.com:15053 ]] , current
 view:  [[peer0.org3.Supplier2.com:11051] [peer1.org2.Supplier1.com:9053 ] [peer1.org4.Retailer1.com:13053 ] [peer1.org1.Distributor.com:7092 ] [peer0.org1.Distributor.com:7051 ] [peer0.org2.Supplier1.com:9051 ] [peer0.org4.Re
tailer1.com:13051 ] [peer0.org5.Retailer2.com:15051 ] [peer1.org5.Retailer2.com:15053 ]]]
```

Figure 9 Gossip data dissemination protocol as used by a peer node

Smart Contract-Enabled Blockchain Framework and Applications in Agri-Food Supply Chain

Using the network topology established earlier in this research as an example, blockchain and smart contract can be integrated into said supply chain network for many general purposes. Below is another blockchain network topology where blockchain technology and smart contract can be integrated into a typical food supply chain. We have 5 total members within this blockchain network: Producer, Manufacturer, Distributor, Retailer, and Customer. These business allies have agreed to establish a blockchain-based supply chain network in which they can partake with various activities. However, for the sake of simplicity, here we establish an example where members use smart contract to automate order processing and payment in this blockchain network. Because this blockchain network is permissioned, there is no worry that any unauthorized third-party user or malicious actor could have access to the information shared among members within the same channel or network.

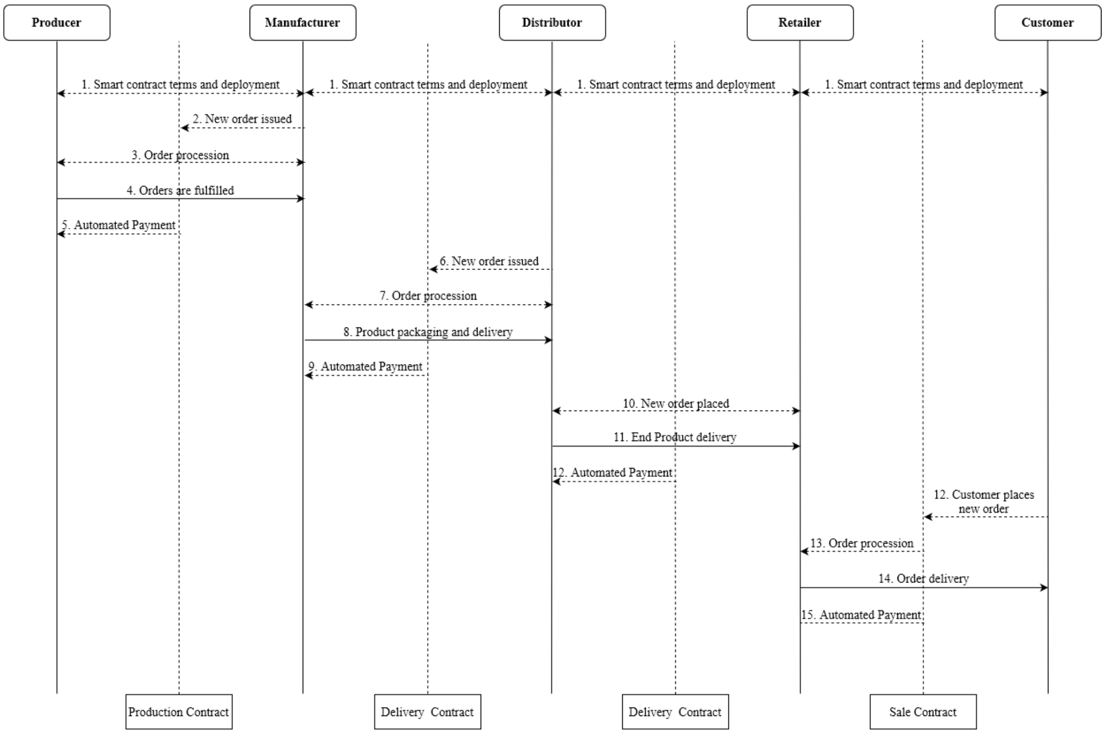


Figure 10 Smart contract implementations in a blockchain-based agri-food supply chain network

From figure 10, there are 5 members inside a blockchain network, using 4 smart contracts: Production Contract between Producer and Manufacturer, Delivery Contract between Manufacturer and Distributor, another Delivery Contract between Distributor and Retailer, and Sale Contract between Retailer and Customer. Each of these contracts can be established by all participants negotiating terms and conditions upon which the transactions and other activities are agreed. These contractual terms can be made into business logics to be implemented inside smart contracts. These business logics should also be automatically invoked using certain activities or status. For instance, Production contract is established with an order coming from Manufacturer which alerts Producer. Producer can then prepare and fulfill the order in step 4, after which the payment will be automatically processed and sent when the goods arrived in certain conditions. These statuses and activities can be translated into business logic inside Production Contract, to which business logic and terms only Producer and Manufacturer has access. Additionally, Manufacturer and Distributor may establish an entirely different smart contract pertaining to logistics of the product. This contract may focus on punctual delivery of said product in good condition that may require logs of product’s status along the way. Whenever Distributor issues a new order in step 6, Manufacturer may start recording the logs of product’s status, from packaging to delivery. Along the way, the logs are automatically recorded until it reaches the destination, upon which the smart contract may activate its final process—to pay manufacturer if the product’s quality is in certain acceptable conditions. All of these activities can be recorded into the blockchain as transaction updates that cannot be modified, and thus, ensure immutability, truthfulness and transparency of data.

Next, another delivery contract between Distributor and Retailer is activated in step 10. Distributor and Retailer may negotiate certain conditions of the order such that it may require certain agreements to order in bulk where both parties have to approve beforehand. Therefore, this Delivery Contract may operate in a distinct manner compared to the previous Delivery Contract. Whenever the order is fulfilled in step 12, the automated payment system setup through the use of smart contract and either digital currency system within the blockchain, or even external payment gateway can be used to transfer payment between both parties. Finally, the smart contract created between Retailer and Customer may involve the use of digital payment system established within the blockchain as to ensure seamless payment and extra advantages. Customer may use the Sale Contract to send order to Retailer, who may utilize this same smart contract to track the order from when it is picked from the warehouse, to packaging, to delivery, and finally to customer's destination. By utilizing creative use of smart contract with data immutability of blockchain, this smart contract can offer plenty of unique selling points to the businesses involved.

More instances of how smart contracts can be utilized, the author encourages readers to explore his previous works about blockchain (Kang & Indra-Payoong, 2019; Kang & Indra-Payoong, 2021). In these articles, the author has explored the conceptual nature of blockchain and smart contract utilization in supply chain where more details regarding smart contracts are further explained. Although the blockchain and smart contract platform itself is not based on Hyperledger Fabric, the fundamentals regarding their usage and practicality may still be applicable and utilized to further enhance the supply chain effectiveness and performance. Besides, the official Hyperledger Fabric case studies offer more practical usage of Hyperleger Fabric Blockchain and smart contracts in which many real-world applications of such system are being tested. One such case is the blockchain system utilized in finance industry for firms like: taXchain, GSBN, Splunk S&P, Joisto, Mindtree, Deutsche Börse Group, (Fabric, 2017, 2018b, 2019b, 2019c, 2020b, 2021a). There are also a number of case studies done in supply chain industries with the following companies: Fujitsu and Botanical Water Technologies for recovering water during agricultural farming processes. (Fabric, 2022), MineHub-KrypC for automated mining workflow globally (Fabric, 2021b), and Dubai's digital Silk Road that prioritizes trade (Fabric, 2020a).

Moreover, some other studies have managed to implement blockchain and smart contract to manage both physical and information flow in agri-food supply chain. For instance, Walmart and DLT Labs collaborated to manage payment and other constant data flow between carriers and Walmart Canada across its 400 stores (Fabric, 2019a). in 2019, DLT labs and Walmart launched a blockchain and smart contract-based network to track, verify, and automate transactions. Walmart possess a massive network of suppliers and carriers within a complex supply chain that is hard to sustain with traditional information management systems. The problem regarding invoicing issues and disputes that results in delayed payment costs Walmart millions of dollars a year. Because Walmart's typical food supply chain comprises of 220+ data points per load, with numerous properties assigned to each load, with 500,000 totaled each year. To combat this, Walmart collaborated with DLT Labs to create a blockchain network based on Hyperledger Fabric because of its scalability and modular framework. As a result, the blockchain network feasibly scales and is able to continuously take on data from Walmart's full fleet of 70 carriers, all in real time and 24/7. The platform runs on 600 virtual machines (VMs) in order to manage and store data points across all nodes. Furthermore, Walmart is able to cut back invoice disputes from 70% to an impressive 1.5%. The typical error threshold of $10 per invoice went to $0 as the timeline for carrier invoice approval also went from 6-8 weeks to less than a week. Carriers get paid much faster using this blockchain-based network as opposed to the traditional one as Walmart can save time and a lot of resources with invoice procession and spend less time on disputes with its carriers.

Another noticeable instance of blockchain and smart contract application in agri-food supply chain is from Walmart's collaboration with IBM (Fabric, 2018a) as it utilizes Hyperledger Fabric for scalability, flexibility, and transparency inside its agri-food supply chain. As food-borne disease can happen at any moment, it usually takes as long as weeks to find the original source of such events. Walmart implements a blockchain network with support for smart contract, and scale this network up with its many suppliers of mangoes, strawberries, leafy greens and more. Walmart's past attempts at scaling a food traceability system never scaled, mainly because they were all centralized database systems. With blockchain's decentralized nature, as well as its scalability and flexibility, the network manages to decrease the time to trace mango's provenance from 7 days to 2.2 seconds. Now Walmart is attempting to expand more of its products such as: leafy greens, chicken, pork, strawberries, dairy products like milk, and even salad into this blockchain network.

In short, there are much more applications in other industries conducted by various start-ups as well as veteran technology firms around the world. All of these projects adapt Hyperledger Fabric blockchain network with smart contract to various creative degree and success that should not be overlooked. The pilot tests and experiments conducted are what fuels blockchain and smart contract adoption in mainstream businesses and industries, all of which could benefit millions of people across the globe.

CHAPTER 5
CONCLUSION

Conclusion

Supply Chain is one of the most vital industries known to modern human society, encompassing all sorts of manufacturing and logistics of products and services. However, there are many existing issues in the field such as power imbalance, transparency, flexibility, and scalability in modern supply chain. Presently, there have yet to be any solutions capable of decentralizing the authority figure, as well distribute equal power and information sharing to various stakeholders in a scalable manner from which parties such as producers can benefit. Enter blockchain, a modern take on decentralized and distributed network system meant to decouple an authoritative figure from controlling any network with absolute power. Blockchain is an emerging technology, focusing on decentralization and security. It is best known for powering Bitcoin, the most adopted cryptocurrency to date. As blockchain is evolving into a tool meant to be used in various scenarios, Hyperledger Fabric came into existence as an open-source blockchain framework since 2016. Because supply chain industry as a whole is plagued with numerous issues like centralization of power, lack of information sharing, and ineffective scalability across the value chain, blockchain network based on Hyperledger Fabric is currently one of the most cutting-edge solutions being implemented in various industries worldwide.

We have taken a look at the current implementations of blockchain technology in supply chain across various industries. What it entails is that current trend seems to be focusing on decentralization of authoritative powers such as big retailers and manufacturers, increasing security and scalability in supply chain—particularly relating to information sharing, as well supply chain traceability which is a blockchain’s core strength. It can also be concluded that projects regarding blockchain integration into supply chain is on the rise because of its untapped potential in the future. Next, we have also seen an application of blockchain based on Hyperledger Fabric into a test network comprising of 5 organization in a supply chain network. These actors or members are grouped into one blockchain network for a number of experiments. These experiments target the usability of blockchain by utilizing a local testing environment and various pluggable components. The network is recreated many times, each with a different number of peer nodes in order for us to witness its scalability and usability in a local testing environment. The result shows a remarkably usable network, with each member organization achieving consistent information keeping, with the performance limit based solely on hardware limitation. The author believes this experiment could still be scaled up further with a more advanced and sophisticated form of network configuration whether on a local testing with multiple computing units, or on the cloud. Moreover, this research also explained and answered in a detailed manner regarding the process and workflow of how a distributed network such as blockchain could be implemented in a supply chain network. The configuration, workflow, as well as the resources needed were fully explained and shown, with how each element is utilized to such effect that the network can be recreated at will with minimal errors.

This research takes the form of testing because existing literature tend to focus more on frameworks and concepts of blockchain application rather than taking the experimental approach. As such, the author prefers using a rather straight on method of utilizing blockchain and smart contract in supply chain in order to fully explore its potential in terms of security, scalability, and flexibility within a supply chain network since these issues occur frequently across all supply chain industries. The author decides to keep the experimentation within a closed, local testing environment primarily because this method keeps the experimentation more straight forward and more flexible, as well as less complex and less time-consuming. This approach also helps with future experiments as the network can be fully and feasibly recreated with any local testing environment, and its components be inspected at will. Since each required component and software is also fully laid out, any errors and inspection can be done in a short time as compared to a more advanced and multi-host setup.

Recommendation

Based on experimentation done in this research, the author would recommend further this testing into a more advanced setup, with more components and smart contracts at play, as well as more channels and security components such as

CA and multi-host setup in order to further extend the capability of this network. More custom solutions to existing components are also recommended, such as custom consensus protocol, database solution, and more peer node configurations. These future configurations may be able to address the issues such as transparency information sharing, enhanced security, as well as flexible business solutions to transaction and communication.

In addition, supply chain industry witnesses a perpetual shifting in conducts for decades due to advancement of technology, shifting customer preferences, increasing value stream complexity, logistics, and more. Because of this, issues around security, flexibility, and scalability pose a serious threat to supply chain’s performance around the world. With the introduction of blockchain and smart contract technology, the author combines the usability of both blockchain and smart contract, with supply chain network, to form an experimental network—one that aims at solving these issues. Besides, through an extensive review of literature, the author found that there is a serious lack of blockchain experimentation in various supply chain industries to try and solve these issues. Many researches focus on frameworks and conceptual designs that have little to do with experimentation and configuration design, both of which are the core principles of scientific testing of this nature. The result is an extremely promising blockchain network capable of hosting 60 peer nodes and sustaining transaction invocation, with data safely stored with all of peers who maintain constant connection. This also increases security and network flexibility as various network configurations can be done as desired. The result also verifies that blockchain utilization in supply chain is a competent and useful novel solution—one that has a ton of potential besides what the author has explained and explored.

REFERENCES

Adamashvili, N., State, R., Tricase, C., & Fiore, M. (2021). Blockchain-based wine supply chain for the industry advancement. *Sustainability, 13*(23), 13070.

Adhav, P. (2020). *System Chaincodes in Hyperledger Fabric—VSCC, ESCC, LSCC, ESCC, QSCC*. Retrieved from https://medium.com/coinmonks/system-chaincodes-in-hyperledger-fabric-vscc-escc-lscc-cscc-a48db4d24dc3

Albino, V., Balice, A., & Dangelico, R. M. (2009). Environmental strategies and green product development: an overview on sustainability-driven companies. *Business strategy and the environment*, *18*(2), 83-96.

Andrade-Salinas, G., Salazar-Chacon, G., & Vintimilla, L.-M. (2020). *Integration of IoT Equipment as Transactional Endorsing Peers over a Hyperledger-Fabric Blockchain Network: Feasibility Study.* Paper presented at the Applied Technologies: First International Conference, ICAT 2019, Quito, Ecuador, December 3–5, 2019, Proceedings, Part I 1.

Antonucci, F., Figorilli, S., Costa, C., Pallottino, F., Raso, L., & Menesatti, P. (2019). A review on blockchain applications in the agri-food sector. *Journal of the Science of Food and Agriculture, 99*(14), 6129-6138.

Aste, T., Tasca, P., & Di Matteo, T. (2017). Blockchain technologies: The foreseeable impact on society and industry. C*omputer, 50*(9), 18-28.

Azzi, R., Chamoun, R. K., & Sokhn, M. (2019). The power of a blockchain-based supply chain. *Computers & Industrial Engineering, 135*, 582-592.

Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., & Danezis, G. (2017). Consensus in the age of blockchains. *arXiv preprint arXiv:1711.03936*.

Belchior, R. (2019). *Hyperledger Fabric: Technical Overview*. Retrieved from https://towardsdatascience.com/hyperledger-fabric-technical-overview-a63046c2a430

Blizkovsky, P., & Berendes, V. (2016). Economics of fairness within the food supply chain in context of the EU. *Applied Studies in Agribusiness and Commerce, 10*(4-5), 107-116.

Hill, B, Chopra, S., & Valencourt, P. (2023). *Blockchain Quick*. Retrieved from https://www.oreilly.com/library/view/blockchain-quick-reference/9781788995788/9b206449-c4b6-4011-a8fc-18fb3a191236.xhtml

Bunte, F. (2006). *Pricing and performance in agri-food supply chains.* In proceedings of the Frontis workshop on quantifying the agri-food supply chain, Wageningen, The Netherlands, 22-24 October 2004.

Cai, W., Wang, Z., Ernst, J. B., Hong, Z., Feng, C., & Leung, V. C. (2018). Decentralized applications: The blockchain-empowered software system. *IEEE Access, 6*, 53019-53033.

Chen, G., Xu, B., Lu, M., & Chen, N.-S. (2018). Exploring blockchain technology and its potential applications for education. *Smart Learning Environments, 5*(1), 1.

Chen, K. (2006). *Agri-food supply chain management: opportunities, issues, and guidelines.* In Proceedings of the International Conference on Livestock Services. Beijing, People’s Republic of China, April.

Chen, K., Shepherd, A., & Silva, C. d. (2005). Changes in food retailing in Asia.

Chiengkul, P. (2017). *The political economy of the agri-food system in Thailand: Hegemony, counter-hegemony, and co-optation of oppositions* (Vol. 90): Taylor & Francis.

Chiranjeevi, K., Tripathi, M. K., & Maktedar, D. D. (2021). *Block chain Technology in Agriculture Product Supply Chain.* Paper presented at the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS).

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE Access, 4*, 2292-2303.

Clark, J., Edward, A., & Felten, W. (2015). *Research perspectives and challenges for bitcoin and cryptocurrencies*. Retrieved from *https://eprint. iacr. org/2015/261. pdf,* 106.

Cocco, L., Mannaro, K., Tonelli, R., Mariani, L., Lodi, M. B., Melis, A., . . . Fanti, A. (2021). A blockchain-based traceability system in agri-food SME: Case study of a traditional bakery. *IEEE Access, 9*, 62899-62915.

Cong, L. W., & He, Z. (2019). Blockchain disruption and smart contracts. *The Review of Financial Studies, 32*(5), 1754-1797.

Dangelico, R. M., & Pujari, D. (2010). Mainstreaming green product innovation: Why and how companies integrate environmental sustainability. *Journal of Business Ethics, 95*, 471-486.

de Carvalho, P. R., Naoum-Sawaya, J., & Elhedhli, S. (2022). Blockchain-Enabled Supply Chains: An Application in Fresh-Cut Flowers. *Applied Mathematical Modelling*.

devendrasalunke. (2022). *Hyperledger Fabric Component Design*. Retrieved from https://www.geeksforgeeks.org/hyperledger-fabric-component-design/

Di Vaio, A., & Varriale, L. (2020). Blockchain technology in supply chain management for sustainable performance: Evidence from the airport industry. *International Journal of Information Management, 52*, 102014.

Dutta, P., Choi, T.-M., Somani, S., & Butala, R. (2020). Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transportation research part e: Logistics and transportation review, 142*, 102067.

Elbashbishy, T., Ali, G., & El-adaway, I. H. (2022). *Role of Transactional Blockchain in Facilitating Procurement in International Construction Projects.* In Proceedings of the Canadian Society of Civil Engineering Annual Conference 2021: CSCE21 Construction Track Volume 1.

Fabric, H. (2023a). *Create a Channel*. Retrieved from https://hyperledger-fabric.readthedocs.io/en/release-2.5/create_channel/create_channel_participation.html

Fabric, H. (2023b). *Getting Started - Install*. Retrieved from https://hyperledger-fabric.readthedocs.io/en/release-2.5/getting_started.html

Fabric, H. (2023c). *Gossip data dissemination protocol*. Retrieved from https://hyperledger-fabric.readthedocs.io/en/release-2.5/gossip.html

Fabric, H. (2023d). *Install Fabric and Fabric Samples*. Retrieved from https://hyperledger-fabric.readthedocs.io/en/release-2.5/install.html

Fabric, H. (2023e). *Transaction Flow*. Retrieved from https://hyperledger-fabric.readthedocs.io/en/release-2.5/txflow.html

Fair World, P. (2018). *Fairness for Farmers: A Report Assessing the Fair Trade Movement and the Role of Certification*. Retrieved from https://fairworldproject.org/wp-content/uploads/2018/09/Fair_World_Project-Fairness-For-Farmers-Report.pdf

Fałkowski, J., Ménard, C., Sexton, R. J., Swinnen, J., Vandevelde, S., Di Marcantonio, F., & Ciaian, P. (2017). *Unfair trading practices in the food supply chain: A literature review on methodologies, impacts and regulatory aspects.* Joint Research Centre (JRC), the European Commission’s science and knowledge …,

Francisco, K., & Swanson, D. (2018). The supply chain has no clothes: Technology adoption of blockchain for supply chain transparency. *Logistics, 2*(1), 2.

Fritz, M., & Matopoulos, A. (2008). *Sustainability in the agri-food industry: a literature review and overview of current trends.* 8th International Conference on Chain Network Management in Agribusiness the Food Industry.

Gatteschi, V., Lamberti, F., Demartini, C., Pranteda, C., & Santamaría, V. (2018). Blockchain and smart contracts for insurance: Is the technology mature enough? *Future Internet, 10*(2), 20.

Gausdal, A. H., Czachorowski, K. V., & Solesvik, M. Z. (2018). Applying Blockchain Technology: Evidence from Norwegian Companies. *Sustainability, 10*(6), 1.

Gęsior, T. (2020). *P2P in Hyperledger Fabric*. Retrieved from https://brightinventions.pl/blog/p2p-in-hyperledger-fabric/

Glaser, F. (2017). *Pervasive decentralisation of digital infrastructures: a framework for blockchain enabled system and use case analysis.* In Proceedings of the 50th Hawaii International Conference on System Sciences Hawaii.

Greenspan, G. (2016). *Beware of the Impossible Smart Contract*. Retrieved from https://www.the-blockchain.com/2016/04/12/beware-of-the-impossible-smart-contract/

Hackius, N., & Petersen, M. (2017). *Blockchain in logistics and supply chain: trick or treat?* Paper presented at the Digitalization in Supply Chain Management and Logistics: Smart and Digital Solutions for an Industry 4.0 Environment. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 23.

Hawlitschek, F., Notheisen, B., & Teubner, T. (2018). The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy. *Electronic commerce research and applications, 29,* 50-63.

Healy, A. (2015). *Fresh data shows decline in farmer share of consumer price for food.* Retrieved from http://www.irishtimes.com/news/consumer/fresh-data-shows-decline-in-farmer-share-of-consumerprice-for-food-1.2126097

Hellberg-Bahr, A., & Spiller, A. (2012). How to treat farmers fairly? Results of a farmer survey. *International Food and Agribusiness Management Review, 15*(1030-2016-82929), 87-97.

Hendrickson, M., Howard, P. H., & Constance, D. (2017). Power, food and agriculture: Implications for farmers, consumers and communities. *Consumers and Communities (November 1, 2017).*

Hingley, M. K. (2005). Power imbalance in UK agri-food supply channels: Learning to live with the supermarkets? *Journal of Marketing Management, 21*(1-2), 63-88.

Hyperledger. (2023). *Hyperledger Fabric: Open, Proven, Enterprise-grade DLT.* Retrieved from https://www.hyperledger.org/wp-content/uploads/2020/03/hyperledger_fabric_whitepaper.pdf

Iakovou, E., Vlachos, D., Achillas, C., & Anastasiadis, F. (2012). *A methodological framework for the design of green supply chains for the agrifood sector.* Paper presented at the de 2nd International Conference on Supply Chains, Greece.

Jaffee, S., Siegel, P., & Andrews, C. (2010). Rapid agricultural supply chain risk assessment: A conceptual framework. *Agriculture and rural development discussion paper, 47*(1), 1-64.

Kakarlapudi, P. V., & Mahmoud, Q. H. (2021). *A systematic review of blockchain for consent management.* Paper presented at the Healthcare.

Kang, P., & Indra-Payoong, N. (2019). *A framework of blockchain smart contract in fair trade agriculture.* Paper presented at the Proc. PIM 9th National and 2nd International Conference.

Kang, P., & Indra-Payoong, N. (2021). A Framework of Blockchain Smart Contract for Sustainable Agri-Food Supply Chain. *INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND TECHNOLOGY (ISJET), 5*(2), 1-14.

Kersten, W., Seiter, M., von See, B., Hackius, N., & Maurer, T. (2017). Trends and strategies in logistics and supply chain management–digital transformation opportunities. *BVL: Bremen, Germany*.

Korpela, K., Hallikas, J., & Dahlberg, T. (2017). *Digital supply chain transformation toward blockchain integration.* In proceedings of the 50th Hawaii international conference on system sciences.

Koulu, R. (2016). Blockchains and online dispute resolution: smart contracts as an alternative to enforcement. *SCRIPTed, 13*, 40.

Krishnan, R. (2020). *Hyperledger Fabric - Components Overview*. Retrieved from https://wiki.hyperledger.org/display/HIRC/Hyperledger+Fabric+-+Components+Overview

Lai, K.-h., & Wong, C. W. (2012). Green logistics management and performance: Some empirical evidence from Chinese manufacturing exporters. *Omega, 40*(3), 267-282.

Lau, W. F., Liu, D. Y., & Au, M. H. (2021). *Blockchain-Based Supply Chain System for Traceability, Regulation and Anti-Counterfeiting.* In 2021 IEEE International Conference on Blockchain (Blockchain).

Lauslahti, K., Mattila, J., & Seppala, T. (2017). Smart Contracts–How will blockchain technology affect contractual practices?

Levy, K. E. (2017). Book-smart, not street-smart: blockchain-based smart contracts and the social workings of law. *Engaging Science, Technology, and Society, 3*, 1-15. Retrieved from https://estsjournal.org/index.php/ests/article/view/107/61

Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2017). A survey on the security of blockchain systems. *Future Generation Computer Systems*.

Lin, W., Huang, X., Fang, H., Wang, V., Hua, Y., Wang, J., . . . Yau, L. (2020). Blockchain technology in current agricultural systems: from techniques to applications. *IEEE Access, 8*, 143920-143937.

Madeley, J. (2000). *Hungry for trade: How the poor pay for free trade*: Zed Books.

Maheshwari, S. (2018). *Blockchain Basics: Hyperledger Fabric*. Retrieved from https://developer.ibm.com/articles/blockchain-basics-hyperledger-fabric/

Mark, J. (2006). *Workers on organic farms are treated as poorly as their conventional counterparts*. Retrieved from https://grist.org/article/mark/

Marr, B. (2018, Feb 16). A Very Brief History Of Blockchain Technology Everyone Should Read. Retrieved from https://www.forbes.com/sites/bernardmarr /2018/02/16/a-very-brief-history-of-blockchain-technology-everyone-should-read/#3d4529a87bc4

Mazoyer, M. (2001). Protecting small farmers and the rural poor in the context of globalization.

Mohanta, B. K., Panda, S. S., & Jena, D. (2018). *An overview of smart contract and use cases in blockchain technology*. In 2018 9th international conference on computing, communication and networking technologies (ICCCNT).

Montecchi, M., Plangger, K., & West, D. C. (2021). Supply chain transparency: A bibliometric review and research agenda. *International Journal of Production Economics, 238*, 108152.

Nakamoto, S. (2008). Bitcoin: *A peer-to-peer electronic cash system*. Retrieved from https://bitcoin.org/bitcoin.pdf

Ongaro, D., & Ousterhout, J. (2014). *In search of an understandable consensus algorithm.* Paper presented at the 2014 USENIX Annual Technical Conference (Usenix ATC 14).

Ongaro, D., & Ousterhout, J. (2015). The raft consensus algorithm. *Lecture Notes CS.* Retrieved from https://raft.github.io/slides/riconwest2013.pdf

Padel, S., & Zander, K. (2010). Regional production'and 'Fairness' in organic farming: Evidence from a CORE Organic project. *WS4. 3–Fair and regional: New trends of organic and sustainable food systems*.

Patnayakuni, R., Rai, A., & Seth, N. (2006). Relational antecedents of information flow integration for supply chain coordination. *Journal of management information systems, 23*(1), 13-49.

Polack, E., Cotula, L., Blackmore, E., & Guttal, S. (2014). *Agricultural investments in Southeast Asia: Legal tools for public accountability*: International Institute for Environment and Development.

Pouliot, S., & Sumner, D. A. (2008). Traceability, liability, and incentives for food safety and quality. *American Journal of Agricultural Economics, 90*(1), 15-27.

Pournader, M., Shi, Y., Seuring, S., & Koh, S. L. (2020). Blockchain applications in supply chains, transport and logistics: a systematic review of the literature. *International Journal of Production Research, 58*(7), 2063-2081.

Reddy, G. J., Kumar, G. H. S., Lohitasya, T., Nilay, V. S., Praveen, K. S., Egala, B. S., & Pradhan, A. K. (2021). *FarmersChain: A Decentralized Farmer Centric Supply Chain Management System Using Blockchain and IoT.* Paper presented at the 2021 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS).

Rilee, K. (2018). *Understanding Hyperledger Fabric -- Endorsing Transactions*. Retrieved from https://medium.com/kokster/hyperledger-fabric-endorsing-transactions-3c1b7251a709

Risius, M., & Spohrer, K. (2017). A blockchain research framework. *Business & Information Systems Engineering, 59*(6), 385-409.

Sathya, D., Nithyaroopa, S., Jagadeesan, D., & Jacob, I. J. (2021). *Block-chain technology for food supply chains.* In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV).

Savelyev, A. (2017). Contract law 2.0:‘Smart’contracts as the beginning of the end of classic contract law. *Information & Communications Technology Law, 26*(2), 116-134.

Schutter, O. (2014). *Who's Got The Power? Tackling Imbalances In Agricultural Supply Chains*. Retrieved from https://fairtrade-advocacy.org/wp-content/uploads/2019/03/EN-Whos_got_the_power-full_report.pdf

Seebacher, S., & Schüritz, R. (2017). *Blockchain technology as an enabler of service systems: A structured literature review.* In International Conference on Exploring Services Science.

Segal, R., & Le Nguyet, M. (2019). Unfair Harvest: The state of rice in Asia.

Seuring, S., & Müller, M. (2008). Core issues in sustainable supply chain management–a Delphi study. *Business strategy and the environment, 17*(8), 455-466.

Singh, J. (1996). The importance of information flow within the supply chain. *Logistics Information Management*.

Sternberg, H. S., Hofmann, E., & Roeck, D. (2021). The struggle is real: insights from a supply chain blockchain case. *Journal of Business Logistics, 42*(1), 71-87.

Sund, T., & Lööf, C. (2019). *Performance evaluation of a blockchain-based traceability system: A case study at IKEA.*

Sunny, J., Undralla, N., & Pillai, V. M. (2020). Supply chain transparency through blockchain-based traceability: An overview with demonstration. *Computers & Industrial Engineering, 150*, 106895.

Surjandari, I., Yusuf, H., Laoh, E., & Maulida, R. (2021). Designing a Permissioned Blockchain Network for the Halal Industry using Hyperledger Fabric with multiple channels and the raft consensus mechanism. *Journal of Big Data, 8*(1), 1-16.

Swan, M. (2015). *Blockchain: Blueprint for a new economy*. United States of America: " O'Reilly Media, Inc.".

Szabo, N. (1997). *Formalizing and securing relationships on public networks*. *First Monday.* Retrieved from https://firstmonday.org/ojs/index.php/fm/article/ view/548/469.DOI:http:/dx.doi.org/10.5210/fm.v2i9.548

Thompson, J., Millstone, E., Scoones, I., Ely, A., Marshall, F., Shah, E., . . . Wilkinson, J. (2007). Agri-food system dynamics: pathways to sustainability in an era of uncertainty.

Tikhomirov, S. (2017). *Ethereum: state of knowledge and research perspectives.* Paper presented at the International Symposium on Foundations and Practice of Security.

Tribis, Y., El Bouchti, A., & Bouayad, H. (2018). *Supply Chain Management based on Blockchain: A Systematic Mapping Study.* Paper presented at the MATEC Web of Conferences.

Tseng, M.-L., Tan, R. R., & Siriban-Manalang, A. B. (2013). Sustainable consumption and production for Asia: sustainability through green design and practice. *Journal of Cleaner Production, 40*, 1-5.

Tsolakis, N. K., Keramydas, C. A., Toka, A. K., Aidonis, D. A., & Iakovou, E. T. (2014). Agrifood supply chain management: A comprehensive hierarchical decision-making framework and a critical taxonomy. *Biosystems Engineering, 120*, 47-64.

Vanpoucke, E., Boyer, K. K., & Vereecke, A. (2009). Supply chain information flow strategies: an empirical taxonomy. *International Journal of Operations & Production Management*.

Viriyasitavat, W., & Hoonsopon, D. (2018). Blockchain characteristics and consensus in modern business processes. *Journal of Industrial Information Integration*, 1.

Vo, K. T., Nguyen-Thi, A.-T., & Nguyen-Hoang, T.-A. (2021). *Building Sustainable Food Supply Chain Management System Based On Hyperledger Fabric Blockchain.* In 15th International Conference on Advanced Computing and Applications (ACOMP).

Wamba, S. F., & Queiroz, M. M. (2020). Blockchain in the operations and supply chain management: Benefits, challenges and future research opportunities. In (Vol. 52, pp. 102064): Elsevier.

Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., & Kishigami, J. (2016). *Blockchain contract: Securing a blockchain applied to smart contracts.* In 2016 IEEE International Conference on Consumer Electronics (ICCE).

Xiong, H., Dalhaus, T., Wang, P., & Huang, J. (2020). Blockchain technology for agriculture: applications and rationale. *Frontiers in Blockchain, 3*, 7.

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology?—a systematic review. *PloS one, 11*(10), e0163477.

Yoo, M., & Won, Y. (2018). A study on the transparent price tracing system in supply chain management based on blockchain. *Sustainability, 10*(11), 4037.

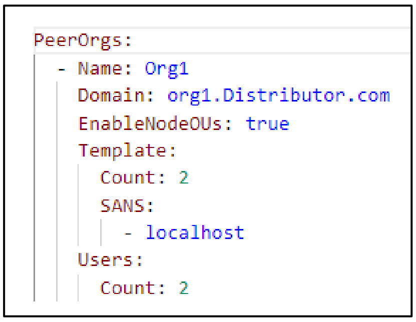
Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In IEEE 6th International Congress on Big Data, Honolulu, Hawaii, USA.

APPENDIX A

Network Creation

The blockchain network in this research can be recreated using the following process:

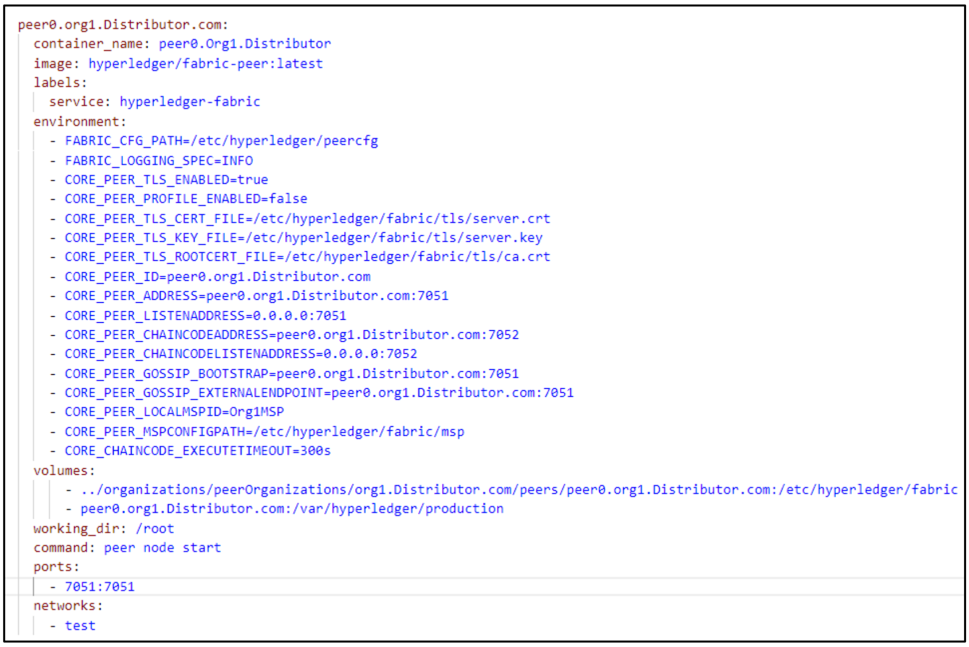
1. Crypto config files for each member organization + orderer. The command used to generate the organization artifacts is: “cryptogen generate –config={PATH TO FILE} –output={PATH}



```
PeerOrgs:
  - Name: Org1
    Domain: org1.Distributor.com
    EnableNodeOUs: true
    Template:
      Count: 2
      SANS:
        - localhost
    Users:
      Count: 2
```

Appendix A.1 Cryptogen file used to generate certificates

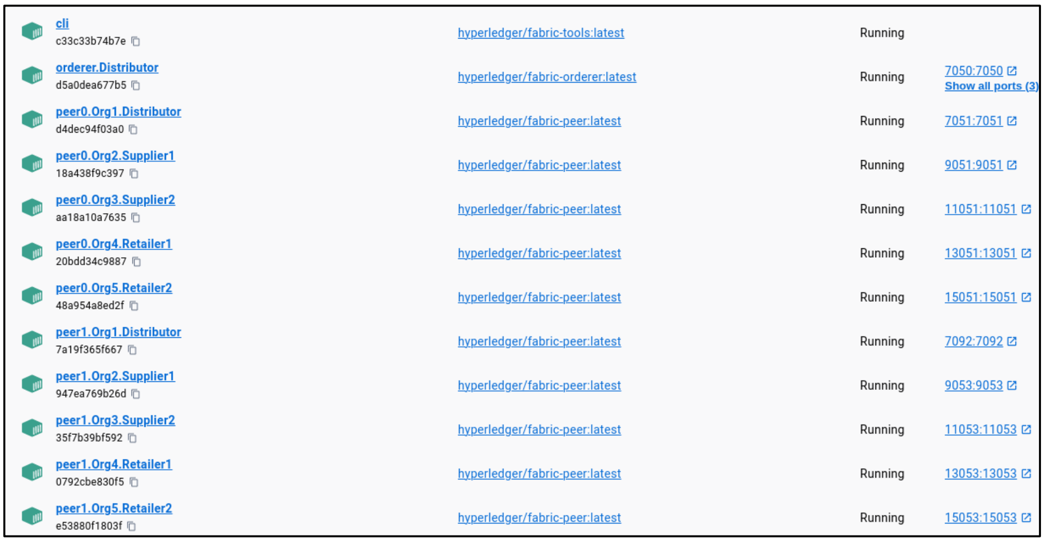
2. Next, a docker-compose file containing details of orderer and each peer nodes such as: container name, images, configurations, commands, volumes, as well as ports to be exposed so that each peer component can communicate to each other within the blockchain network. Docker helps isolate the connection of blockchain components from outsiders so that data tampering is prohibited. The command used to launch the containers is “docker-compose -f {PATH TO FILE} up -d”



```
peer0.org1.Distributor.com:
  container_name: peer0.Org1.Distributor
  image: hyperledger/fabric-peer:latest
  labels:
    service: hyperledger-fabric
  environment:
    - FABRIC_CFG_PATH=/etc/hyperledger/peercfg
    - FABRIC_LOGGING_SPEC=INFO
    - CORE_PEER_TLS_ENABLED=true
    - CORE_PEER_PROFILE_ENABLED=false
    - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
    - CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
    - CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
    - CORE_PEER_ID=peer0.org1.Distributor.com
    - CORE_PEER_ADDRESS=peer0.org1.Distributor.com:7051
    - CORE_PEER_LISTENADDRESS=0.0.0.0:7051
    - CORE_PEER_CHAINCODEADDRESS=peer0.org1.Distributor.com:7052
    - CORE_PEER_CHAINCODELISTENADDRESS=0.0.0.0:7052
    - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org1.Distributor.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.Distributor.com:7051
    - CORE_PEER_LOCALMSPID=Org1MSP
    - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/fabric/msp
    - CORE_CHAINCODE_EXECUTETIMEOUT=300s
  volumes:
      - ../organizations/peerOrganizations/org1.Distributor.com/peers/peer0.org1.Distributor.com:/etc/hyperledger/fabric
      - peer0.org1.Distributor.com:/var/hyperledger/production
  working_dir: /root
  command: peer node start
  ports:
    - 7051:7051
  networks:
    - test
```

Appendix A.2 Docker-compose file used to launch orderer and peer containers

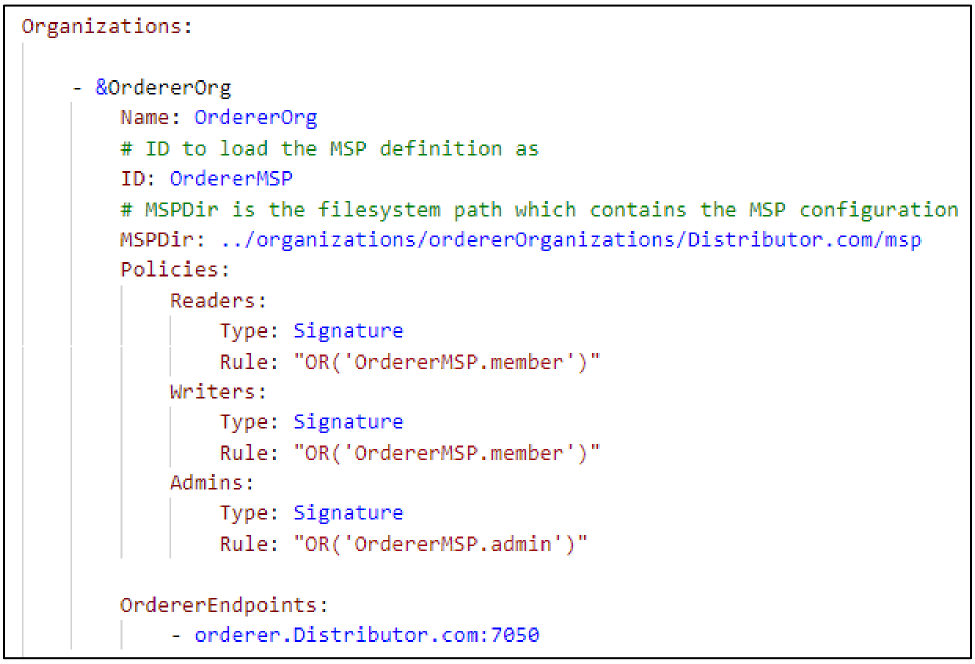
3. After this, we should have our docker containers running as seen below:



cli c33c33b74b7e	hyperledger/fabric-tools:latest	Running	
orderer.Distributor d5a0dea677b5	hyperledger/fabric-orderer:latest	Running	7050:7050 Show all ports (3)
peer0.Org1.Distributor d4dec94f03a0	hyperledger/fabric-peer:latest	Running	7051:7051
peer0.Org2.Supplier1 18a438f9c397	hyperledger/fabric-peer:latest	Running	9051:9051
peer0.Org3.Supplier2 aa18a10a7635	hyperledger/fabric-peer:latest	Running	11051:11051
peer0.Org4.Retailer1 20bdd34c9887	hyperledger/fabric-peer:latest	Running	13051:13051
peer0.Org5.Retailer2 48a954a8ed2f	hyperledger/fabric-peer:latest	Running	15051:15051
peer1.Org1.Distributor 7a19f365f667	hyperledger/fabric-peer:latest	Running	7092:7092
peer1.Org2.Supplier1 947ea769b26d	hyperledger/fabric-peer:latest	Running	9053:9053
peer1.Org3.Supplier2 35f7b39bf592	hyperledger/fabric-peer:latest	Running	11053:11053
peer1.Org4.Retailer1 0792cbe830f5	hyperledger/fabric-peer:latest	Running	13053:13053
peer1.Org5.Retailer2 e53880f1803f	hyperledger/fabric-peer:latest	Running	15053:15053

Appendix A.3 Docker containers running

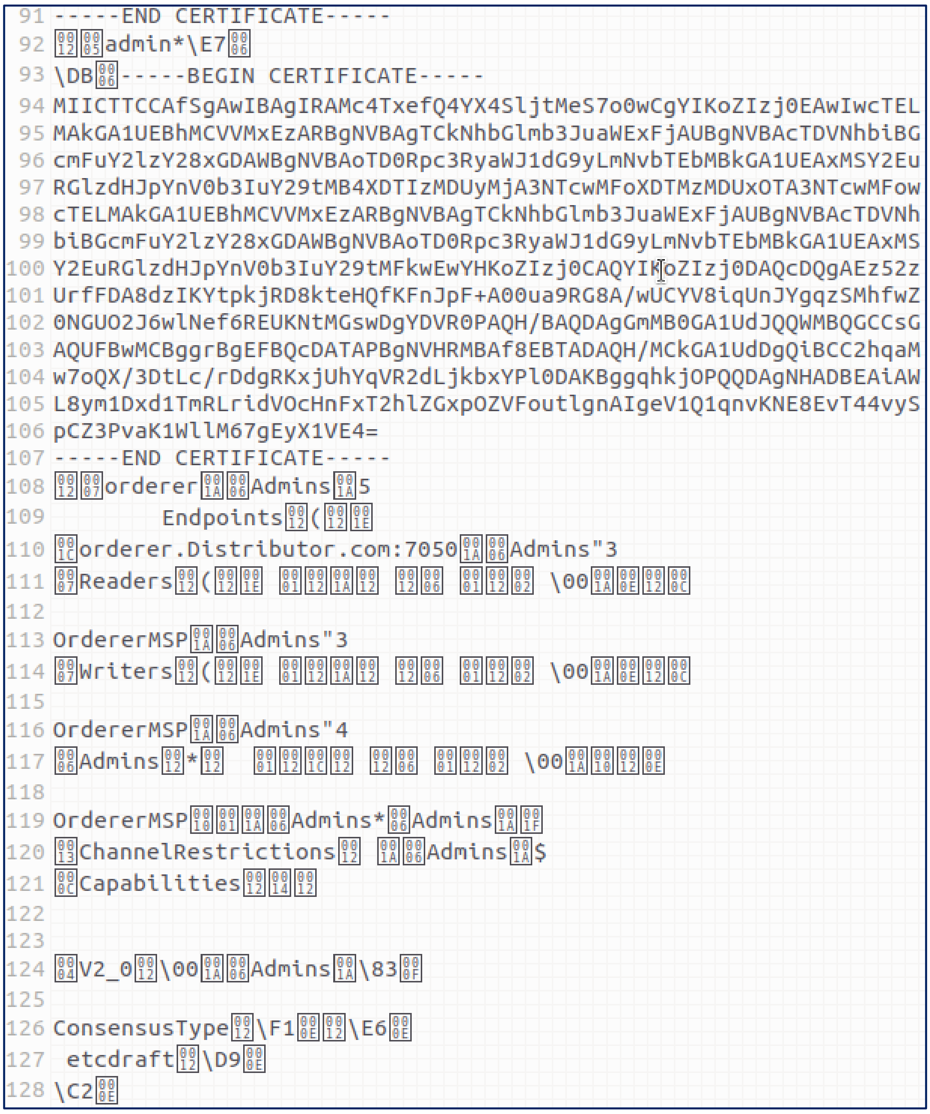
4. Configtx file containing organization members' identities, channel policies, orderer policies, channel profiles, and more.



```
Organizations:

    - &OrdererOrg
        Name: OrdererOrg
        # ID to load the MSP definition as
        ID: OrdererMSP
        # MSPDir is the filesystem path which contains the MSP configuration
        MSPDir: ../organizations/ordererOrganizations/Distributor.com/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Writers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Admins:
                Type: Signature
                Rule: "OR('OrdererMSP.admin')"

        OrdererEndpoints:
            - orderer.Distributor.com:7050
```

Appendix A.4 Configtx file used to manage network policies and create channel

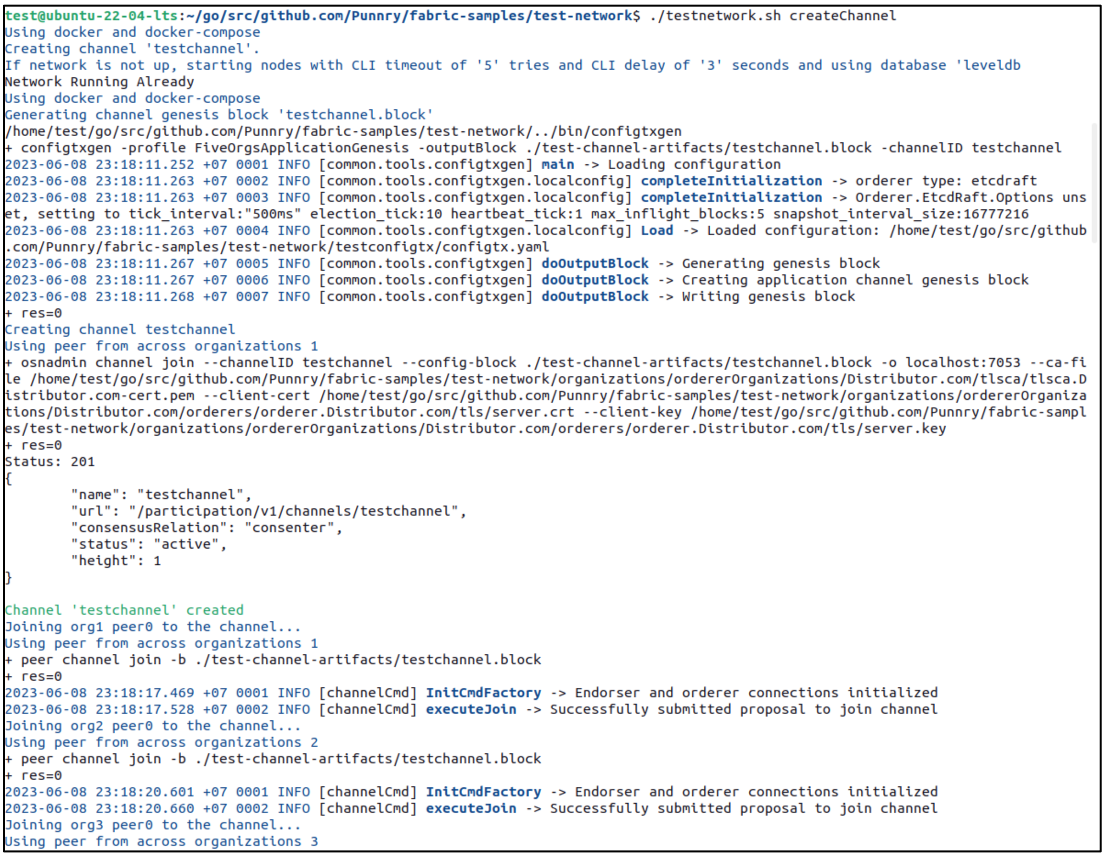
5. Next, we can start creating the channel by using command “configtxgen -profile {PROFILE NAME} -outputBlock {PATH TO FILE} -channelID {CHANNEL NAME}”. This creates a bootstrap file called a genesis block that which the orderer can join in order to initialize the channel operation. This file contains certificates and identity of network members so that any attempt to join the channel from outsiders is immediately rejected.



```
-----END CERTIFICATE-----
admin*\E7
\DB-----BEGIN CERTIFICATE-----
MIICTTCCAfSgAwIBAgIRAMc4TxefQ4YX4SljtMeS7o0wCgYIKoZIzj0EAwIwcTEL
MAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNhbiBG
cmFuY2lzY28xGDAWBgNVBAoTD0Rpc3RyaWJ1dG9yLmNvbTEbMBkGA1UEAxMSY2Eu
RGlzdHJpYnV0b3IuY29tMB4XDTIzMDUyMjA3NTcwMFoXDTMzMDUxOTA3NTcwMFow
cTELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNh
biBGcmFuY2lzY28xGDAWBgNVBAoTD0Rpc3RyaWJ1dG9yLmNvbTEbMBkGA1UEAxMS
Y2EuRGlzdHJpYnV0b3IuY29tMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEz52z
UrfFDA8dzIKYtpkjRD8kteHQfKFnJpF+A00ua9RG8A/wUCYV8iqUnJYgqzSMhfwZ
0NGUO2J6wlNef6REUKNtMGswDgYDVR0PAQH/BAQDAgGmMB0GA1UdJQQWMBQGCCsG
AQUFBwMCBggrBgEFBQcDATAPBgNVHRMBAf8EBTADAQH/MCkGA1UdDgQiBCC2hqaM
w7oQX/3DtLc/rDdgRKxjUhYqVR2dLjkbxYPl0DAKBggqhkjOPQQDAgNHADBEAiAW
L8ym1Dxd1TmRLridVOcHnFxT2hlZGxpOZVFoutlgnAIgeV1Q1qnvKNE8EvT44vyS
pCZ3PvaK1WllM67gEyX1VE4=
-----END CERTIFICATE-----
ordererAdmins5
        Endpoints(
orderer.Distributor.com:7050Admins"3
Readers(     \00

OrdererMSPAdmins"3
Writers(     \00

OrdererMSPAdmins"4
Admins*      \00

OrdererMSPAdmins*Admins
ChannelRestrictions Admins$
Capabilities


V2_0\00Admins\83

ConsensusType\F1\E6
 etcdraft\D9
\C2
```

Appendix A.5 Test channel genesis block

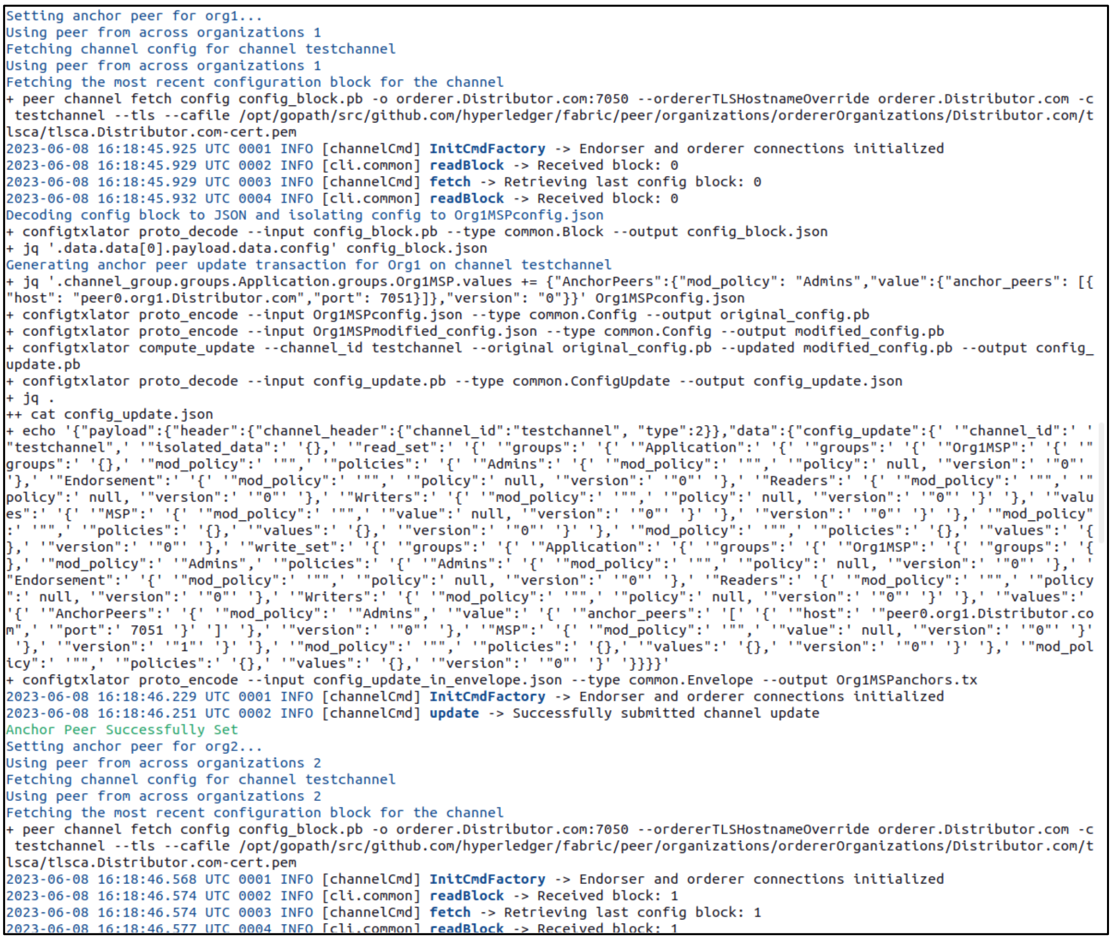
6. Furthermore, we command the orderer to join this channel, using command “osnadmin channel join –channelID {CHANNEL NAME} –config-block {PATH TO GENESIS BLOCK} -o {ORDERER ADDRESS:PORT} –ca-file {ORDERER CERTIFICATE} –client-cert {ORDERER_ADMIN_TLS_SIGN_CERT} –client-key {ORDERER_ADMIN_TLS_PRIVATE_KEY}”. Then each peer node is commanded to join the network one by one using “peer channel join -b {PATH TO GENESIS BLOCK}”. It is noted that each peer possesses unique address and certificate, and thus, the operator utilizes a script in which the address and certificates are automatically entered into the command before execution.



```
test@ubuntu-22-04-lts:~/go/src/github.com/Punnry/fabric-samples/test-network$ ./testnetwork.sh createChannel
Using docker and docker-compose
Creating channel 'testchannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb
Network Running Already
Using docker and docker-compose
Generating channel genesis block 'testchannel.block'
/home/test/go/src/github.com/Punnry/fabric-samples/test-network/../bin/configtxgen
+ configtxgen -profile FiveOrgsApplicationGenesis -outputBlock ./test-channel-artifacts/testchannel.block -channelID testchannel
2023-06-08 23:18:11.252 +07 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-06-08 23:18:11.263 +07 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: etcdraft
2023-06-08 23:18:11.263 +07 0003 INFO [common.tools.configtxgen.localconfig] completeInitialization -> Orderer.EtcdRaft.Options uns
et, setting to tick_interval:"500ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2023-06-08 23:18:11.263 +07 0004 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/test/go/src/github
.com/Punnry/fabric-samples/test-network/testconfigtx/configtx.yaml
2023-06-08 23:18:11.267 +07 0005 INFO [common.tools.configtxgen] doOutputBlock -> Generating genesis block
2023-06-08 23:18:11.267 +07 0006 INFO [common.tools.configtxgen] doOutputBlock -> Creating application channel genesis block
2023-06-08 23:18:11.268 +07 0007 INFO [common.tools.configtxgen] doOutputBlock -> Writing genesis block
+ res=0
Creating channel testchannel
Using peer from across organizations 1
+ osnadmin channel join --channelID testchannel --config-block ./test-channel-artifacts/testchannel.block -o localhost:7053 --ca-fi
le /home/test/go/src/github.com/Punnry/fabric-samples/test-network/organizations/ordererOrganizations/Distributor.com/tlsca/tlsca.D
istributor.com-cert.pem --client-cert /home/test/go/src/github.com/Punnry/fabric-samples/test-network/organizations/ordererOrganiza
tions/Distributor.com/orderers/orderer.Distributor.com/tls/server.crt --client-key /home/test/go/src/github.com/Punnry/fabric-sampl
es/test-network/organizations/ordererOrganizations/Distributor.com/orderers/orderer.Distributor.com/tls/server.key
+ res=0
Status: 201
{
        "name": "testchannel",
        "url": "/participation/v1/channels/testchannel",
        "consensusRelation": "consenter",
        "status": "active",
        "height": 1
}

Channel 'testchannel' created
Joining org1 peer0 to the channel...
Using peer from across organizations 1
+ peer channel join -b ./test-channel-artifacts/testchannel.block
+ res=0
2023-06-08 23:18:17.469 +07 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-06-08 23:18:17.528 +07 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
Joining org2 peer0 to the channel...
Using peer from across organizations 2
+ peer channel join -b ./test-channel-artifacts/testchannel.block
+ res=0
2023-06-08 23:18:20.601 +07 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-06-08 23:18:20.660 +07 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
Joining org3 peer0 to the channel...
Using peer from across organizations 3
```

Appendix A.6 Joining orderer and peers to test channel

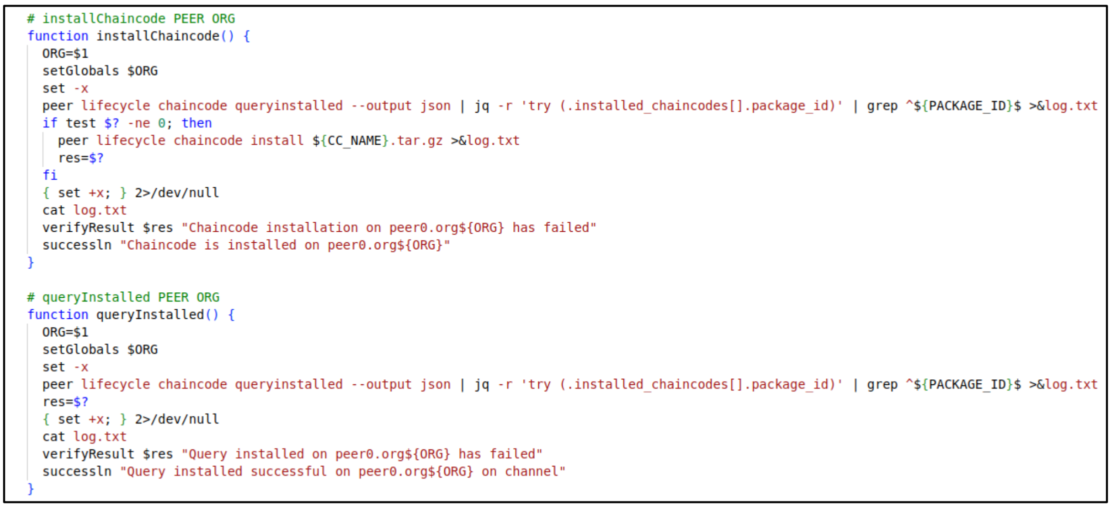
7. The next step is to set anchor peers for each organization so that at least one anchor peer per member can discover and communicate with others of the same kind cross-organization, and within the same channel.



```
Setting anchor peer for org1...
Using peer from across organizations 1
Fetching channel config for channel testchannel
Using peer from across organizations 1
Fetching the most recent configuration block for the channel
+ peer channel fetch config config_block.pb -o orderer.Distributor.com:7050 --ordererTLSHostnameOverride orderer.Distributor.com -c
 testchannel --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/organizations/ordererOrganizations/Distributor.com/t
lsca/tlsca.Distributor.com-cert.pem
2023-06-08 16:18:45.925 UTC 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-06-08 16:18:45.929 UTC 0002 INFO [cli.common] readBlock -> Received block: 0
2023-06-08 16:18:45.929 UTC 0003 INFO [channelCmd] fetch -> Retrieving last config block: 0
2023-06-08 16:18:45.932 UTC 0004 INFO [cli.common] readBlock -> Received block: 0
Decoding config block to JSON and isolating config to Org1MSPconfig.json
+ configtxlator proto_decode --input config_block.pb --type common.Block --output config_block.json
+ jq '.data.data[0].payload.data.config' config_block.json
Generating anchor peer update transaction for Org1 on channel testchannel
+ jq '.channel_group.groups.Application.groups.Org1MSP.values += {"AnchorPeers":{"mod_policy": "Admins","value":{"anchor_peers": [{
"host": "peer0.org1.Distributor.com","port": 7051}]},"version": "0"}}' Org1MSPconfig.json
+ configtxlator proto_encode --input Org1MSPconfig.json --type common.Config --output original_config.pb
+ configtxlator proto_encode --input Org1MSPmodified_config.json --type common.Config --output modified_config.pb
+ configtxlator compute_update --channel_id testchannel --original original_config.pb --updated modified_config.pb --output config_
update.pb
+ configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate --output config_update.json
+ jq .
++ cat config_update.json
+ echo '{"payload":{"header":{"channel_header":{"channel_id":"testchannel", "type":2}},"data":{"config_update":{' '"channel_id":' '
"testchannel",' '"isolated_data":' '{},' '"read_set":' '{' '"groups":' '{' '"Application":' '{' '"groups":' '{' '"Org1MSP":' '{' '"
groups":' '{},' '"mod_policy":' '"",' '"policies":' '{' '"Admins":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"'
'},' '"Endorsement":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '},' '"Readers":' '{' '"mod_policy":' '"",' '"
policy":' null, '"version":' '"0"' '},' '"Writers":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '}' '},' '"valu
es":' '{' '"MSP":' '{' '"mod_policy":' '"",' '"value":' null, '"version":' '"0"' '}' '},' '"version":' '"0"' '}' '},' '"mod_policy"
:' '"",' '"policies":' '{},' '"values":' '{},' '"version":' '"0"' '}' '},' '"mod_policy":' '"",' '"policies":' '{},' '"values":' '{
},' '"version":' '"0"' '},' '"write_set":' '{' '"groups":' '{' '"Application":' '{' '"groups":' '{' '"Org1MSP":' '{' '"groups":' '{
},' '"mod_policy":' '"Admins",' '"policies":' '{' '"Admins":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '},' '
"Endorsement":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '},' '"Readers":' '{' '"mod_policy":' '"",' '"policy
":' null, '"version":' '"0"' '},' '"Writers":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '}' '},' '"values":'
'{' '"AnchorPeers":' '{' '"mod_policy":' '"Admins",' '"value":' '{' '"anchor_peers":' '[' '{' '"host":' '"peer0.org1.Distributor.co
m",' '"port":' 7051 '}' ']' '},' '"version":' '"0"' '},' '"MSP":' '{' '"mod_policy":' '"",' '"value":' null, '"version":' '"0"' '}'
 '},' '"version":' '"1"' '}' '},' '"mod_policy":' '"",' '"policies":' '{},' '"values":' '{},' '"version":' '"0"' '}' '},' '"mod_pol
icy":' '"",' '"policies":' '{},' '"values":' '{},' '"version":' '"0"' '}' '}}}}'
+ configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope --output Org1MSPanchors.tx
2023-06-08 16:18:46.229 UTC 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-06-08 16:18:46.251 UTC 0002 INFO [channelCmd] update -> Successfully submitted channel update
Anchor Peer Successfully Set
Setting anchor peer for org2...
Using peer from across organizations 2
Fetching channel config for channel testchannel
Using peer from across organizations 2
Fetching the most recent configuration block for the channel
+ peer channel fetch config config_block.pb -o orderer.Distributor.com:7050 --ordererTLSHostnameOverride orderer.Distributor.com -c
 testchannel --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/organizations/ordererOrganizations/Distributor.com/t
lsca/tlsca.Distributor.com-cert.pem
2023-06-08 16:18:46.568 UTC 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-06-08 16:18:46.574 UTC 0002 INFO [cli.common] readBlock -> Received block: 1
2023-06-08 16:18:46.574 UTC 0003 INFO [channelCmd] fetch -> Retrieving last config block: 1
2023-06-08 16:18:46.577 UTC 0004 INFO [cli.common] readBlock -> Received block: 1
```

Appendix A.7 Setting anchor peer

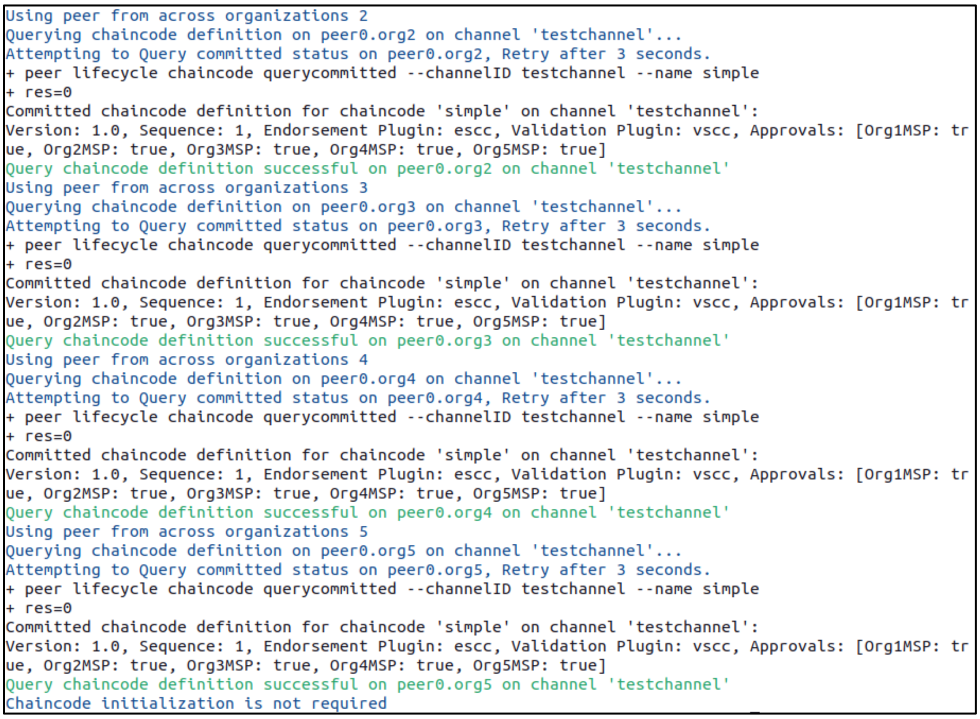
8. Now that the channel is properly set up. A chaincode needs to be installed so that transactions can be invoked. Using a script, the operator may package, install, approve, and commit the chaincode.



```
# installChaincode PEER ORG
function installChaincode() {
  ORG=$1
  setGlobals $ORG
  set -x
  peer lifecycle chaincode queryinstalled --output json | jq -r 'try (.installed_chaincodes[].package_id)' | grep ^${PACKAGE_ID}$ >&log.txt
  if test $? -ne 0; then
    peer lifecycle chaincode install ${CC_NAME}.tar.gz >&log.txt
    res=$?
  fi
  { set +x; } 2>/dev/null
  cat log.txt
  verifyResult $res "Chaincode installation on peer0.org${ORG} has failed"
  successln "Chaincode is installed on peer0.org${ORG}"
}

# queryInstalled PEER ORG
function queryInstalled() {
  ORG=$1
  setGlobals $ORG
  set -x
  peer lifecycle chaincode queryinstalled --output json | jq -r 'try (.installed_chaincodes[].package_id)' | grep ^${PACKAGE_ID}$ >&log.txt
  res=$?
  { set +x; } 2>/dev/null
  cat log.txt
  verifyResult $res "Query installed on peer0.org${ORG} has failed"
  successln "Query installed successful on peer0.org${ORG} on channel"
}
```

Appendix A.8 Chaincode script

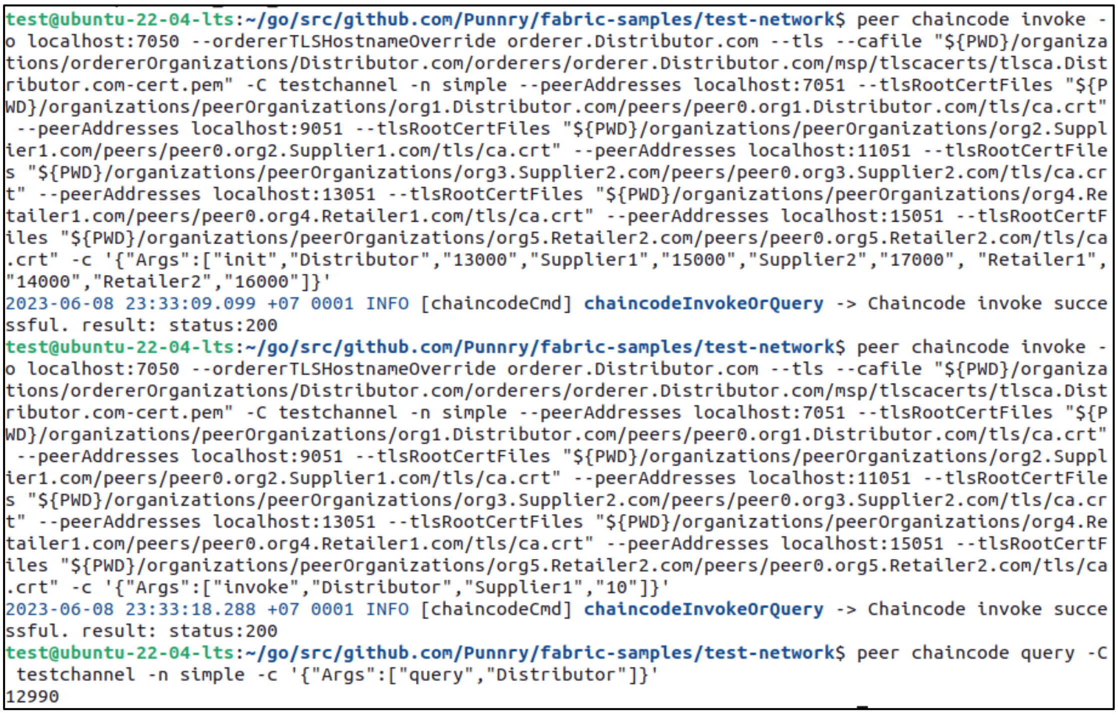
9. Chaincode installation can be done using a pre-written script, or a script provided by the Hyperledger Fabric Sample Network.



```
Using peer from across organizations 2
Querying chaincode definition on peer0.org2 on channel 'testchannel'...
Attempting to Query committed status on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID testchannel --name simple
+ res=0
Committed chaincode definition for chaincode 'simple' on channel 'testchannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: tr
ue, Org2MSP: true, Org3MSP: true, Org4MSP: true, Org5MSP: true]
Query chaincode definition successful on peer0.org2 on channel 'testchannel'
Using peer from across organizations 3
Querying chaincode definition on peer0.org3 on channel 'testchannel'...
Attempting to Query committed status on peer0.org3, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID testchannel --name simple
+ res=0
Committed chaincode definition for chaincode 'simple' on channel 'testchannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: tr
ue, Org2MSP: true, Org3MSP: true, Org4MSP: true, Org5MSP: true]
Query chaincode definition successful on peer0.org3 on channel 'testchannel'
Using peer from across organizations 4
Querying chaincode definition on peer0.org4 on channel 'testchannel'...
Attempting to Query committed status on peer0.org4, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID testchannel --name simple
+ res=0
Committed chaincode definition for chaincode 'simple' on channel 'testchannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: tr
ue, Org2MSP: true, Org3MSP: true, Org4MSP: true, Org5MSP: true]
Query chaincode definition successful on peer0.org4 on channel 'testchannel'
Using peer from across organizations 5
Querying chaincode definition on peer0.org5 on channel 'testchannel'...
Attempting to Query committed status on peer0.org5, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID testchannel --name simple
+ res=0
Committed chaincode definition for chaincode 'simple' on channel 'testchannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: tr
ue, Org2MSP: true, Org3MSP: true, Org4MSP: true, Org5MSP: true]
Query chaincode definition successful on peer0.org5 on channel 'testchannel'
Chaincode initialization is not required
```

Appendix A.9 Chaincode installation

10. Finally, by using command “peer invoke”, user can invoke transactions based on what is written inside the smart contract. Below is an example of a successful transaction invocation in terminal. Then a transaction speed can be obtained in Docker Desktop log of the peer that invoked the transaction, as per shown in figure 5 under chapter 4 of this research.



```
test@ubuntu-22-04-lts:~/go/src/github.com/Punnry/fabric-samples/test-network$ peer chaincode invoke -
o localhost:7050 --ordererTLSHostnameOverride orderer.Distributor.com --tls --cafile "${PWD}/organiza
tions/ordererOrganizations/Distributor.com/orderers/orderer.Distributor.com/msp/tlscacerts/tlsca.Dist
ributor.com-cert.pem" -C testchannel -n simple --peerAddresses localhost:7051 --tlsRootCertFiles "${P
WD}/organizations/peerOrganizations/org1.Distributor.com/peers/peer0.org1.Distributor.com/tls/ca.crt"
 --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.Suppl
ier1.com/peers/peer0.org2.Supplier1.com/tls/ca.crt" --peerAddresses localhost:11051 --tlsRootCertFile
s "${PWD}/organizations/peerOrganizations/org3.Supplier2.com/peers/peer0.org3.Supplier2.com/tls/ca.cr
t" --peerAddresses localhost:13051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org4.Re
tailer1.com/peers/peer0.org4.Retailer1.com/tls/ca.crt" --peerAddresses localhost:15051 --tlsRootCertF
iles "${PWD}/organizations/peerOrganizations/org5.Retailer2.com/peers/peer0.org5.Retailer2.com/tls/ca
.crt" -c '{"Args":["init","Distributor","13000","Supplier1","15000","Supplier2","17000", "Retailer1",
"14000","Retailer2","16000"]}'
2023-06-08 23:33:09.099 +07 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke succe
ssful. result: status:200
test@ubuntu-22-04-lts:~/go/src/github.com/Punnry/fabric-samples/test-network$ peer chaincode invoke -
o localhost:7050 --ordererTLSHostnameOverride orderer.Distributor.com --tls --cafile "${PWD}/organiza
tions/ordererOrganizations/Distributor.com/orderers/orderer.Distributor.com/msp/tlscacerts/tlsca.Dist
ributor.com-cert.pem" -C testchannel -n simple --peerAddresses localhost:7051 --tlsRootCertFiles "${P
WD}/organizations/peerOrganizations/org1.Distributor.com/peers/peer0.org1.Distributor.com/tls/ca.crt"
 --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.Suppl
ier1.com/peers/peer0.org2.Supplier1.com/tls/ca.crt" --peerAddresses localhost:11051 --tlsRootCertFile
s "${PWD}/organizations/peerOrganizations/org3.Supplier2.com/peers/peer0.org3.Supplier2.com/tls/ca.cr
t" --peerAddresses localhost:13051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org4.Re
tailer1.com/peers/peer0.org4.Retailer1.com/tls/ca.crt" --peerAddresses localhost:15051 --tlsRootCertF
iles "${PWD}/organizations/peerOrganizations/org5.Retailer2.com/peers/peer0.org5.Retailer2.com/tls/ca
.crt" -c '{"Args":["invoke","Distributor","Supplier1","10"]}'
2023-06-08 23:33:18.288 +07 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke succe
ssful. result: status:200
test@ubuntu-22-04-lts:~/go/src/github.com/Punnry/fabric-samples/test-network$ peer chaincode query -C
 testchannel -n simple -c '{"Args":["query","Distributor"]}'
12990
```

Appendix A.10 Transaction invocation

AWARDS OR GRANTS	2018	Awarded a Fully-Funded Master Degree Program in Logistics and Supply Chain Management (Burapha University) under Her Royal Highness Pricess Maha Chakri Sirindhorn Education Project in Cambodia 2018.
	2012	Awarded a Fully-Funded Bachelor Degree Program in Economics (University of Cambodia) by Dr. Haruhisa Handa, President and Founder of International Foundation for Arts and Culture, Japan.