Visualizing Provenance In A Supply chain Using Ethereum Blockchain

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Parastoo Veisi

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Visualization is a widely used in different fields of studies such as supply chain management when there is a need to communicate information to general users. However, there are multiple limitations and problems with visualizing information within traditional systems. In traditional systems, data is in control of one single authority; so data is mutable and there is no guarantee that system administer does not change the data to achieve a desired result. Besides, such systems are not transparent and users do not have any access to the data flow. In this thesis, the main goal was to visualize information that has been saved on top of a new technology named blockchain to overcome the aforementioned problems. All the records in the system are saved on the blockchain and data is pulled out from blockchain to be used in visualization. To have a better insight, a review has been done on relevant studies about blockchain, supply chain and visualization. After identifying the gap in literature review, an architecture was proposed that was used in the implementation. The implementation contains, a system on top of ethereum blockchain and front-end which allows users to interact with the system. In the system, all the information about products and all the transactions that ever happened in the system, are recorded on the blockchain. Then, data was retrieved from the blockchain and used to visualize provenance of products on Google Map API. After implementing the system, the performance was evaluated to make sure that it can handle different situations where various number of clients sending request to the system simultaneously. The performance was as expected in which system responds longer when number of clients sending requests were growing. The proposed solution fill the gap that was identified in the literature review. By adding provenance visualization users can explore previous owners and locations of a product in a trustable manner. Future research can focus on analysis of data which will allow organizations to make informed decisions on choosing popular products to sell.

Acknowledgements

I would love to express my profound appreciation and sincere thanks to Dr. Ralph Deters, who has supported me during the past two years patiently. His continuous guidance and assistance helped me to pursue my masters degree. genuine thanks to to my family, friends, and everyone who has taught me something along the way.

To my dear husband, my lovely parents and my encouraging brother,
for their unconditional love and continuous support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Data is constantly generated, and Internet users are becoming more concerned about collecting data through third parties. Indeed, there is an increasing need for more visibility and transparency in the process of monetizing data. This becomes a significant concern when it comes to supply chain, especially in the food supply chain. In supply chain management traceability and transparency are essential factors, and the consumers are always eager to follow the process of moving a commodity or service from a supplier to a destination.

Revealing product provenance can be a great choice to increase visibility in a supply chain. However, the accuracy of information is always in doubt since the majority of commonly used supply chain management systems are controlled by a central computer, which causes serious problems.

Traditional supply chain management systems are always open to getting attacked since the central server which is the single point of failure, exists in the network and leading to manipulation of data. Especially in a supply chain, accurate data plays an important role.

Another problem with the traditional supply chain is the fraud and fake data that cannot be prevented. There are different parties involved in the supply chain, and there is a need to make all the records visible to deter fraud.

The limited transparency is another issue where users do not have any access to data. In a supply chain, transparency and traceability are two crucial factors that were missed out in traditional systems. Also, consumers have become more concerned about the origin of the products they are buying.

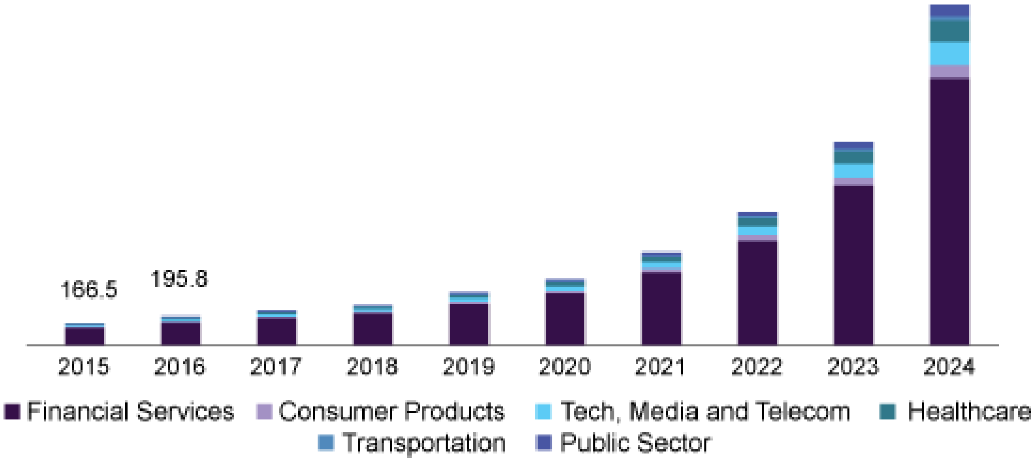


Figure 1.1: Blockchain Technology Market Size [1]

In 2008 a novel idea was published in a paper called Bitcoin: Peer-to-PeerElectronic Cash System[2] that changed lots of perspectives by establishing the first cryptocurrency. The technology behind the Bitcoin network was the most appealing to researchers because financial transactions could be made without any middle intermediary. After proposing Blockchain 1.0, which was introduced by Satoshi Nakamoto for exchanging cryptocurrency, Blockchain 2.0 initiated Smart Contracts to create small, autonomous computer programs to be executed on the blockchain. Blockchain 3.0 introduced the decentralized application, which combined front-end and contracts. Currently, Blockchain 4.0 is mainly used to address business demands in different fields. The technology continues to evolve globally, and designing a secure, scalable, transparent, and robust network is highly needed. Distributed ledger technology is an option to overcome critical deficiencies caused by customary supply chain networks mentioned earlier in this chapter.

Distributed ledger technology can prevent data manipulation in the supply chain by providing append-only structure. The inedible ledger is shared among all participants and cannot be altered since all the records are permanently saved without any paperwork.

When the distributed ledger is used in the supply chain, parties can confirm authenticity by tracking the creation and modification of a transaction, which decreases the risk of falsifying significantly. The end-to-end visibility into digital records creates a fraud less system.

Transparency has been taken care of effectively in distributed ledger technology. With technology, the consumer has this ability to track the life cycle of a product. Moreover, the ledger would benefit supply chain management by eliminating the middleman because there is no central point to own and control data anymore. Besides, data is shared easily over the network, and products' journey can be easily followed, which results in more transparency and traceability.

In this thesis, we tried to adopt the blockchain technology as a distributed decentralized network to provide a visualization based on accurate information of a product life path. We proposed a solution to visualize provenance of products in the supply chain to give the users a better insight of products life path. To visualize the life cycle of a product, organizations need to exchange the product. So, blockchain technology is used to give the organizations an ability to exchange a product without any intermediary. Also, blockchain helped to record all the information permanently and retrieve the information to be used in the visualization.

1.1 Problem Definition

Every day, billions of products are being manufactured and delivered all over the world. Meanwhile, the products raw materials are not produced in a single company, and commonly, the components are originated from different manufacturers. Indeed, in a process called supply chain, many activities are involved in moving a product from suppliers to consumers. In the supply chain, multiple parties, including transporters, retailers, distributors, storage facilities, and suppliers, through a network, are involved in completing the chain. All the parties in the network are controlled by a main or central system.

However, as the product authenticity verification has always been the main concern for consumers, the need to replace traditional client server architectures has been raised due to the following reasons:

- Documenting records causes errors, delays, and conflicting information in the tracking of shipments.

- Only one main server approves all the incoming requests which result in low performance and high cost of the server maintenance

- The system is vulnerable and can be taken down quickly because the central authority is the single point of failure and attack.

- Transparency and traceability are crucial issues with traditional networks as users do not own or control any data.

- The information provided by central authorities cannot be trusted easily because there is no way to prove that data has not been manipulated.

To overcome these problems and increase transparency and visibility there are various studies [3, 4, 5, 6] which have tried to eliminate the need of central authority in the supply chain by using the blockchain technology. However, in these studies, information visualization has been missed out.

Charts and graphs always communicate better with people with general knowledge because they can understand visuals better than written language. Since visualization increases productivity, it has been used widely in different fields. Specifically, in the supply chain, visualization ensures product safety and help customers to have a better insight into the products life cycle. Also, it allows customers to explore and discover relationships.

This work proposes a system that contains three main entity types; Users, the Smart Contract, and the Blockchain Network. We tried to increase supply chain visibility by tracking the changes in a product life cycle and visualizing from origin to destination within a map. As visibility and data accuracy are two essential factors in the supply chain, we combined these two factors in our system. When the system is on top of the blockchain, data is permanent, immutable and cannot be altered by anyone in the network; so we guaranteed the data accuracy. Besides, by providing an appealing visualization to the user, we converted written information to map, to increase the visibility and transparency of the system. The blockchain-based smart contracts, which are automated programs, are used in the proposed system to be executed on each node on the network. Therefore, the involved parties in the supply chain may have direct interactions with one another, and trust is not required anymore.

Such secured direct connections are leading to more transparency, visibility, efficiency, and also less cost and risk of failure in the shipment tracking process.

In this research, we tried to answer the following questions:

1. How data can be retrieved from blockchain to be used in visualizing provenance?

2. How to represent supply chain provenance to the general users?

By answering these two questions, the main purpose of this research which is providing a visualization with reliable data will be addressed. Data is the most important aspect of any visualization especially when it comes to supply chain. Providing a visualization with reliable data also can help with making the system transparent which at the end customers as the last entity in the supply chain can benefit from it.

The remaining section of this thesis is organized as follows:

- Chapter 2 -Literature Review reviews the related research and technologies
- Chapter 3 -Architecture explains the proposed architecture and presents the implementation of the architecture.
- Chapter 4 -Experiment tests the performance of the proposed architecture and discusses the result.
- Chapter 5 -Conclusion and Future Work describes the outline of the contributions and next steps of this research.

Chapter 2

Literature Review

This chapter presents important literature and technologies including Supply Chain Management, visualization, and its importance in supply chain management, blockchain, and its components, features, and applications. Essential technologies to interact with blockchain is another part of this chapter. Also, a review on supply chain tracking systems and related research involving the blockchain technology is provided. The literature review helps to overcome the challenges described in the previous chapter.

- Supply Chain Management
- Visualization
- Blockchain
  - Blockchain application
  - Public & Private Blockchain
  - Distributed Consensus
  - Ethereum
- Technologies To Interact With Blockchain
  - Rinkeby Network
  - Next Server
  - React.js
  - Web3
  - MetaMask
- Blockchain and supply chain

2.1 Supply Chain Management

Different definitions have been proposed for the terms supply chain and supply chain management. [7]defines the supply chain as the direct involvement of three or more entities in the whole flow of products, finances, services and information from the origin to the end, and the supply chain management as the functions required to manage the flows, aiming to improve the whole supply chain and long-term performance of individual entities.

In the last few years, significant emphasis has been giving to traceability and transparency as they are essential factors in supply chain management. Traceability is a significant factor since customers are eager to follow the physical location of the product in any stage of production. Specifically, in the food supply chain, traceability plays an important role. Food supply chain provenance is critical today. According to [8], 600 million which is 1 in 10 people fall ill and among this 600 million, 420,000 of them die every year due to eating contaminated food. Also, in the last four years there is 20% increase in the total cost caused by foodborne illness , which cause $55.5B to $93.2B loss each year in the U.S. alone.

For instance, not using fresh and untrustworthy products, harmed McDonald's reputation, which is the largest fast-food provider. The company tried not to be idle and compensate for the negative criticisms by reviewing its supply chain practices and making them more transparent. However, it takes time a long time to change their negative reputation in people's minds and make them trust the company again[9]. In [10], the author tries to explain the importance of traceability. He mentions six important elements of traceability including:

- product traceability; refers to the physical location of the product
- process traceability; refers to how, where and when a product has been produced
- genetic traceability; refers to the genetic composition of the product
- inputs traceability; refers to origin and type of food products
- disease and pest traceability; refers to how bacteria and viruses have been driven from raw products
- measurement traceability; refers to comparing individual measurement results to accepted reference standards through an unbroken chain of calibrations.

These six elements give customers complete oversight of a life history of food products utilizing records. Traceability in [11] is defined as the ability of a chain in all steps to identify and verify the components and chronology of events. They also discuss that transparency and traceability are related but not in a linear and straightforward fashion. When a network contains few participants, higher levels of traceability do not make the system transparent. On the other hand, transparent information may increase traceability. Indeed, there is a need for big companies to adopt technologies in order to meet consumers demands and keep them satisfied. To achieve safety and quality and keep consumers satisfied, participants in food supply rely on two methodologies. One of the methodologies is when standards or certifications are mainly responsible for managing the supply chain. The second one relies on food traceability system that provides transparent trace back and tracks forward information[12].

Traceability systems can be centralized, linear, or distributed approach.

- In the centralized approach, shared databases are used for collecting data.
- In the linear approach, each partner records the supplier and customer of specific products, and traceable data is passed from one partner to another. It is also referred to as the one step forward and one step back principle.
- In a distributed approach, the partners in the food supply chains create a traceability system to exchange traceability data[13]

In [14], a distributed architecture was designed to be used throughout a food supply chain to monitor, control, plan, and optimize processes remotely via the Internet-based on virtual objects. [15] is another study that tried to track the products using RFID in wire bond station. They proposed a system with a capability of the real-time monitor and tracked the delay of work-in-process, the qualification lots, the rejection of lots of semi-product and the maintenance of wire bond machines. [16] describes that the efficiency of a traceability system relies on the ability to track and trace products, which requires continuous monitoring from primary production until final disposal by the consumer. [17] tries to increase consumer confidence in beef products by introducing RFID based framework. The system identifies all the aspects of beef traceability including identification of individual cattle, and biometric identifiers for verification of cattle identity from farm to slaughter to decrease the risk of fraudulent activities. In [18], researchers tried to attach Flexible Tag Data logger (FTD) to the bottle of wines to trace the containers and to monitor environmental parameters like temperature, humidity, and light intensity. Data is stored in a compact format which can be read with a Smartphone or PDA with integrated infrared port to evaluate the safety of the bottles. [19] developed an automated system with RFID smart tag, which can help with real-time traceability and cold chain monitoring for food applications.

Although customers are the endpoint in a supply chain, the researchers mentioned above have not pointed out how they present information to the involved parties in the chain. Information visualization can be a tool to make users more engaged in the process of producing a product.

2.2 Visualization

Data visualization dates back to the pre-17th century, which was used in early maps and diagrams. So, data visualization is not a modern statical development[20]. With the help of visualization, we may be able to explore and discover relationships, convey end-result, and increase audience interpretation. As Arthur Brisbane once said: ”Use a picture. It is worth a thousand words”, use of pictures, graphs, charts, etc. provide a nice insight into the information, especially for general Internet users.

Visuals can be a way of communication to convey information. Accurately delivering content has always been a challenge in different disciplines like science, environment, and medicine. For example, psychologists[21] have done an experiment on how viewers comprehend multivariate data presented in graphs to investigate the effect of format of the graph, familiarity of viewers with variables, and graphical literacy skills of viewers on the comprehension. In public health[22], researchers use data visualization to examine cultural data representations. To communicate cardiovascular risk[23] in 73% of 70 consultations, verbal qualifiers were used, 11% numerical and 16% visual formats. According to findings, a patient subjective understanding was significantly higher in visual counseling compared to verbal counseling. Also, [24] discusses that graphical displays can help to summarize and reveal patterns in data. Moreover, since data is displayed in concrete and visual terms, it can attract and hold peoples attention.

Data visualization also is an essential tool in the supply chain since it can support experts and non-experts in collaboration and decision making. Today, spreadsheets are mainly used to keep track of records in companies. However, when a supply chain grows, dealing with massive amount of raw data in spreadsheets cannot give the manager a proper insight as it needs a lot of effort and time. On top of that, spreadsheets are vulnerable and difficult to maintain; so they cannot be the safest tool to keep records. Visual tools can make a difference by changing raw data into insight because they are easier to understand and attract people's attention better. Visualization also helps with increasing visibility and transparency in the supply chain.

According to [25], lack of visibility has been reported in 75 percent of supply chains in business organizations. Also, most companies have little information on second and third-tier suppliers; so it can lower the efficiency of the network. However, supply chain visibility provides speed, reliability, and flexibility to gain a competitive advantage in the form of well-controlled and managed supply chain functions[26].

Business owners use supply chain modeling technology to create baseline models of their network. Once have a living model, it can be visualized in different ways such as maps or charts and graphs. Interactive dashboards can be another choice to help answer specific questions or outliers in the model[26].

Researchers at the University of Stanford introduced SCVisualizer (Supply Chain Visualizer), an information visualization tool using Web Services and computer for the rapid and seamless generation of a virtual supply chain in construction. In their study, real-time information is available at any time to increase information transparency and identify potential risk. To make the system more secure, in the SCVisualizer, information retrieval is from heterogeneous systems, and it is intractable for practitioners[27].

In [28], the author tries to discuss that sharing of useful and meaningful information among different participants within the supply chain is the key requirement to achieve high visibility. He also points out the benefits of information visibility in the supply chain:

- improved responsiveness
- improved planning and replenishment capabilities
- improved decision making
- improved quality of products

Another study which focuses on creating a tool to visualize supply chain information is [29]. In the study, they developed Supply Chain Visualizer, a visualization tool to analyze RFID event data to increase visibility using both automated analysis techniques and human effort. The map-based visualization of product-flow data in their work can detect inconsistencies. Their platform also allows users to explore the data by categorical filtering capabilities, performance metrics, and activity indices.

RiskVis is another supply chain visualization platform which introduced for risk management and real-time monitoring. The main goal of their work was to provide a live model of the supply chain network of an organization, a platform for risk analysis, and evaluating the vulnerability of the existing supply chain network. Three main features of RiskVis are listed below:

- better visibility of the entire supply chain and its happenings
- Scenario analysis to prepare the plan for supply chain risk mitigation
- rapid responses to unplanned events

According to [30] revealing various stages of production of a product can help customers and organization in making a better decision. So, researchers designed Sourcemap, which is a collaborative database which developed in MIT Media Lab research project. This supply chain mapping platform analyzes the environmental and social impact of products to make the environment sustainable. In the proposed system, they have used visualization techniques to give a clear picture of the entire life-cycle of a product in their third-party network.

Although visualization is a helpful tool for better comprehension, all previously mentioned studies use visualization over centralized systems. These visualizations can not be trustworthy, and the risk of fraud is high because one single authority has control of data; so they can manipulate data as they wish. One possible solution is to retrieve data from the blockchain to eliminate the middleman and reduce the risk of manipulation of data.

2.3 Blockchain

The blockchain is a distributed peer-to-peer network which was created in 2009 in a cryptocurrency-based protocol for the exchange of digital currency called Bitcoins. In 2008, a paper Bitcoin: A Peer-to-Peer Electronic Cash System, written by Satoshi Nakamoto [2] was a novel approach of transferring funds in the form of Bitcoin which is purely peer-to-peer version of electronic cash." Satoshi Nakamoto proposed a particular type of data structure used in some distributed ledgers which can transfer and store data in packages called blocks. In the paper, he explains how online payments can be made in the Bitcoin network by eliminating the need for third parties.

There are lots of applications that previously run through a trusted intermediary. Those applications can now operate in a decentralized fashion and achieve the same functionality with the same amount of certainty without any central authority. The figure 2.1 gives an overall overview of blockchain. In the blockchain network, everyone needs a pair of public and private keys that allow them to join and interact with the rest of the network. Private keys let users sign transactions which can be deciphered with public keys. Using asymmetric cryptography increases authentication, integrity, and non-repudiation of the network[31]. Each signed transactions will be broadcast to all the parties of the network, and before adding the block to the chain, other peers make sure that the transaction is valid. Otherwise, if the peers do not approve the validation of the block, it will be discarded. As a new block formation requires a certain number of transaction approvements, if the specified number of nodes approve a transaction, a new block would be attached to the chain. A block contains all state changes of a transaction and a hash value. Using the hash value is a way to prevent modifying the data. Also, the new block's hash is created by using the previous block's hash. The blockchain is immutable, and after adding a block to the chain, it cannot be changed even by system administer which makes the blockchain differ from other files or databases

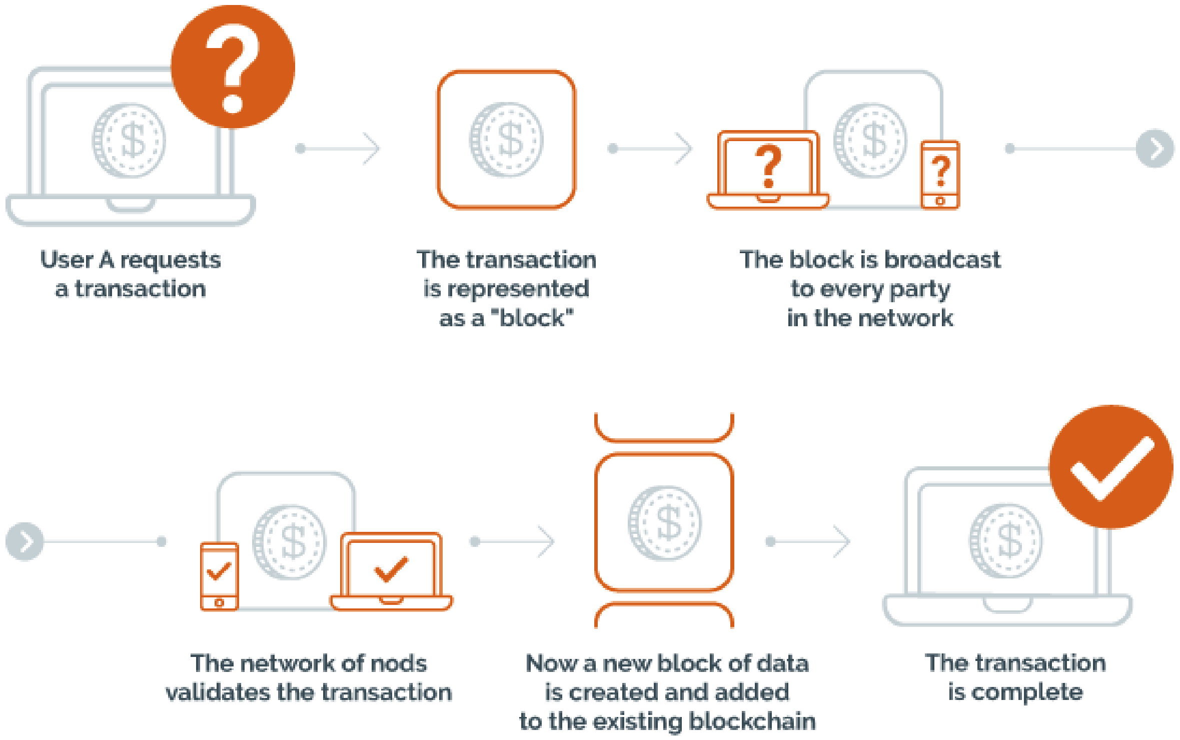


Figure 2.1: How Blockchain Works [32]

The blockchain adopts distributed ledger technology which records transactions of parties and stores them in multiple locations. The distributed ledger technology is an approach to record, share, and synchronize data across a decentralized network of computer systems with different degrees of control of the ledger. Moreover, the distributed database can be the implementation of choice for a distributed ledger[33]. In the decentralized systems:

- Users are able to control their data
- As the system does not rely on a single node, it does not go down.
- As the system does not have a single point of control, third party risks would be removed, and attacks would be expensive for malicious entities

2.3.1 Blockchain Applications

- Cargill Agriculture Promotion Inc. tried to take advantage of the blockchain technology to allow consumers to trace the turkey from its origin farm to the store they bought it from. They used the technology to make the whole process paperless and create immutable data records which nobody can change or remove them[34].
- In order to using metadata to detect suspicious transactions in Smart Living Spaces, [35] proposes the idea of adopting blockchain features into software-defined components to host them on edge devices. The metadata accounts proposed in this research, work more dynamically compared to traditional smart contracts.
- Coca-Cola is another company, tries to employ the blockchain. According to KnowTheChain(KTC), a partnership founded by U.S.-based Humanity United, 20 large companies including Coca-Cola, fall short to fight against forced labor, especially in regions with enormous stakes. After this study conducted, the company aimed to employ the blockchain technology to increase transparency and create a secure record-keeping for workers. The company committed to deal with forced labor, child labor, and land rights by using blockchain technology at country level by 2020[36].
- In an attempt to offering blockchain services for IoT, [37] evaluates the cloud (IBM Bluemix) vs the fog layer (multichain) to host blockchain in order to address the challenge of hosting blockchain when it comes to deployment.

- Carrefour is a European company that uses blockchain for the first time in France to trace chickens. The company uses QR Code to let consumers scan the products label using their smartphones[38].

- In [39] authors try to make advantage of blockchain and artificial intelligence (AI) to make things autonomous. The AI features are implemented using CLIPS programming language. The communication between the smart things is implemented using Multichain blockchain hosted on a Fog network.

- The first company that reveals everything about a beer is Downstrem[40] beer which used blockchain for the first time. The company had this idea to bring integrity to the beer industry, and they knew the blockchain technology could help to achieve this goal. The beers have QR smart tags which can be used to identify every bottle and show customers the bottle number and more content such as ingredients and brewing methods. Every little detail is recorded to the blockchain as the guarantee of transparency and authenticity.

- Authors in [41] tried to use blockchain to implement a distributed data access control management. In this work, the owners of the data are allowed to manage and administer access to their data instead of having a traditional central administration authority that grant access to data.

- Paddock to the plate is another research project which uses BeefLedger as its technology platform to track beef from pasture to the plate. Everyone including consumer, farmer, the butcher can access the whole history of the meat by scanning QR Code. Researchers aimed to protect Australian brand and industry as it has a strong reputation around the world. Also, consumers can be confident that they buy the top-quality Australian beef[42].

- By using blockchain, [43] manage the communication between the virtualized IoT resources. This paper proposes the novel idea of virtualizing IoT resources to build distributed systems and data views hosted towards the cloud, fog, and dew computing to get the most benefit from each network and avoid high costs regarding latency and bandwidth consumption.

- Grass Roots Farmers Cooperative [44] is a company that tries to reveal everything one wants to know about the final meat product.

- In order to resolve access conflicts, [45] provides proper views of data to different users depending on the permissions they have by integrating blockchain technology

- In [46] authors tried to combine blockchain and off-blockchain storage to construct a personal data management platform to ensure that users control and own their personal data. In the platform, users have complete transparency over collected data. Also, access-control policies would be stored on a blockchain where just the user is able to change them. So, with mobile applications, users are not required to grant a set of permission indefinitely.

- MedRec, a decentralized record management system, providing an immutable log to its patients and full access to their medical information with the help of blockchain technology. MedRec is able to manage authentication, confidentiality, accountability efficiently, and in sharing data, it can handle sensitive information by granting special access to the users. In their system, they encourage researchers, health authorities, etc. to participate in the network as blockchain miners to make the system sustain and secure via Proof of Work[47].

- The US startup, based on the Ethereum Blockchain technology launched the Gem Health Network. The system creates a global data which is available for different healthcare specialists through the network without compromising privacy and security. Their approach introduces a new class of Blockchain-based applications in healthcare that will unlock wasted resources and solve important operational problems[48].

- Amatista[49] is a research that proposing the idea of implementing a zero-trust management in IoT by using blockchain-based middleware to manage IoT networks. Amatista implements a hierarchical mining model to avoid trusting either the constrained infrastructure or transactions.

- Healthbank is a Swiss startup which introduces an innovative way to put data into the hands of patients. The next step for them is to apply blockchain to the platform to let patients create a medical history throughout their life and share it easily with doctors. The data can be collected from smart devices such as wearable gadgets. As a result, doctors will gain access to lots of information about the patient, including heart rate, blood pressure, taken medicines, sleep patterns, eating habits, physician visits, etc[50].

The table 2.1 also shows examples of blockchain applications and platforms.

Cryptocurrencies
Bitcoin https://bitcoin.org/
Peercoin http://peercoin.net/
Colouredcoins http://coloredcoins.org/
Omni http://www.omnilayer.org/
Nxt http://www.omnilayer.org/

Smart contract platforms
Etheruem https://www.ethereum.org/
Counterparty http://counterparty.io/

Ledger platforms
Factom http://factom.org/
Ripple https://ripple.com/
Eris https://erisindustries.com/
MultiChain http://www.multichain.com/
Enigma http://enigma.media.mit.edu

Table 2.1: Blockchain Applications

2.3.2 Public Blockchain

In public blockchain, as there are no central owners or any privileged users, every participant can join the network to access all information on blockchain and validate new transactions[47]. In other words, since public blockchain is open, all records are visible to any user who wants to interact with the network. However, one drawback of the public blockchain is that when the chain gets bigger, requiring more time and energy, as the nodes need, to be synchronized in the blockchain [51]. Such network permits the unidentified network users to be motivated and increase the accuracy of the ledger[52]. In [53], the author tries to explain why public blockchain is required for health data. In their work, public blockchain has been used as an access control manager to health records since everyone can join the system and store their health data.

2.3.3 Private Blockchain

A private blockchain is considered as a centralized network when it is managed by one pre-selected organization that controls the network. In private blockchain, only those nodes coming from one specific organization would be allowed to join the consensus process, and others requiring a permission to access the network and add records to the ledger[54]. In other words, participants without an invitation or permission cannot join the network. Such ability makes the network more fitted into regulatory frameworks and institutional arrangements. Private networks also work better in handling issues related to identity verification and data privacy[52]. However, as permissions are held centralized, it sacrifices the decentralized concept [51].

An example of a permissioned blockchain framework is Hyperledger Fabric. Hyperledger Fabric enables participants using chaincodes, endorsing peers, and ordering service to manage transactions. Chaincode is a piece of code deploys on the Hyperledger Fabric Network where the endorsing peers, those who maintain the ledger, are responsible for executing and validating it. The ordering service is another component of the network which creates blocks for the distributed ledger and maintains the order where each block is appended to the ledger.

In [55], authors describe a Byzantine Fault Tolerance ordering system for Hyperledger Fabric version 1.0. Hyperledger Fabric version 1.0 only supports crash tolerance through an ordering service based on Apache Kafka. According to this paper, Hyperledger Fabric with Byzantine Fault Tolerance ordering service can achieve up to 10k representative transactions per second. It can also write a transaction on the blockchain in half a second, even if the consensus nodes spread through different continents. Ripple is another example of a permissioned blockchain which determines roles for a specific number of participants to be as transaction validators on their network.

2.3.4 Distributed Consensus

An efficient consensus mechanism is needed before each transactions being incorporated into the blockchain to make sure that individuals would have the same copy of the ledger. Nodes have this ability to decide to commit a transaction to the ledger when a new transaction broadcasts to the network. When most of the nodes agree on single data value, the consensus is achieved[56]. The consensus foundation is an excellent advantage of the technology that keeps the network secure where any nodes cannot be trusted.

In the blockchain, there are four main consensus algorithms such as the Proof-of-Work (PoW), the Practical Byzantine Fault Tolerance (PBFT), Proof-of-Stake (PoS), and the Delegated Proof-of-Stake (DPoS). Such mathematical algorithm mechanisms in the blockchain technology make the whole network secure, as they make sure the ledger is consistent across the network.

Practical Byzantine Fault Tolerance algorithm (PBFT)

As there is no authority in the blockchain, some users send inaccurate or misleading information about transactions and make the blockchain unreliable, which is called Byzantine faults. Therefore, the blockchain system may encounter loss, damage, repetition of information, and so always, there is a need to protect the system. One potential solution to address such problems is PBFT [56].

PBFT is an algorithm that provides a solution to the Byzantine Generals[57] Problem. In this algorithm, a new block is determined in a round. According to some rules, a primary node would be selected in each round, which is responsible for ordering the transaction. The algorithm is a three-phase protocol could be divided into three phases: pre-prepared, prepared, and commit. In each step, if each node receives votes from over 2/3 of all nodes, the node would enter the next phase. So, all the nodes should be known to the network in this algorithm[54].

Proof of Work

Proof of Work (PoW) was introduced by Cynthia Dwork and Moni Naor back in 1993 before the existence of Bitcoin[58]. In this algorithm, miners solve complex mathematical problems to create blocks and add them to the blockchain. The primary goal of Proof of work is to prevent cyber-attacks which try to take the system down by sending multiple fake requests. In PoW a reward is given to first participating node, first miner, who tries to validate transactions and produce blocks, to find a solution for a mathematical puzzle known as the proof-of-work problem[59]. In this algorithm, miners need to use a significant amount of computing power, which results in a huge amount of energy costs. Conversely, these energy costs defend the system since utilizing a lot of computational power and time make it worthless to attack the system[60].

Proof of Stake

The idea of Proof of Stake (PoS) first mentioned in 2011 on the bitcointalk forum, but the first digital currencies used this method in 2012. In this algorithm, there are no miners, but validators. Validators are required to own ethers and use the ethers to validate a block. The advantage of this is that if a malicious activity happens, they lose their own ether, their stake on the network[61]. Also, according to their stakes, the difficulty level of the crypto-puzzle will be determined.

The algorithm in PoS is different from PoW since the chance to win a mining competition to validate a transaction is based on their wealth or stake in the network. So, validators can collaborate instead of competing because if everyone can reach consensus faster, they can complete more transactions which result in a higher profit[61]. Unlike the Pow, PoS requires mild cost and computing power on mining competition, which makes the system more efficient[62].

Proof of Authority

Proof of Authority(PoA) is considered a new family member of BFT algorithms, which is mainly used in permissioned blockchain where all the participants in the network are known. Indeed, PoA was proposed in Ethereum ecosystem for private networks due to a better performance and fault tolerance. The set of trusted and assumed honest nodes in PoA, identified by a unique id are called authorities[63]. PoA may not be appealing to use in global blockchains where users are able to exchange value anonymously since participants must be trusted and known with more control on them[64].

2.3.5 Ethereum

Ethereum[65] is an open-source platform based on blockchain technology which was introduced by Vitalik Buterin. Decentralized applications can be built on Ethereum in an automated manner without any third parties interfering. Indeed, Ethereum relies on a public blockchain, which means everyone can join the network. In this platform, users may use Ether as a currency to interact with each other.

Ethereum Virtual Machine[66] (EVM), the heart of Ethereum, is a runtime environment which executes smart contracts. Its leading role is to provide security and run programs on all nodes in the network. EVM has been successful in the prevention of double-spending when the same amount of money is used for buying different items by using the Proof of Work algorithm. Also, all the transactions will be recorded on the ledger all across the network; the same amount of money cannot be used twice. In Ethereum, there is no need to develop a fresh blockchain from the beginning because the platform allows users to implement different applications. Programmers may use different operations in their developed decentralized applications based on the network purpose. The developed programs are reliable, as they are self-executing without any demand for a trusted third party to administering the system.

Ethereum is built with a Turing-complete language that can run any program on the network if enough time and memory are given to the network, it also can support all types of computations, including loops [67]. Turing completeness enables users to define different rules and types of transactions for their system which brings new challenges to the developers.

In Ethereum, when an operation is executing, a fee has to be paid because each line of code in the smart contract needs computational effort. We determine that fee by Gas which exists inside of the Ethereum virtual machine. Indeed, when we execute complex operations in the smart contract, the amount of gas that has to be paid will be raised. The account performing the transaction has to pay the Gas. So, the account has to have enough ether on its account balance to pay the fee. Gas ensures that the system will not take down with huge computational problems or intensive works. As Ethereum and its EVM is Turing Complete, gas is used to limit infinite loops. GasLimit is used to designate the maximum gas we would like to spend on a transaction. We also specify gasPrice the price for each gas unit in the ether.

Smart Contracts

Ethereum is used as the main platform for implementing smart contracts, which are collections of attributes and functions that can use inheritance on other contracts[68]. Its machine-level programming language, Solidity[69], the primary language of smart contracts in the Ethereum blockchain influenced by C++, Python, and JavaScript. In the traditional sense, a contract defines as an agreement between two or more parties when the parties must trust each other to complete the contract as promised. In smart contracts, the need for trust between parties has been removed as a computer executes them. In other words, smart contracts are a piece of code running on the blockchain without any possibility of downtime, censorship, or fraud[70]. Smart contracts may receive information as input, and the process is based on the pre-defined rules and perform actions as output[71]. However, as the contracts are written code implemented by people, code bugs or oversights may happen. This leads to an attack or exploitation which violates the essence of the immutability of the blockchain. The attack cannot be stopped unless rewriting the underlying code[72]

. Using an interpreter in EVM, Smart contract compiles into a series of numbers and letters called bytecode which will be stored on the blockchain. Bytecode is an instruction set for the EVM which encode references to other functions and contracts that are called during execution using an application binary interface (ABI). ABI is the interface between machine-level instructions and a human-readable higher-level programming language. In other words, ABI lists all the function definition and arguments that exist on the contract in JavaScript Object Notation (JSON) format[73]. ABI also specifies arguments, types of arguments, return values, and type of return values of each function in the code. [74] developed two smart contracts, both written in Solidity, to having a voting system without a third party interfering. One of the contracts is called the voting contract, which is used to implement the voting protocol, control the election process and verifies of zero-knowledge proofs. In a method called zero-knowledge protocol, a party or the prover, can prove some facts to other parties or the verifiers, without revealing any other information[75].

The second contract in their implementation is the cryptography contract which distributes the code for creating zero-knowledge proofs. This results in, the same locally cryptography code for all voters without interacting with the Ethereum network.

2.3.6 Hyperledger

Hyperledger[76] is an open-source system which is not dependent on cryptocurrencies, but it is considered as an important permissioned blockchain which is mainly used in organizations. The main components of Hyperledger are endorsers, committers, smart-contracts, orderers, and validators. By supporting modular consensus protocols, Hyperledger allows the system to be adapted to particular trust models and use cases. Fabric is considered as the first blockchain system that runs distributed applications written in general-purpose programming languages including Go, JavaScript, Python, and Java[77]. The four main Hyperledger frameworks are listed below:

- Hyperledger Iroha: is written in C++ and uses the Byzantine Fault Tolerant consensus algorithm. The primary purpose of Iroha is for developing mobile application projects

- Hyperledger Sawtooth: Can be used for both permissioned and permissionless. Sawtooth uses a new consensus algorithm named Proof of Elapsed Time. Hyperledger Sawtooth, which is mainly designed for enterprise use, supports both smart contract virtual machine or business logic within one instance of the blockchain network.

- Hyperledger Burrow: is a modular blockchain client which uses a permissioned smart contract interpreter to follow the Ethereum Virtual Machine specification.

- Hyperledger Fabric: is a framework that contributed by IBM to develop modular architecture applications. It allows components like membership services and consensus to be plug-andplay. The smart contracts on the Hyperledger Fabric are called chaincode, which acts as the logic of a system.

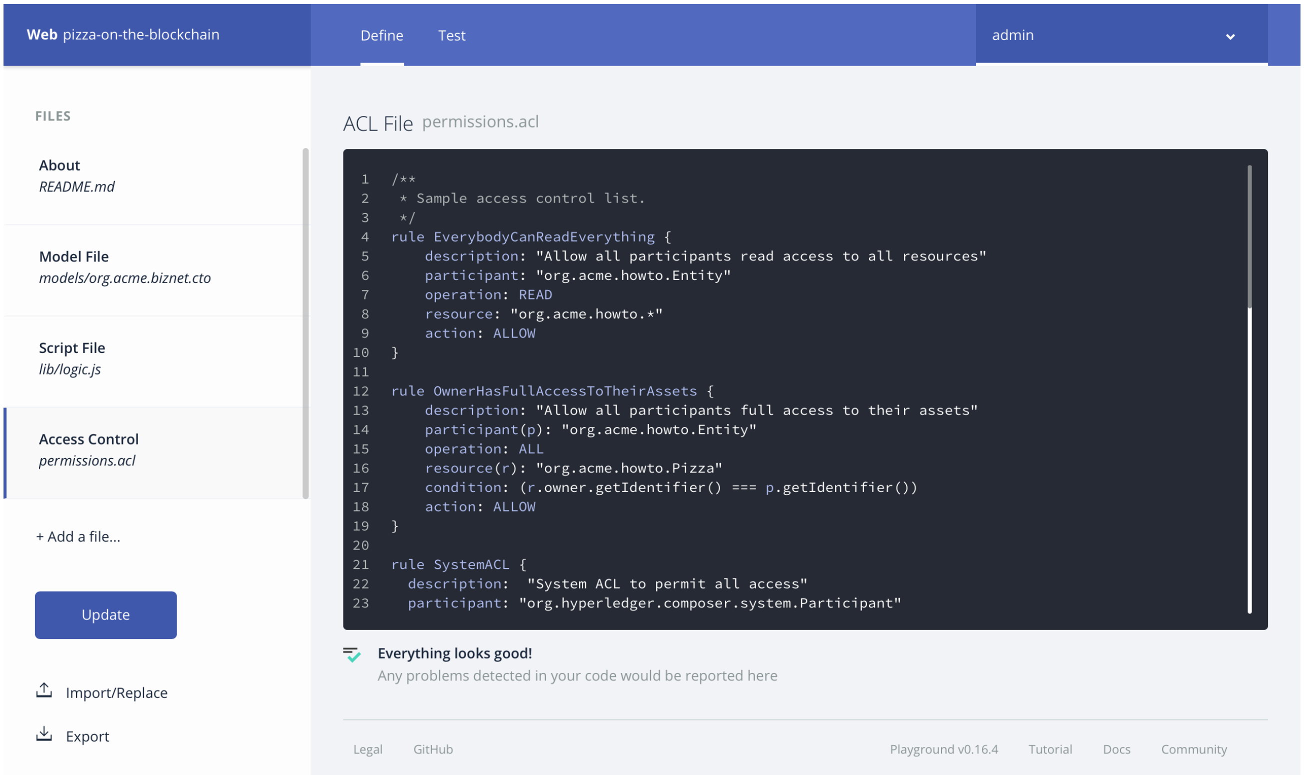
Currently, there are five hyperledger projects in the development of process[78]. Hyperledger Fabric is used in food source tracking to ensure food safety and increase transparency[79]. This project is an attempt to prevent incidents like sickening or dying after eating contaminated foods. The application allows authorized users to access food supply chain data, including certifications temperature data, from farm to the ultimate consumer. Another usage of Hyperledger Fabric is in the airline industry to make ticketing processes easie[80]. They tried to improve transparency by using a web-based interface that helps with saving money, record-keeping, and improving agility and security in such a complex business. The third project associated with coffee industry[78]. Crucial data about harvesting, packaging, and shipping is recorded to Hyperledger Sawtooth using QR scan codes. They aimed to increase customer's trust by letting them scan the codes to explore coffee history from origin to local stores and into customers' grocery carts.

Hyperledger Fabric also powers the fourth project[81]. In this project, the largest retailer in China provides an open platform to help enterprise customers with creating and using their blockchain applications by creating and updating smart contracts on private or public enterprise clouds. For instance, one insurance company uses the platform to track e-invoices, which are required official receipts for business in China.

The American Association of Insurance Services developed a project on IBM Blockchain, which is powered by Hyperledger Fabric[82]. In the proposed solution they aimed to make statistical data collecting and sharing between insurers easier; so they have better control over their own data and who is allowed to explore what data. With the help of blockchain the platform is more efficient and secure since all the new and old data is recorded on an immutable ledger.

Hyperledger Composer

Hyperledger Composer is developed by IBM to accelerate developing blockchain application. The composer is written in JavaScript and runs over Hyperledger Fabric, helps with modeling business network and integrating existing systems with blockchain applications. A business owner can define the assets, business rules, involved participants, and access controls, which helps to solve business problems. Developers can benefit from the composer by interacting with it, developing smart contracts, and testing their applications easily without installing anything. Figure 2.2 shows an example of the composer web interface.



```
/**
 * Sample access control list.
 */
rule EverybodyCanReadEverything {
    description: "Allow all participants read access to all resources"
    participant: "org.acme.howto.Entity"
    operation: READ
    resource: "org.acme.howto.*"
    action: ALLOW
}

rule OwnerHasFullAccessToTheirAssets {
    description: "Allow all participants full access to their assets"
    participant(p): "org.acme.howto.Entity"
    operation: ALL
    resource(r): "org.acme.howto.Pizza"
    condition: (r.owner.getIdentifier() === p.getIdentifier())
    action: ALLOW
}

rule SystemACL {
  description:  "System ACL to permit all access"
  participant: "org.hyperledger.composer.system.Participant"
```

Everything looks good!
Any problems detected in your code would be reported here

Figure 2.2: Hyperledger Composer Web Interface-taken from [83]

2.3.7 Bitcoin vs. Ethereum

Bitcoin and Ethereum, the peer to peer decentralized payment systems, are the most prominent platforms for exchanging cryptocurrencies with $150B of value as of Sept 2017[84]. In both platforms, miners sequence transactions into a block. Also, everything in both network is written on public ledgers. So, the need for the third party has been eliminated. Another similarity is that additional ether or bitcoin is released via the mining process. The table 4.1 summarizes some differences between the two platforms.

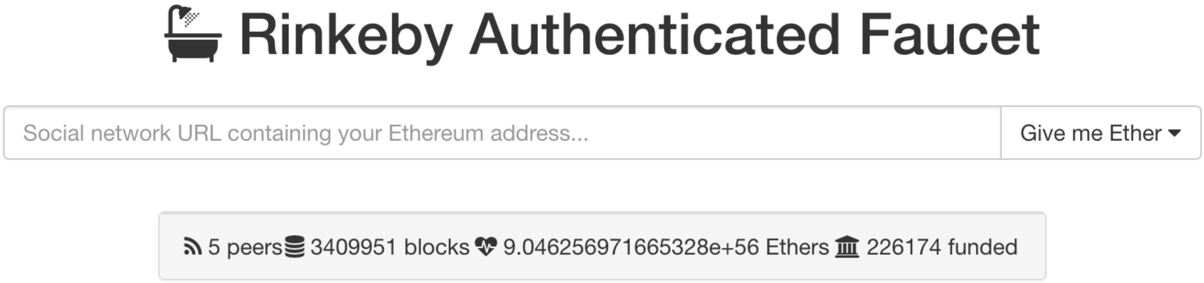
Bitcoin	Ethereum
The smart contract is not very programmable and extensible	Most popular blockchain for creating smart contracts
Bitcoin block time is 10 minutes	Ethereum block times are currently at about 14 seconds
Bitcoin uses C++ programming	Ethereum is written in Turing complete language.
The reward in Bitcoin is 25 per block	The reward per block is five ether
Reward halves every 210,000 blocks	reward remains constant, and it does not halve.
A limited amount of Bitcoins can be created; the maximum is 21,000,000.	In Ethereum there is no limitation in creating ether
Bitcoin operates on a proof of work basis.	Ethereum operates on a proof of work basis but working towards changing to a proof of stake.
Miners are rewarded Bitcoin	Miners are rewarded for processing transactions and executing smart contracts

Table 2.2: Bitcoin vs Ethereum

2.4 Technologies To Interact With Blockchain

2.4.1 Rinkeby Network

Testnets simulate Ethereum which gives developers a chance to test their projects before deploying on the Main Ethereum Network. Currently, there are three main testnets, such as Rinkeby, Ropsten, and Kovan. Ropsten is based on proof-of-work consensus, and both Rinkeby and Kovan are based on proof-of-authority consensus. Acquiring no real-world value ethers on the Rinkeby network is pretty straightforward. The user has to make a public post including his address on either Facebook, Twitter or Google Plus and then share the link with Rinkeby Faucet as shown in Figure 2.3 Rinkeby is used in [85], [86], [87] for Ethereum simulation and testing purposes.



Rinkeby Authenticated Faucet

5 peers 3409951 blocks 9.046256971665328e+56 Ethers 226174 funded

Figure 2.3: Rinkeby Faucet-taken from [88]

2.4.2 Next Server

Next.js is a framework for server-rendered applications. In server-rendered applications, all the resources are housed on the server by default. Compared to client-side rendering in which all the resources are rendered in the browser, server-rendered applications are faster and have a better performance For instance, when we implement web apps with PHP(Hypertext Preprocessor), we create some files, write PHP code, and then deploy it. The app then is rendered on the server by default; so there is no need to worry about routing much. That is exactly what happens with Next.js. Instead of PHP, we build the app with JavaScript and React. Next.js loads pages faster and it has a great performance[89]. Choosing Next is a great option when developing server-side rendering applications. With server-side rendering, HTML converts files into usable information for the browser and loads the first page faster. Also, the browser does not need to download all the JavaScript to render HTML from the server. This will result in better performance compared to client-side rendering.

2.4.3 React.js

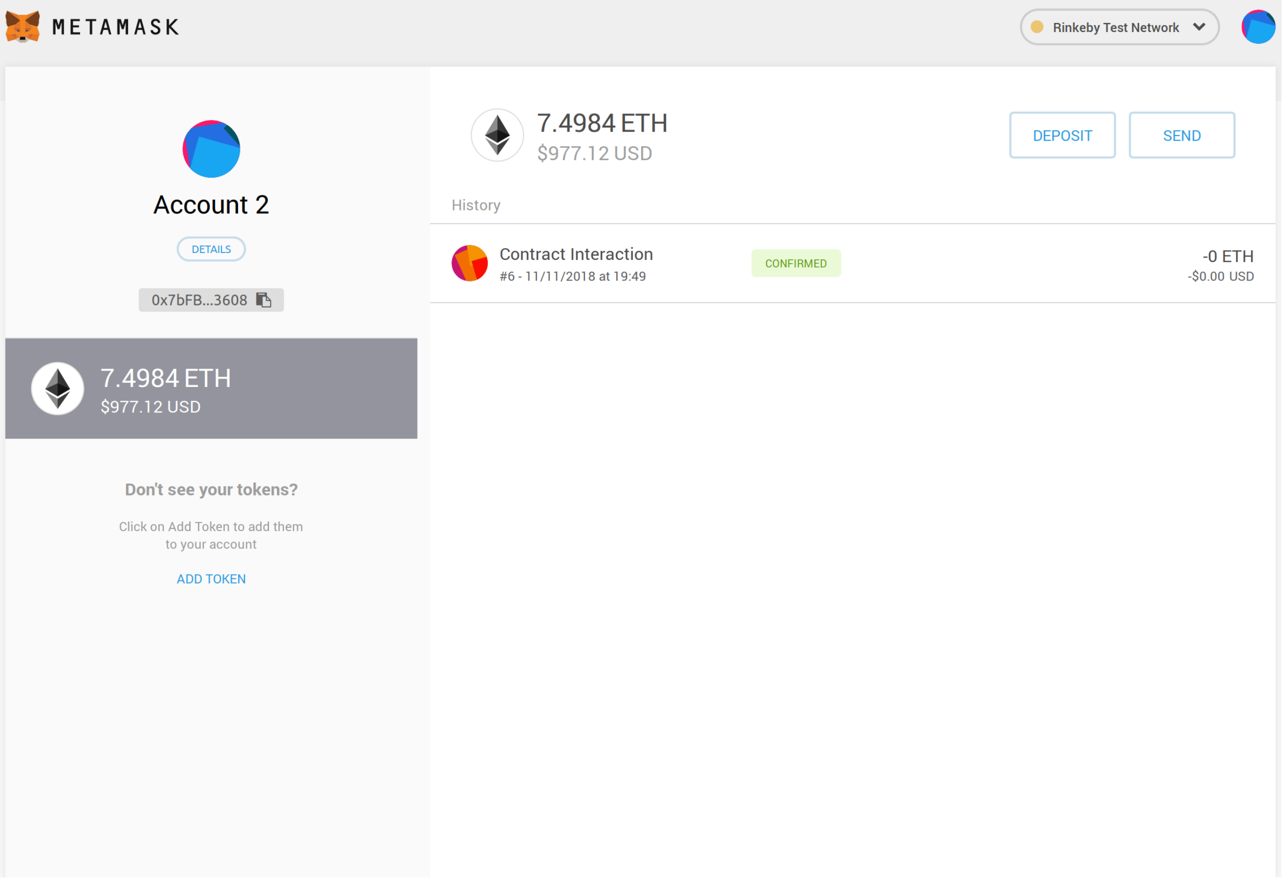
React is a flexible and efficient JavaScript library to help the developer build an interactive user interface. With the help of this library, developers can define interfaces like a function, design simple views for each state in their application and develop new features. In React, encapsulated components are used to make complex UIs since component manage their state. React also efficiently update and render the components when the data changes. Data can be passed through the application and keep the state out of the DOM since components do not follow any specific templates, but instead the logic is written in JavaScript[90].

2.4.4 Web3

The web3 is a JavaScript library which is a collection of modules and contains specific functionality for the ethereum ecosystem. For instance, the web3-eth is used for the ethereum blockchain and smart contracts; the web3-utils contains useful helper functions for DApp developers. Ethereum as a blockchain has different levels of finality and therefore needs to return multiple stages of an action. For this reason, Web3 provides multiple ways to act on asynchronous functions to help web3 integrate into different projects with different standards, including allowing a callback.[91]. Web3 gives the developer an option to choose to connect to an Ethereum node. Inter-process communication(IPC), Websockets, and HTTP are three different providers to connect to the node. The provider acting like a bridge between web3 and the blockchain. In other words, the providers submit JSON-RPC requests to an HTTP or IPC socket-based server and return the response.

2.4.5 MetaMask

MetaMask is a browser extension that allows developers to run Ethereum dApps in the browser without running a full Ethereum node. Indeed, it turns browsers into Ethereum browsers to retrieve data from the blockchain and help users sign blockchain transactions and manage their identities on different websites over the Internet. MetaMask also includes a secure identity vault which is encrypted and locally stored in the browser[92]. In MetaMask a seed phrase is used to restore all the accounts a user has been created. Also, users can select their desired network such as Main Ethereum Network, Ropsten Test Network, Kovan Test Network, Rinkeby Network, Localhost on port 8545, or Custom RPC. All transactions, including ether transfers transactions, rejected transactions, and contract deployment transactions can be viewed under the History section in the interface. Figure 2.4 shows MetaMask interface.



Account 2

0x7bFB...3608

7.4984 ETH
$977.12 USD

Contract Interaction
#6 - 11/11/2018 at 19:49
-0 ETH
-$0.00 USD

Don't see your tokens?

Click on Add Token to add them to your account

Figure 2.4: MetaMask User Interface - taken from[92]

2.5 Escrow Payment

Over the Internet, a lot of purchases are taking place every day. People prefer to save time by not purchasing in person at the stores. Companies also encourage people to buy online by offering attractive deals to customers. However, there is always a tension between both parties; customers are worried if they receive the product that they have paid for; companies or the sellers are concern if they receive the payment when they deliver the product. In other words, both parties need to trust each other and seeking a way to guarantee the exchange of product and payment. According to [93], the need for trust resolved somehow by paying through a trusted third party for an escrow. For instance, if the buyer does not receive a paid purchase, the third party intervene to refund the buyer. The scenario will be the same for credit card companies; they reverse charges if the customer makes a fraud complaint. Even though escrow is adapted by a lot of online markets in today online world, it cannot be a good option since the reliability of the third-party mediator cannot be authorized. Escrow can be a good option if it is used alongside with the blockchain. [94]

tried to solve the Buyer and Sellers Dilemma by providing an automated trade protocol by using smart contract over the blockchain network. [5] is another study that tried to eliminate the need for trust in third-party by using blockchain and make automatic payments by using the escrow payment method.

2.6 Blockchain and Supply Chain

As mentioned previously, the risk of fraud and falsifying in traditional systems is significantly high. The use of blockchain can bring revolution to the shipping industry as a secured, decentralized, and encrypted public ledger. The technology makes the process paperless that all the involved parties may interact with each other by using public and private keys.

Industries can gain the advantage by adopting the technology into their business, including:

1. Real-time updates all the parties may interact with each other instantly as the need for mailing documents to exchange of data is removed
2. Higher accuracy: As all the processes are automated, the errors are much less likely to happen.
3. Full Transparency: Each party by having the required keys may access the information and all previous transactions on the ledger.
4. Increased security: users cannot make any changes in the system. This feature protects the system from fraud and manipulations. Also, as all the information is encrypted, make the system more secure[95].

Some research has been done on this area, for instance,

- Authors in [3] tried to discuss how supply chain manufacturing can benefit from blockchain technology to develop innovative platforms for transparent decentralized transaction mechanism in the industries and business. They proposed that due to inherited characteristics of the technology, trust can be enhanced through transparency and traceability within any transaction of goods, data, and financial resources. In their study, to demonstrate usage of blockchain technology, they used the technology in a global supply chain network.
- As people pay more attention to the standard of living and consuming habits recently, a study[4] in China, proposed a solution to use RFID for gathering data, circulation in production, processing, warehousing and distribution in the agri-food supply chain to address the

issue of food safety and quality in the supply chain. In their research, blockchain technology is used to guarantee information on the system to enhance food safety and prevent food loss. The point that makes the developed platform unique is that third parties such as government departments and third-party regulators, have not eliminated. Indeed, such parties have responsibilities, including inspecting members authenticity, but they are normal nodes without any specific privileges.

- Weber et al.[5] discussed that blockchain as an innovative technology could pave the way to integrate business process across organizations since decentralized and transactional data can be shared across a network of untrusted participants. In their system, they integrate blockchain into the choreography of processes to remove the existence of central authority need, but trust maintained.

- To increase transparency and provenance in electricity usage, authors in [6] proposed a system where user can monitor how the electricity is used, and also there is no manipulation from either party.

- As a try to eliminate inefficiencies of the coffee supply chain, Bext360[96] uses blockchain to track all elements of the worldwide coffee trade, from origin to consumer.

- Pacific Tuna Project[97] is another attempt to track tuna life path and prevent illegal or unethical produced seafood products into the seafood supply chain. They have used blockchain to increase transparency and traceability and exclude illegal and unethical producers out of the supply chain.

- Walmart requires leafy green suppliers to use end-to-store blockchain system. The company has a plan to apply the technology to other products covering fruits and vegetables as they believe the technology will increase efficiency. Walmart aims to use a blockchain-based system to track foodborne illnesses together with Centers for Disease Control and Prevention (CDC) in the U.S. [98]

- Several big companies such as Dole, Unilever, Walmart, Kroger and Nestle, Golden State Foods, as well as Tyson Foods, McLane Company, and McCormick and Company in partnership with IBM gathered together in a consortium to examine blockchain benefits in supply chain[99]. The companies aim to answer consumers' demand to increase transparency by using blockchain technology to easily remove a contaminated product from store shelves and restaurants since they can trace contaminated product to its origin.

In this thesis, we tried to visualize the products life path in the form of a map to the users in the implemented food supply chain. Organizations can benefit from the system by identifying attempted fraud more easily. Also, they can gain profit as it can increase customers trust in that organization.

2.7 Summary

In this chapter, we reviewed some topics related to our research, including supply chain management, visualization, the basic concepts of the blockchain technology and its use cases, Ethereum and the technologies to interact with Ethereum. In this section, we want to summarize our findings and potential benefits of blockchain technology in visualizing provenance of created products in the system.

According to Google, the definition of the supply chain is "the sequence of processes involved in the production and distribution of a commodity". In other words, the supply chain is a network of different entities or organization involved in a process to deliver a service or product from the supplier to the consumer or end-user. The supply chain needs to be supervised to get the most out of it with minimum cost and effort. Supply chain managers need to make a profit along the way to answer customers' demand, give them the product they want, whenever as often as they want with a satisfying price. Transparency and traceability are two important factors that supply chain managers need to pay attention to as customers have become more concern about the production of a product, especially in food supply chains. Various studies have been tried to increase transparency and traceability. Visibility is another vital element in an efficient supply chain which has been missed out in some studies. Information visualization can be a potential solution to increase visibility. As it has been discussed earlier, visibility is a key factor to manage risk and improve the supply chain strength and efficiency; also a better flow of information results in customers' trust and satisfaction. There are already some applications that tried to increase transparency of the supply chain by using visualization. However, there is a huge drawback in their systems; the information shared by members in the supply chain and used in visualization cannot be trusted because the centralized entity of the system controls everything. Participants in the network do not own or cannot access data unless the powerful third-party unit grants permission. In such systems, the central unit is a vulnerable target for attack; the whole system breaks down if the central node shuts down. Another problem with retrieving information from the central authority is the trustworthiness of data. There is no guarantee that data has not been manipulated on purpose. For instance, if the system administers be bribed, the valid data would tamper; so the whole system cannot be trusted. Facebook can be considered as an example. It was in 2018 that a big political scandal took place; Cambridge Analytica harvested personal data of about 82 millions of Facebook users without their consent[100]. Personal information of those 82 millions people instead of being in control of themselves was in control of Facebook.

With the advent of new technology, blockchain, we would be able to retrieve data from a system which is not dependent on the system administrator. Satoshi Nakamoto introduced the first blockchain application in 2008 with Bitcoin. The main goal was to eliminate the need for central authority and guarantee successful transactions over a decentralized and trustless network of nodes or computers. In the network, each participant has a copy of the ledger, which has all the transactions recorded. Different applications were developed to make use of great features of the technology. Blockchain has permanent records of data; it cannot be altered or deleted, so it creates a reliable source of information. Furthermore, the application is open source, and trust among members of the network is not required anymore. However, the technology is not entirely flawless. The 51% attack is a possible problem exists in the network where a miner or group of miners take more than 50% of computing power. When they take control, they would be able to prevent transactions confirmation, interfering payments between users or double-spending tokens.

In a supply chain, consumers need to be sure that data is not manipulated and the most accurate is used when they browse life cycle of a product since accuracy is crucial for them to trust the organization. The blockchain technology can help in implementing a supply chain in which parties have a direct connection with one another, the history of transactions are recorded immutably and retrieving data is without any intermediary. Information visualization has not been explored in a supply chain over the blockchain network before. In this thesis, we tried to apply the blockchain technology to make use of its benefits to reveal the history of products and let users track a product from origin to destination. Table 2.3 shows the summary of related works.

Authors	**Topic and Results**
[7], [8], [11], [10], [12], [14], [15], [16], [17], [18], [19]	**Supply Chain Management:** Supply Chain Management is defined and various studies has been explored to get a better insight about traceability and transparency and their importance in supply chain.
[22], [21], [24], [25], [26], [29], [30], [20], [28], [23], [27]	**Visualization:** is a tool to explore relationships, convey end-result, and increase audience interpretation. It can help with increasing visibility, collaboration and decision making in supply chain.
[2], [31], [33], [47], [51], [52], [53], [54], [55], [56], [56], [57], [54], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [70], [71], [72], [74], [73], [75], [77], [84], [85]	**Blockchain:** Fundamentals and characteristics of blockchain is discussed such as public and private blockchain, Consensus algorithms, Hyperledger, Ethereum and its features, Bitcoin and Ethereum comparison. Also, the important and recent technologies to interact with the blockchain are discussed.
[34], [35], [36], [38], [40], [42], [44], [39], [46], [47], [48], [50]	**Blockchain Applications:** Currently, blockchain is widely used in different areas including meat industries, health, agriculture and data management
[95], [4], [5], [6], [96], [97], [98], [99], [10]	**Blockchain and supply chain:** Blockchain brings revolution to the supply chain as a secured, decentralized, and encrypted ledger. The technology makes the process paperless and all the involved parties may interact with each other directly without the need of any middle authority.

Table 2.3: Summary of Related Works

Chapter 3

Architecture & Implementation

The main objective of this thesis is to visualizing provenance of products registered in a supply chain. Visualization, was the last step in this research. As stated in the previous chapter, visualization is a tool to increase audience interpretation by letting them explore and discover relationships. Also, the most important element in the visualization is information. So, We tried to implement and evaluate a complete supply chain system on top of blockchain technology to record and then reveal history of products. This chapter will discuss the architectural design of the system.

As it mentioned previously, in order to visualizing information, a system is needed to record history of registered products. In a real supply chain parties interact with each other and exchange value and products. The Literature Review presented in Chapter 3 evidenced that little has be done to apply blockchain to supply chains. In this thesis we tried to use blockchain based systems over traditional client-server for creating a supply chain and recording history of products. In traditional systems,

- Users are not able to own their data since a single point controls everything.
- Transparency of the system is limited
- The whole system relies on a single node; the probability of attacks is increased.
- If the central node goes down, other parties cannot access the entire system.
- As all incoming requests must be approved by the central party, performance of the system may be limited.

In this chapter, we tried to address the first question raised in chapter two.

1. How data can be retrieved to be used in visualization?

In order to answer the question, blockchain is used to develop a visualization of products provenance based on accurate, reliable and trustworthy information. Indeed, blockchain is used to take advantage of its great features including transparency, traceability and immutability and also to create tradable assets. A blockchain-based system is introduced to provide a secure decentralized tracking system. The system architecture is based on the Ethereum blockchain to eliminate the need for administrating the system by a third party. Thus the users can trust the information they view in the system as the data is not manipulated in any way. Specifically, the Ethereum test network is used to transfer money and store data. The network can contain one or more nodes in which they have a direct connection with each other. Each node is a machine, running an ethereum client, which can contain a full copy of the blockchain which is like a database that stores a record of every transaction that has ever taken place.

Figure 3.1 Shows the overall system architecture.

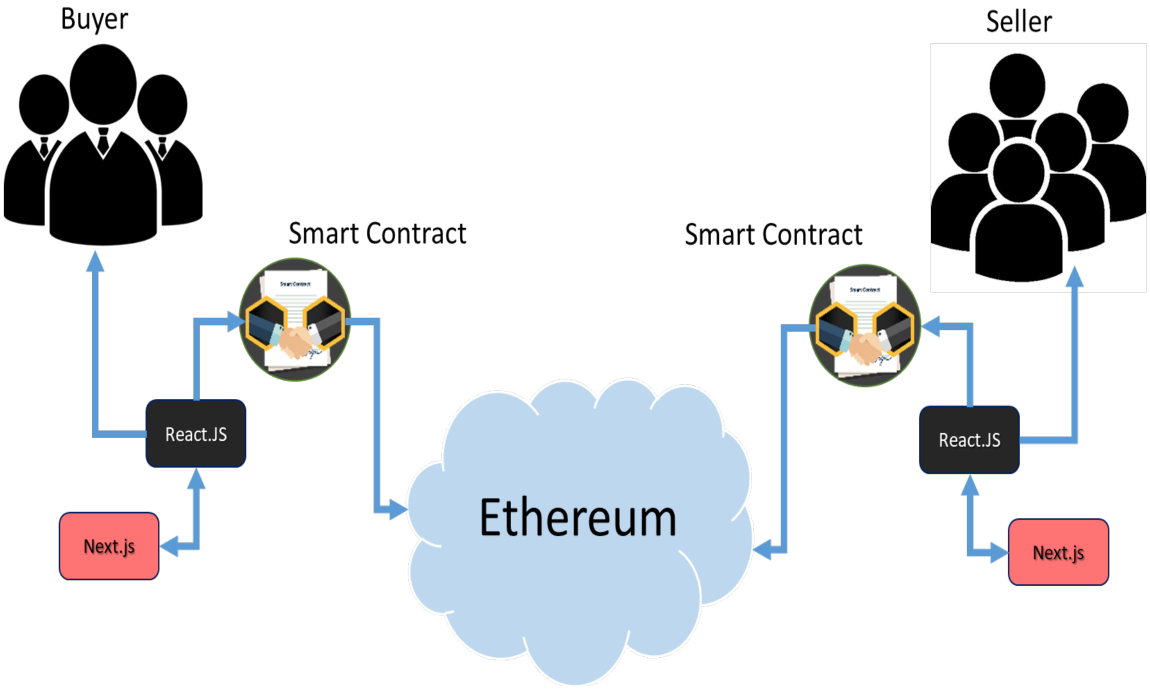


Figure 3.1: Architecture of the System

3.1 System Components

In this section we discuss main components of the architecture as shown in Figure 3.1.

Blockchain In order to lower the risk of fraud, falsifying in traditional systems and record history of products permanently and immutably, the blockchain used as the main component in this system. By using this technology, the system is decentralized in which all involving parties can have one-to-one communication. Also, immutability was another important reason to adopt the technology. When records on the blockchain are immutable, a person who has received a product can not claim that he has not bought it and refuse to pay for the service he has received. Also, transactions cannot be changed or removed; this results in reducing the risk of falsifying.

The primary purpose of the solution is to reveal details of digital assets which have been created in the system. So, we want to keep the data as it is and once added to the network nobody could change it even system administer. Specifically, the Ethereum platform was selected over other blockchain platforms such as Hyperledger. Hyperledger is most suitable for businesses which try to have confidential transactions within their organization. On the other hand, we tried to reveal every little detail of an asset to users; so confidential transaction is not suitable for having an interactive system in this study. Furthermore, all the information about accounts or transactions that ever happened on the network can be viewed in Etherscan.io according to the account address.

Smart Contract Can be considered as the system heart where all the logic and rules are defined. In this study, the smart contract allows the parties create assets, stores the assets, checks if an asset has been created, who has created it, retrieve all parties and assets information, users can look for an asset based on its ID. Also, parties can transfer an asset to a new owner. The contract is then deployed on a test Ethereum network called Rinkeby, a testnet that emulates Ethereum behavior. We connect to a specific node on the Rinkeby network so, the contract can be deployed on that specific node. Figure 3.2 Shows the contract compile and deployment work flow in more detail. When the contract is written in solidity, it needs to be compiled. After being compiled ABI is generated which lists all the function definition and arguments that exist on the contract in JavaScript Object Notation format. We stored the ABI in a file to use it for the contract deployment. When the contract is deployed on the blockchain node an address is returned which is used to create an instance of the contract.

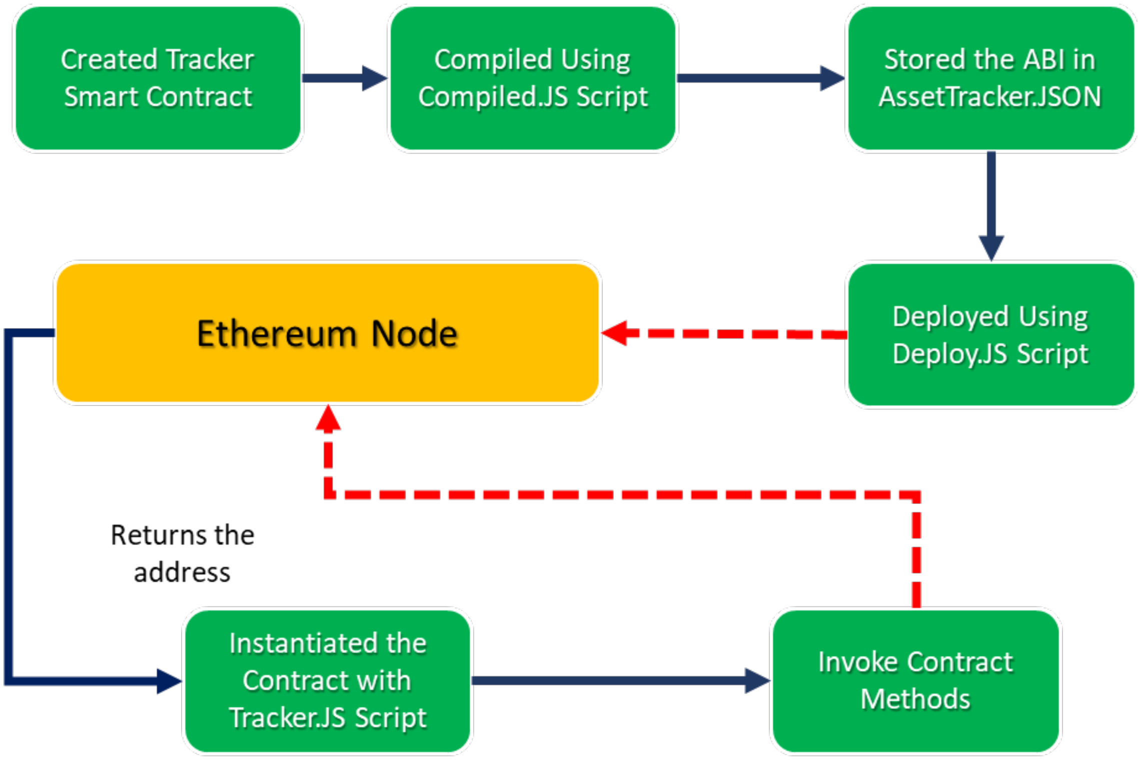

Created Tracker Smart Contract
Compiled Using Compiled.JS Script
Stored the ABI in AssetTracker.JSON
Deployed Using Deploy.JS Script
Ethereum Node
Returns the address
Instantiated the Contract with Tracker.JS Script
Invoke Contract Methods

Figure 3.2: Deploying the smart contract on Ethereum

Infura API In order to access to an Ethereum Node we need to connect to the node on the Ethereum Main Net which needs a lot of data to be downloaded from blockchain and kept it in sync. Infura allows developers to connect to Ethereum node without running one themselves. Infura API in this process acts as a connection point and gives access to the remote Ethereum node. Infura is an Infrastructure-as-a-Service(IaaS) product that helps to ease the access to the Ethereum Network. So, by deploying the contract on the Rinkeby which is remote Ethereum node using Infura API, the data is always in sync. Also, there is no need to spend real ethers to deploy the contract on the main Ethereum network, and the user does not have to download the whole Ethereum node.

Web3 When a smart contract is developed and deployed on the blockchain there is a need for client or front-end to talk to the blockchain. Web3 is the JavaScript library that helps with reading or writing data to the blockchain. Web3 is the main component that is used to interact with the Ethereum Blockchain including sending ethers between accounts, write and read information from smart contract. Web3 talks to the network through JSON RPC (Remote

Procedure Call) Protocol. When there is a need to read or write data from blockchain, web3 send the request with JSON RPC to the node in the network.

HDWallet Provider The provider or the connection point between the blockchain and the web3 is HDWallet Provider. With the Web3 provider, HD Wallet, Ethereum transactions can be signed for addresses that have been unlocked with the 12-word mnemonic. Account mnemonic, the twelve-word phrase is passed to HDWallet Provider to unlock MetaMask account and use the account to deploy the contract.

Front End The front-end of this system is created using React which helps to implement the interface to show the web page content to the user faster. We chose React to develop front end because it is quite popular among front-end frameworks and it is not complicated compared to other libraries. It is also easy to learn, fast and scalable which allows quoting HTML tag syntax. React web application can also transferable to mobile applications easily with available tools such as React Native. In this study we focused on web application because viewing map in the computer screen is more suitable and preferable. Besides, Next server is used to render React application. Next has great features like server-side rendering, working great with React and it is preferred over other frameworks when developing a multi-page application. As mentioned earlier, Next.js Server renders the entire react application, executes all the JavaScript code on the server, builds up the HTML document and send it to the browser. Using Next helps the user see the content of the web page much faster, especially when the application is extensive.

Figure 3.3 Shows the Architecture in more detail.

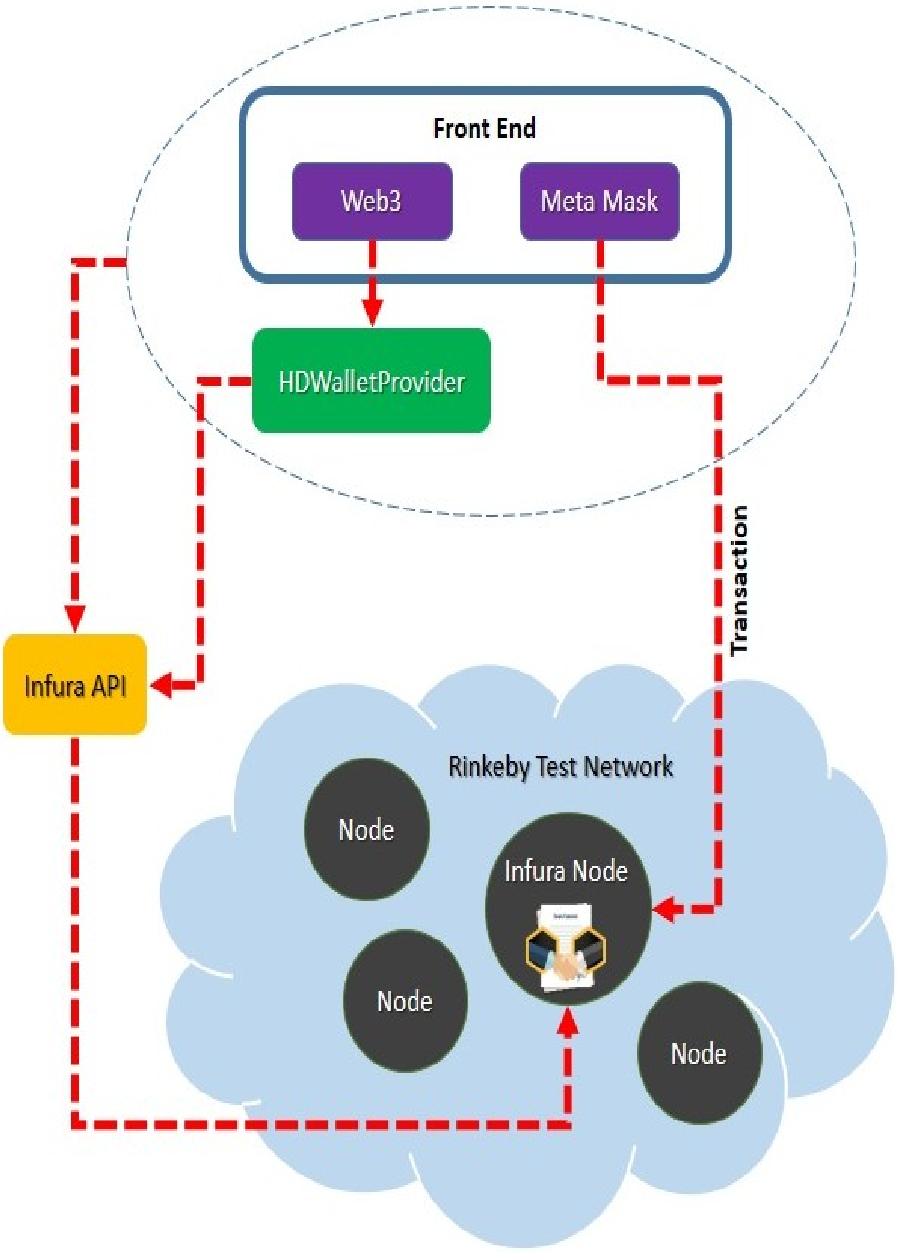


Figure 3.3: Detailed Architecture of the System

By interacting with the application users are able to:

- **Register a Product:** This allows a user to create new product information on the platform. It requires the product name, description, manufacturer and price
- **Buy:** Retailer is able to request an available product on the platform from the product owner. It requires a unique product identifier.
- **Stake:** This allows both seller and retailer to deposit money into the system to make sure that the transaction will conclude successfully. In other words, the seller ensures that he will receive the corresponding value of the product and retailer ensures that he will receive the product. Both parties are required to deposit twice the amount of the product's price. It requires a unique product identifier.
- **Confirm a Purchase:** This allows the retailer to confirm the purchase after receiving the product. When the confirms the purchase, the smart contract will be triggered to send the remaining ethers to both parties which conclude the transaction.
- **View the History:** This allows the customer to view the life cycle of a product.
- **View the list of products:** Data is retrieved from the blockchain to give the users of the system a summary of created products.

3.2 User Authentication

In this system MetaMask is used for user registration and authentication. MetaMask is an open source web browser extension and a cryptocurrency wallet which stores users public and private keys. Public and private keys are used in the system to allow different parties in the network exchange ethers for the service they provide or receive. Public key is available to the public and it can be seen by anyone as it is clear from its name. On the other hand, private key should be kept in secret since that is the only way of proving ownership of cryptocurrencies that users have in their account.

MetaMask used in this system due to hierarchical deterministic characteristic of this online wallet. Users are able to backup their accounts using 12 word phrase or account mnemonic. By using MetaMask public and Private keys are stored in users browser instead of being stored in a remote server which improves privacy. Another important reason to use MetaMask instead of traditional login methods is that it can interact with the web page by injecting web3 to the browser. In web3 there are some functions that require signing data with private keys. In that case those functions trigger MetaMask to ask for user confirmation for signing data with their private keys. It is noteworthy to mention that the roles in the system have simplified into three main categories including producer, retailer to make the system less complicated. By using MetaMask, producer has the ability to register a product by entering its information including name, description, and price. As Solidity does not support random number generator, a random number generator has been implemented to assign a unique ID to each product to make sure that the ID of each product is unique. If the transaction is successful, the product will appear on the main page of the application.

3.3 Payment Workflow in the System

As it mentioned earlier, when a product enters the chain, organizations pass the product to the next organization and make payment in the path until reaches the end point or customers. In order to develop the whole supply chain, the escrow payment service has been implemented in order to exchange the product and the corresponding value of the product. Figure 3.4 shows how the payment works.

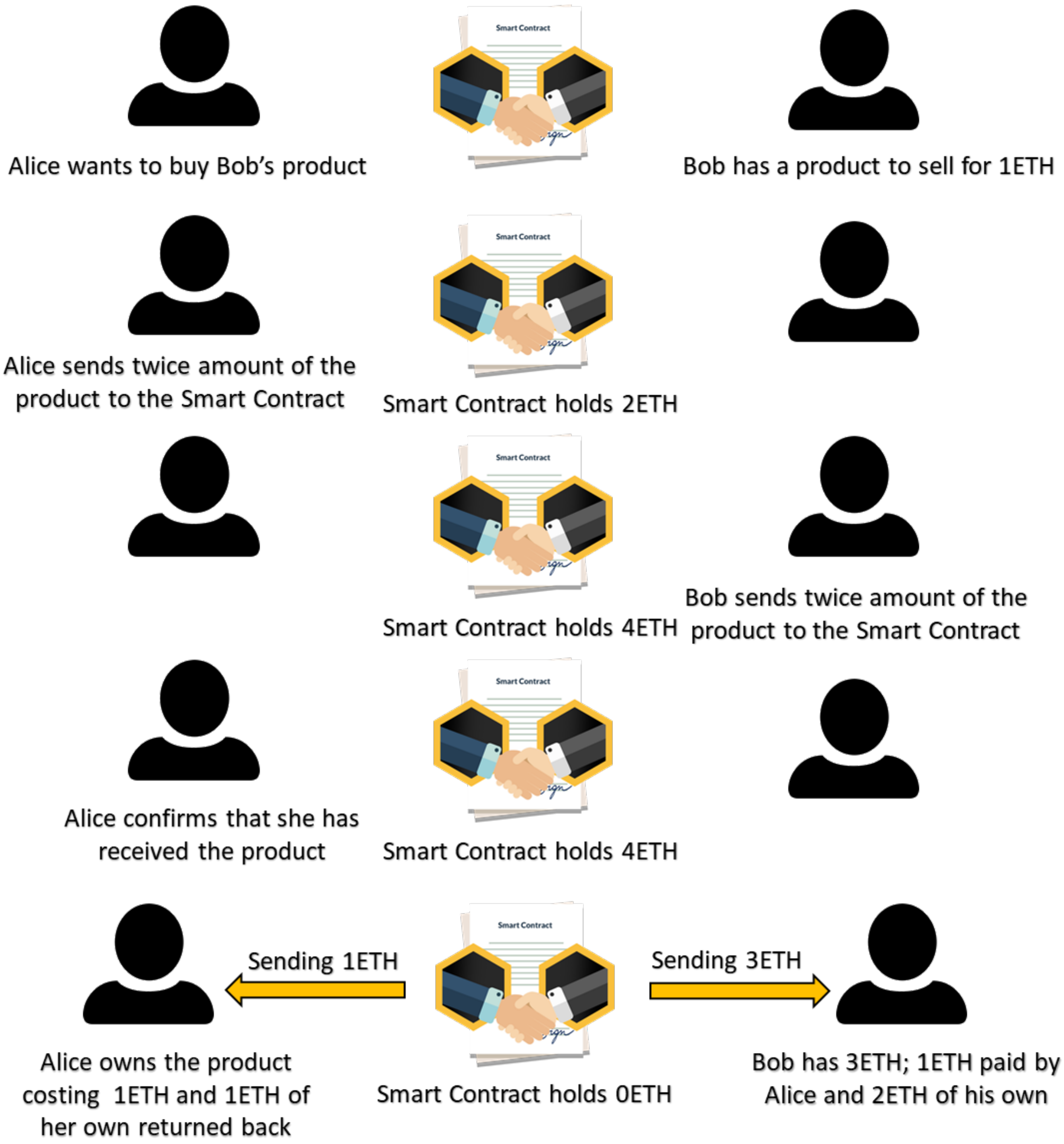

Smart Contract
Alice wants to buy Bob's product
Bob has a product to sell for 1ETH
Alice sends twice amount of the product to the Smart Contract
Smart Contract holds 2ETH
Smart Contract holds 4ETH
Bob sends twice amount of the product to the Smart Contract
Alice confirms that she has received the product
Smart Contract holds 4ETH
Sending 1ETH
Sending 3ETH
Alice owns the product costing 1ETH and 1ETH of her own returned back
Smart Contract holds 0ETH
Bob has 3ETH; 1ETH paid by Alice and 2ETH of his own

Figure 3.4: Payment Workflow

By using escrow payment the smart contract holds the deposit, none of the parties involved in the process can get hold of it. Also as parties stake 2x the value of the product, it motivates both parties to complete the transaction, so it prevents any conflict situations. After escrow payment is completed the ownership of the product will be changed in the whole system. However, the address of all the owners will be recorded to be used in the timeline of each product to determine provenance. Users can browse the history of a product to follow its life cycle in the system.

3.4 Retrieving Data from Blockchain

The main goal of this research is to provide a visualization of products provenance for users to view. To reach this goal we need to retrieve data that has been recorded to blockchain. As transactions are submitted to an Ethereum node only through a textual encoding format called JSON RPC, participating nodes expose this interface in different ways including HTTP connections, WebSockets or IPC. In this research based on software implementations, HTTP connections are used to submit transactions through JSON RPC. The JSON RPC interface also plays an important role when we want to retrieve data from the blockchain. As submitting JSON data-structures to a node to obtain information is time-consuming and cumbersome, programmers rely on different libraries to send transactions to the network and retrieve the data in JSON RPC format. Web3, a collection of JavaScript libraries, is one of the popular libraries which facilitates the connection between a blockchain network and users.

Figure 3.5 shows how web3 is used in the system. In order to read data from the blockchain using web3, an instance of the smart contract implemented in forms of JavaScript representation. In other words, implementing a client or website and calling functions of the contract are required to interact with on-chain products to read and write data. When the data is retrieved we need to find a way to convert it to understandable way for the users to view. This will be discussed in the next section.

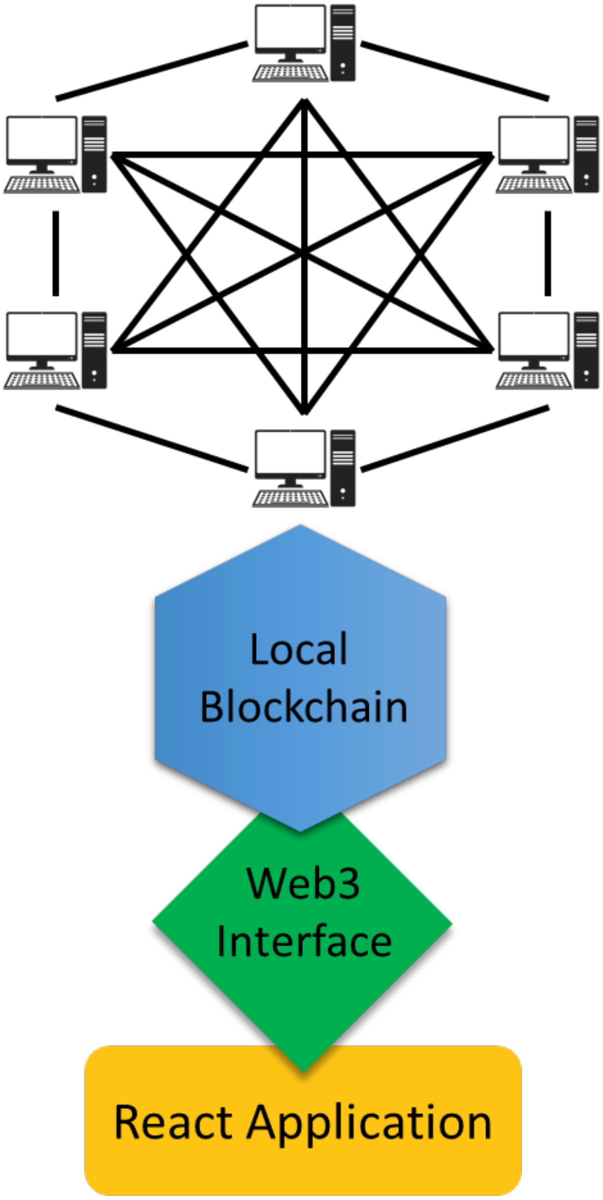


Figure 3.5: Usage of Web3 in the system

3.5 Sequence Diagram

Figure 3.6 shows the process of buying, selling and viewing the history in the system. Both seller and buyer need to register in the system through MetaMask. After setting up a password, a twelve word account mnemonic and address will be assigned to the user. By having twelve word account mnemonic users can access their account any where with any browser they prefer. At the main page, seller can create a new product in the system by choosing create new asset option in navigation bar under . In the new page, seller registers a new product by entering information, including name, description, manufacturer, price, coordinates and amount of stake. After completing information on the form by clicking create button, seller needs to confirm that amount of stake will be deducted from account. Seller also has the option to reject the transaction. On the other hand, buyer can buy a product he wants by looking at product summary at the main page. He can also, view the history of the product to browse the information such as location. Browsing the map is beneficial for the buyer to make decision. For instance, he may decide he does not need a specific product because it takes much longer than expected to move the product from the origin location to his location. So he may choose to buy a product from a seller with a closer location. After browsing the map and making decision, buyer can head to navigation bar to buy the product he wants. In the new page, a table of all products is also available for the buyer to choose the product. The buyer needs to enter the specific product's ID, coordinates of his location and amount of stake. Similar to the seller, after completing the form, a MetaMask window pops up to let the buyer know that the amount of stake will be deducted from his account. He has also this option to reject the transaction. After this step, the buyer needs to confirms the delivery to let the core of the blockchain or the smart contract know that he received the product. When the buyer he confirms that he received the product, the smart contract releases the money that was deposited and it will automatically change the product ownership in the system. In the last step in the process, the address and location of buyer will be added to the map.

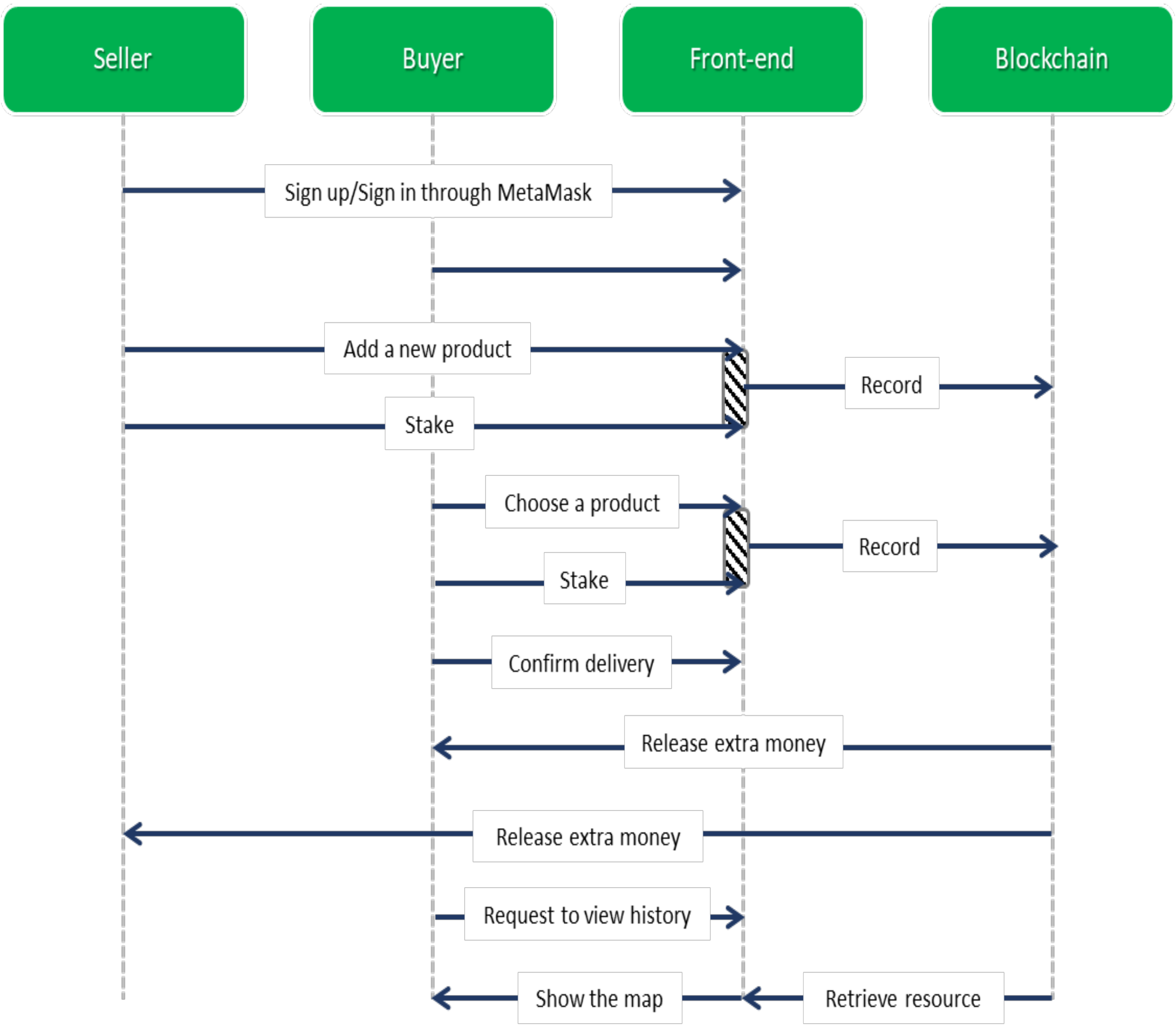


Figure 3.6: The process of buying, selling and viewing the history

3.6 Implementation

The system, described in chapter 4. This section describes the implementation of the system and necessary steps to setup the blockchain and web application for interacting with the blockchain.

3.6.1 Compiling the Contract

The smart contract, the logic of the system has been implemented in Solidity, the primary language of smart contracts. Node, throw an error and does not recognize .sol file because it consider all the files as JavaScript for the execution. Thus, we could not require it as file to compile it. For this reason, we installed two extra modules, including ‘fs‘ and ‘path‘ to help with compiling the contract.Path module helps with finding the contract code or .sol file within the directory. Fs or file system module allows working with any file systems on the computer. Here, the module is used to encode the .sol raw source code with UTF-8. After encoding the contarct, by using Solc which is module to compile the smart contracts, the solidity compiler has been compiled. With solidity compiler, multiple contracts can be compiled at once, but in this study, only one contract has been compiled. As it mentioned earlier, smart contract is stored on the blockchain network as the bytecode or binary data which makes the smart contract work on the network. With the ABI or Application Binary Interface we can tell Ethereum Virtual Machine to invoke a requested function. Not only functions of smart contract can be called with ABI but also we can expect getting data back in an specific format. In other words, ABI holds a list of argument, functions and data types in JSON format. In this work, a JavaScript script has been written to write the ABI and bytecode of the compiled contract in separate files so they can be accessed more easily and the smart contract does not need to compile more than once. Listing 3.1 shows the part of compiling script to save the ABI and bytecode to be used for the contract deployment.

```
const trackerPath = path.resolve(__dirname, ”contracts”, ”AssetTracker.sol”);
const source = fs.readFileSync(trackerPath, ”utf8”);
const output = solc.compile(source, 1).contracts;
fs.outputJsonSync(
  path.resolve(buildPath, ”AssetTracker.json”),
  output[”:AssetTracker”]
);
```

Listing 3.1: Contract Compiling

3.6.2 Deploying the Contract

To deploy the contract we need to get access to an unlock account and a node that connects to the blockchain. With the help of Infura API we did not need to install and run our own Ethereum node which takes a lot of resource and time. One more thing that we needed to deploy the contract, is the contract bytecode which has been produced and stored in a separate earlier by compile.js. The listing 3.2 shows a part of deployment script.

```
const result = await new web3.eth.Contract(JSON.parse(compiledTracker.interface))
    .deploy({ data: compiledTracker.bytecode })
    .send({ gas: ”2000000”, from: accounts[0] });
```

Listing 3.2: Contract Deployment

- By parsing the interface we inform web3 what methods are in the contract.
- Deploy command is used to tell web3 we want to deploy a new copy of the contract.
- Send command is used to send out a transaction that creates this contract.

3.6.3 Web3 and HDWallet Provider

In this study Web3, the JavaScript library is used to interact with The Ethereum Blockchain. With the help of Web3, data can be read or written to the blockchain. As it mentioned earlier provider in web3 can be thought as communication between the web3 library and an ethereum network, has a set of methods which allow the web3 library to send a request to a local network and receive the response to that request. In this study, HDWalletProvider will be used as a provider to unlock accounts. Indeed, HDWalletProvider allows us to connect to the Rinkeby which is hosted through Infura and also unlock an account to use simultaneously. The listing 3.3 Shows the script that we implement to create a web3 instance.

```

if (typeof window !== ”undefined” && typeof window.web3 !== ”undefined”) {
  web3 = new Web3(window.web3.currentProvider);
} else {
  const provider = new Web3.providers.HttpProvider( // creating our own provider
    ”https://rinkeby.infura.io/v3/8fe1f2192b9c400cb7722968d1837c18”
  );
  web3 = new Web3(provider);
}
```

Listing 3.3: Creating an Instance of Web3

Furthermore, to use HDWalletProvider, Truffle HDWallet module needs to be installed, and then by heading to infura.io, we need to sign up for an Infura API Key to use the service. By providing HDWallet with the MetaMask Account Mnemonic and Infura API, we can create an instance of web3 that is completely enabled for the Rinkeby Network. HDWalletProvider take two arguments as input; the first one is account mnemonic which is used to unlock accounts and the second argument is the ethereum node that we want to connect.

```
const provider = new HDWalletProvider(

  ”doll add humble swear soda gasp doctor thrive family shrug rack marble”,
  ”https://rinkeby.infura.io/v3/8fe1f2192b9c400cb7722968d1837c18”

);
const web3 = new Web3(provider);
```

Listing 3.4: Connecting To The Ethereum Node and Unlocking Accounts With Account Mnemonic

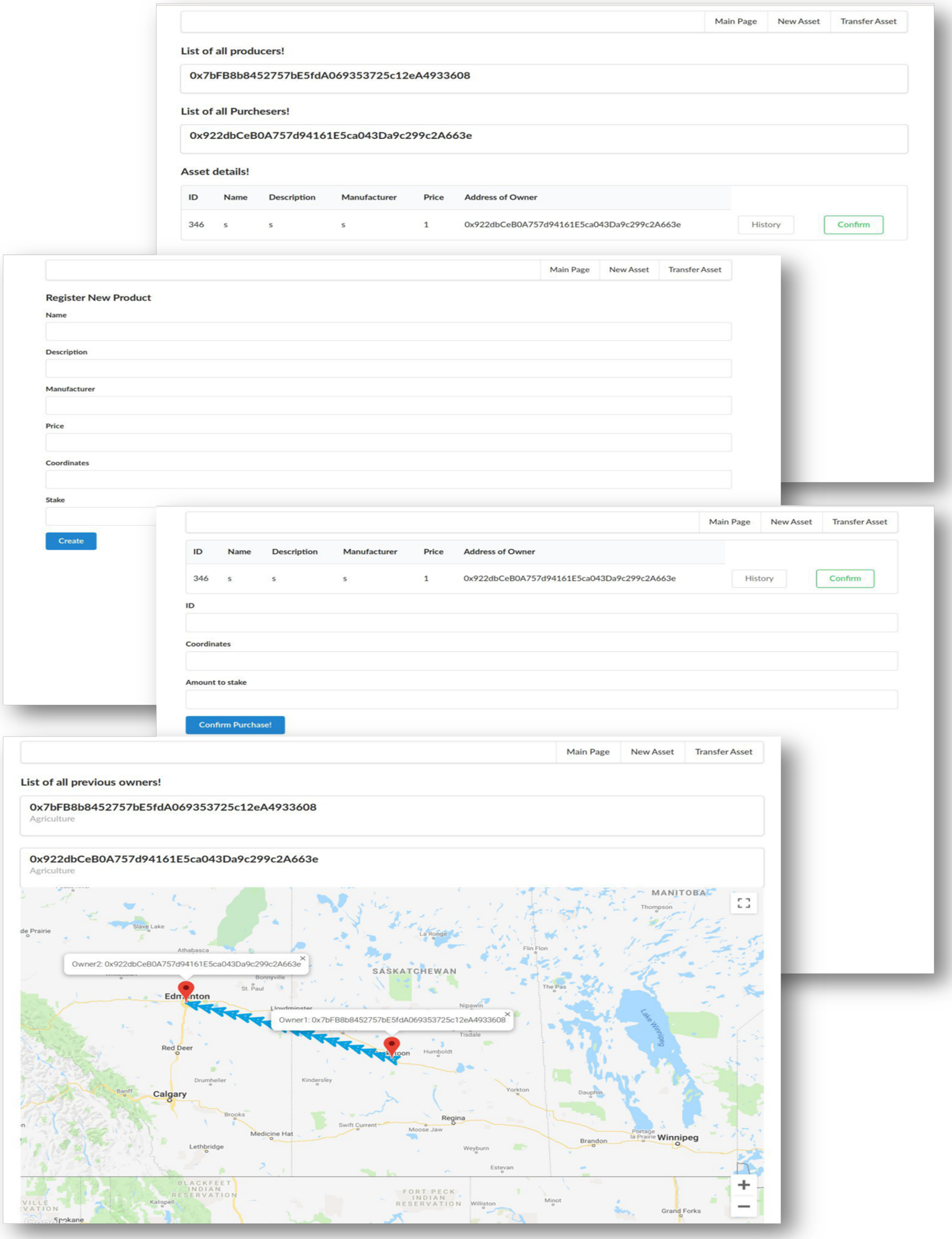
In order to interact with the contract from React to implement front end of the system, a local instance of the contract has been created. The contract ABI and the address that the contract has been deployed on, are required to create a copy of the contract.

```
const instance = new web3.eth.Contract(
  JSON.parse(AssetTracker.interface), //contract ABI
  ”0x091A1c4F41A9CFDDdfeCF728201b0525592E2ecB”
);
```

Listing 3.5: Creating an Instance of Contract

3.6.4 Front-end

The front end in this study is developed by using React; an open source JavaScript Library to build user interfaces which has been very successful in loading web pages faster. For every web page in this system a react component was created. Figure 3.7 Shows some examples of front-end which has been created with React. These pages are main page, creating a new product, buying a product, viewing the history. Both buyer and seller can view the list of buyers, sellers and registered products at the main page. Seller, can register a new product in the system by entering its information including name, description, manufacturer, coordinates, price and amount of stake. In the transfer page, the buyer can choose to buy a product based on the ID of that product in the system. User can view the history of a specific product in history page. Table 3.1 shows current dependencies that used in creating the front-end.



List of all producers!

0x7bFB8b8452757bE5fdA069353725c12eA4933608

List of all Purchesers!

0x922dbCeB0A757d94161E5ca043Da9c299c2A663e

Asset details!

ID	Name	Description	Manufacturer	Price	Address of Owner		
346	s	s	s	1	0x922dbCeB0A757d94161E5ca043Da9c299c2A663e	History	Confirm

Register New Product

Name

Description

Manufacturer

Price

Coordinates

Stake

ID	Name	Description	Manufacturer	Price	Address of Owner		
346	s	s	s	1	0x922dbCeB0A757d94161E5ca043Da9c299c2A663e	History	Confirm

ID

Coordinates

Amount to stake

List of all previous owners!

0x7bFB8b8452757bE5fdA069353725c12eA4933608
Agriculture

0x922dbCeB0A757d94161E5ca043Da9c299c2A663e
Agriculture

Owner2: 0x922dbCeB0A757d94161E5ca043Da9c299c2A663e

Owner1: 0x7bFB8b8452757bE5fdA069353725c12eA4933608

Figure 3.7: Example of front-end pages

Dependency	**Version**	**Description**
fs-extra	^7.0.1	A node module to work with file systems
google-map-react	^1.1.4	A component that allows to render any React component on the Google Map
mocha	^5.2.0	A test framework for Node.js
next	^4.1.1	A JavaScript library to render React applications
next-routes	^1.4.2	A package to create dynamic routes for Next.js
react	^16.8.4	A JavaScript library for creating web interfaces
react-dom	^16.8.4	A package for working with the rendering and DOM
semantic-ui-react	^0.82.5	A node module to work with file systems
solc	^0.4.25	Solidity Compiler
truffle-hdwallet-provider	^0.0.3	A module to provide connection between the blockchain and the web3
web3	^1.0.0-beta.35	JavaScript library to interact with Ethereum blockchain

Table 3.1: Front-end Dependencies

The implementation of front-end consists of six modules:

- **server.js:** In this module Next server is implemented to render React Application.
- **routes.js:** This file defines the different custom routes inside of the application. In other words it acts as navigation or route mapping of the whole system.
- **index.js:** This module is the root route of the system. It connects to the blockchain and retrieves information including address of organizations and summary of all products that have been registered in the system.
- **new.js:** This module implements a web page which connects to the blockchain that allows producer to register a new product by entering its name, description manufacturer, price and stake or deposit money to sell the product.
- **transfer.js:** This module implements a web page which connects to the blockchain that allows retailer to buy a specific product by entering its ID and amount of stake to deposit.

- **history.js:** This module implements a web page which connects to the blockchain and retrieves all the previous owners of a product based on ID that allows consumer to browse how the product exchanged a long the way.

3.6.5 Visualization

We discussed how we connect to the blockchain and retrieve information from it in the previous sections. Web3 is the main JavaScript library to interact with the blockchain. So, when the user wants to view the history of specific product, with web3 a connection is made to the smart contract to restore information. In order to visualize provenance we decided to use Google Maps API in which we are able to show the exact locations and the path between the locations. To use Google Maps API google-map-react module needs to be installed which allows us to render the developed react application on the Google Map. An other step was registering for a unique API key to use Google Maps API which is used for authenticate and billing. Listing (3.6, 3.7, 3.8) shows a part of the script to make a connection with the smart contract, mark the locations of the product and draw a directed poly-line to show the path.

- Making connection to the smart contract:

```
static async getInitialProps(props) {
    const lengthOwners = await
    tracker.methods.lengthOwners(props.query.id).call();
    let addresscounts = await tracker.methods.zipCounts().call()

    let owners = await Promise.all(
      Array(parseInt(lengthOwners))
        .fill()
        .map((element, index) => {
          return tracker.methods.owners(props.query.id,index).call();
        })
    );
    let addressArray = await Promise.all(
      Array(parseInt(addresscounts))
        .fill()
        .map((element, index) => {
          return tracker.methods.zipcodes(props.query.id,index).call();
        })
    );

  const myLatLng = addressArray.map(coords => {
    const [lat, lng] = coords.split(',').map(Number);
    return {
      lat,
      lng
    };
  });

  return {
    lengthOwners,
    owners,
    myLatLng
  };
}
```

Listing 3.6: Making Connection to the Smart Contract to Retrieve Information

- Marking locations on the map:

```
renderMarkers(map, maps) {

  for(let i = 0; i < this.props.myLatLng.length; i++){
  let marker = new maps.Marker({
    position: this.props.myLatLng[i],
    map,
    title: i
  });

  marker.addListener('click', function() {
    infowindow.open(map, marker);
  });
  }
}
```

Listing 3.7: Marking the Locations

- Draw Poly line to show the path:

```

renderPolylines (map, maps) {
    let geodesicPolyline = new maps.Polyline({
      path: this.props.myLatLng,
      geodesic: true,
      strokeColor: '#00a1e1',
      strokeOpacity: 1.0,
      strokeWeight: 4,
      icons: [{
        icon: {path: google.maps.SymbolPath.FORWARD_CLOSED_ARROW},
        offset: '100%',
        repeat: '20px'
    }]
    })
    geodesicPolyline.setMap(map)
  }
```

Listing 3.8: Polyline Between Locations

3.7 Summary

In the system, research question one that is mentioned in the chapter two solved by using the blockchain to record the data and web3 to make the connection to retrieve the data. By using the proposed architecture the single point of failure has been eliminated; so the system would be cost effective since the maintaining cost will be minimized. Besides, by using web3 a connection can be made to restore information that has been recorded by users. Then with the help of visualization tool,Google Maps, we turned the information to human-readable data which can be understandable by general users. We dynamically marked the location of the product whenever there is a change in location. Also with the help of React Library of Google Map, we draw poly line between locations to show the sequential direction. We visualized the history of products on top of blockchain to use its feature such as immutability. When the user wants to review the locations of a specific product he needs a certainty that data has not been altered or manipulated. Visualizing trustful and accurate was only achievable through blockchain only which can record data without any alteration.It is noteworthy to mention that, in this thesis we used Infura API which is a platform as a service to connect to Rinkeby Network which is the exact replica of Ethereum network. By using Infura there is no need to sync the blockchain which is complicated and requires a lot of time, energy and memory. We developed this decentralized application much easier because we benefited from reliability and scalability of this API without being worried about the infrastructure.

Chapter 4

Performance Evaluation

In Chapter three, an architecture proposed and the implementation discussed later. In this chapter the main goal was to evaluate the system performance to determine whether it can answer the questions has been raised in chapter one. The main goal in this work is to visualize data which is retrieved from blockchain. Based on the goal, the evaluation was set up to measure the user's experience if they want to view the visualization. As discussed in chapter one two main challenges were as follows:

- How data can be retrieved to be used in visualization?
- How to represent data to the general users?

As discussed earlier, after being recorded on the blockchain, data is retrieved using Web3, a JavaScript library to interact with the blockchain, so the first challenge has been solved. For the second challenge, Google Map API as a third party is used for data representation to general users to explore data provenance in the system. Data that is being retrieved from blockchain is meaningless and not understandable in any way to the general users. Thus, when data is retrieved and being showed on Google Map API, user experience when viewing the visualization needs to be evaluated. In order to evaluate and assess the performance of the system, a key performance metric has been measured on the Local Area Network:

- **Average response time** Average response time is the time that server responds to a request which is measured in milliseconds (ms)the response time which

Delay in response time, can negatively impact performance and more importantly user experience and it can end in HTTP timeout. We tried to measure response time in different situations. In the first situation the system performance was compared with one and three servers responding to the clients requests. This method can show system's scalability properly which can be used to improve system performance. Number of clients sending requests is an other investigation metric as it can change the performance. System with fewer number of clients responds better than a system with more number of clients. The performance is tested with 1, 30, 60, 150, 300 number of clients sending requests to the servers to get an overview of possible effects of clients(users) load on the performance. To make the network more real, three different delays were considered in each situation since delays always exist in real networks and all the clients do not send requests to the servers simultaneously. 0ms, 250ms, and 500ms are the delays that were considered in the third situation. Figure 4.1 shows the technological stacks which be the same for the experiments.

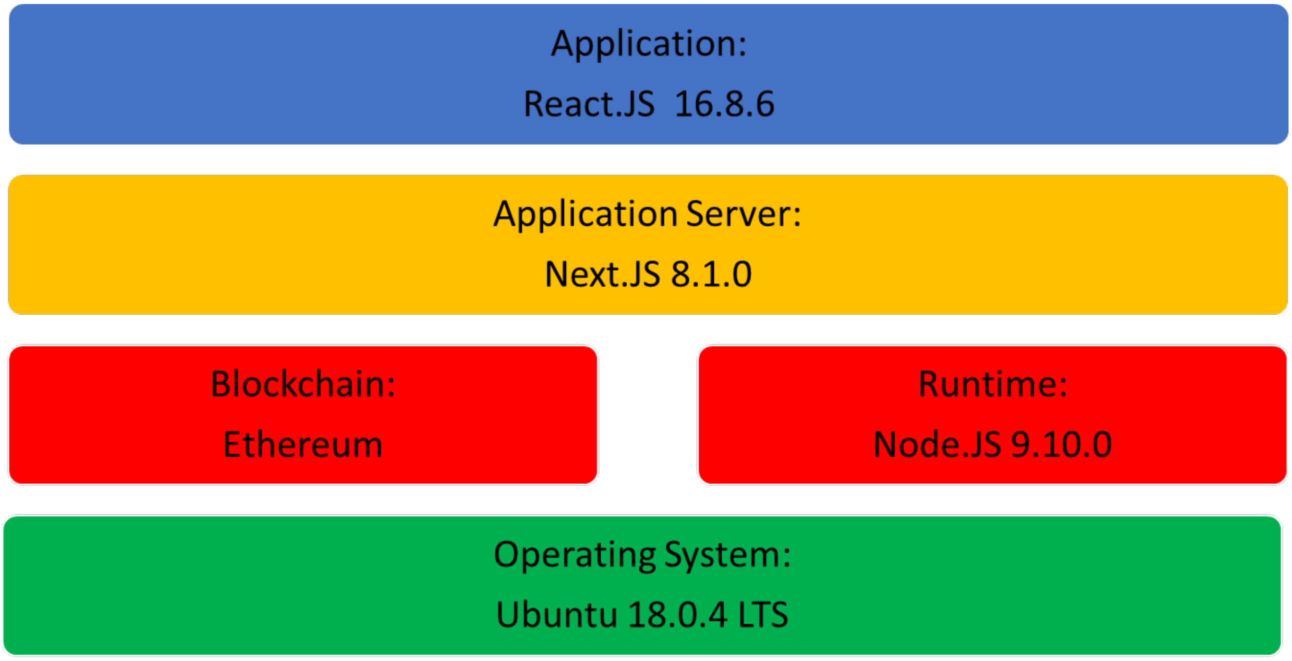


Figure 4.1: Technological Stacks

Apache JMeter 5.0; an open source Java application software was downloaded and installed on Linux machine as the main evaluation tool to get a better insight into affects of number of clients on the system performance. In order to simulate a larger load on the server and stress the server by enough users, Distributed Testing in JMeter was conducted. As the first step, three Ubuntu systems were created in Oracle VM VirtualBox 6.0. Then, a network is created with these Ubuntu nodes in which implemented application was installed on all three nodes.

Table 4.1 Shows the client and three node machine specifications.

Operating System	Ubuntu 18.04.2 LTS
System type	64-bit Operating System, x64-based processor
Processor	Intel Core i7-7700HQ CPU @ 2.80GHz 8
Memory	16GiB
Operating System(Node one)	Ubuntu 18.04.2 LTS
System type	64-bit Operating System, x64-based processor
Memory	2 GiB RAM
Operating System(Node two)	Ubuntu 18.04.2 LTS
System type	64-bit Operating System, x64-based processor
Memory	2 GiB RAM
Operating System(Node three)	Ubuntu 18.04.2 LTS
System type	64-bit Operating System, x64-based processor
Memory	2 GiB RAM

Table 4.1: System Specifications

After preparing the client and three Ubuntu nodes for distributed testing, 1 client was administrated to hit the three servers. Afterwards, by changing the number of clients to 30, 60, 150, 300 and 480 we measured the servers response time. Since we wanted to evaluate the system scalability, we compared the performance of the system when different number of clients hit three servers versus the time that only one server responded to the clients. We estimated that, adding more servers can have a positive impact on average response time as it make the system more scalable. So to evaluate our estimate, we recorded response time when the same number of clients send request to one server to view the visualization. As it is mentioned previously, in the proposed system, seller can register a product by entering its information and deposit money to sell it. On the other hand, buyer can view the summary of product, deposit money to buy the product, view the provenance of the product. Since the process of buying and transferring ownership of a product requires payment which is done through online wallet called MetaMask, we could not test system performance of the system in those operations. On top of that, the main focus of this thesis is to provide the user with a visualization of products provenance on the blockchain; so we decided to evaluate the system when user wants to read information from the blockchain.Figure 4.2 show the local environment layout.

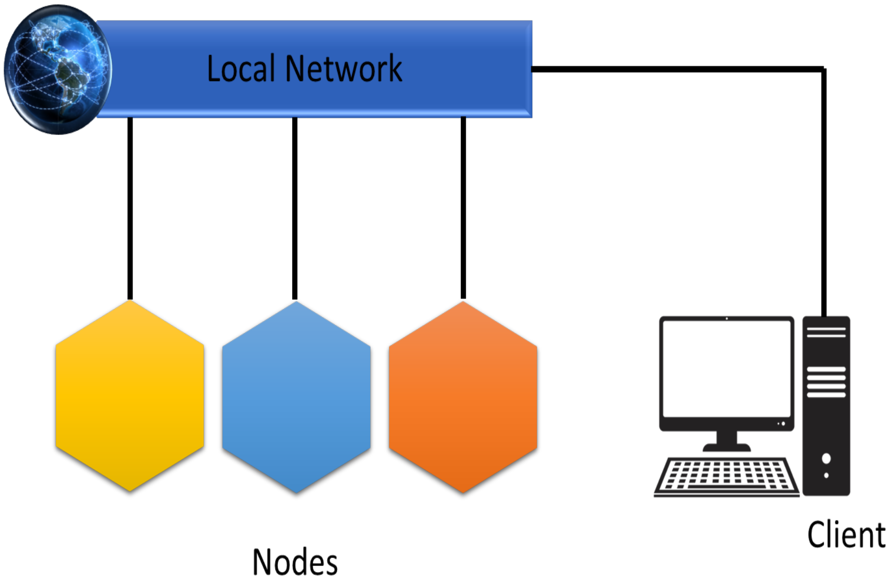


Figure 4.2: Local environment setup

4.1 Experiment One: Average Response Time in the LAN Without Delay

When user wants to view visualization and read information from the blockchain, the average response time is almost the same in both situations, one server responding versus three servers responding to the requests. When only one client sends request, response time is 864ms when there are three servers versus to 870ms for one server. As it can be seen from figure, response time reaches to 7447ms when 480 clients hit the servers while it takes 9072ms when 480 clients hit one server. The gap between these two configurations demonstrates that adding more servers improves horizontal scalability. In other words, three servers, response better which impact average response time as an important metric.

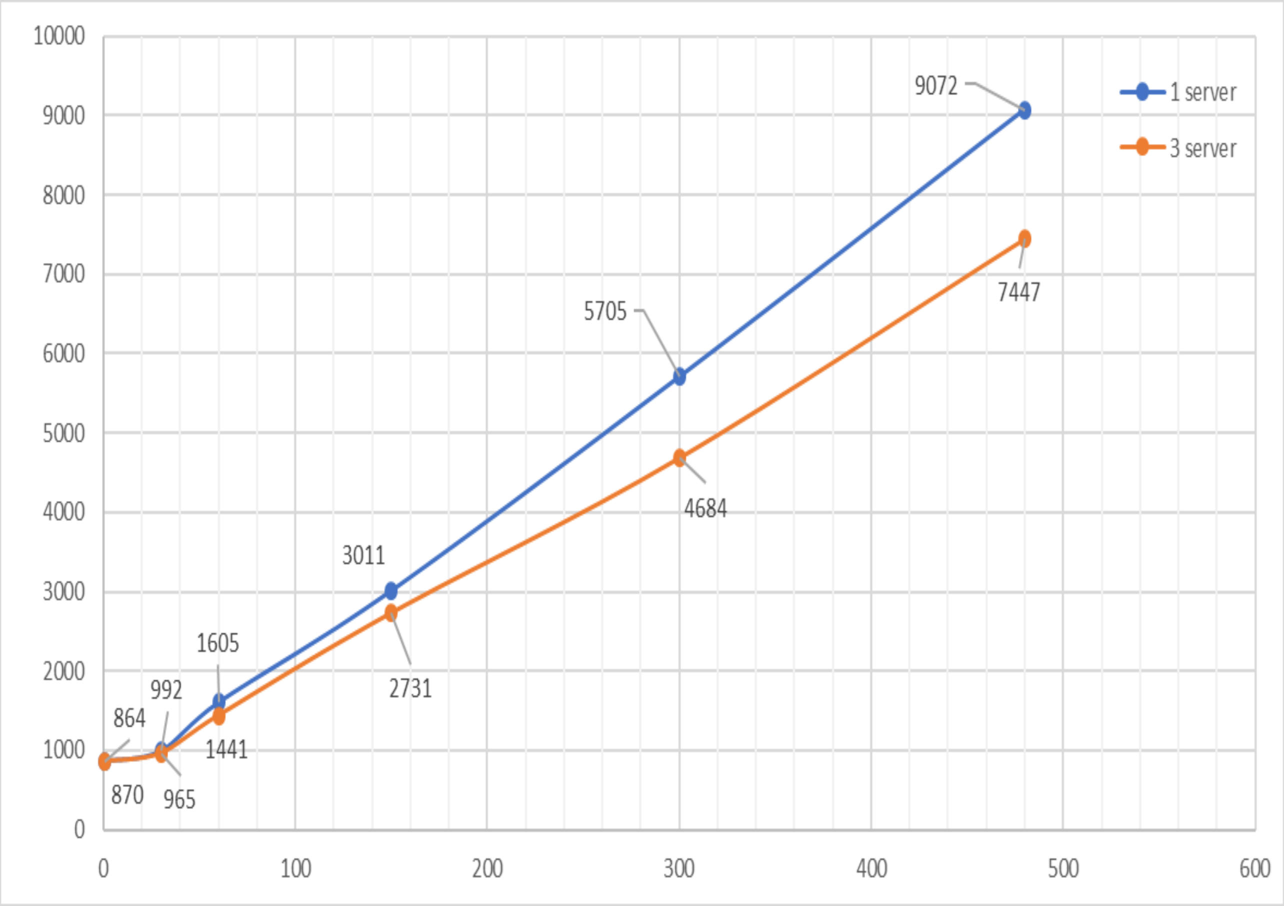


Figure 4.3: Average Response Time Without Delay

4.2 Experiment Two: Average Response Time in the LAN With 250ms Delay

The average response time, improved slightly when we evaluated the performance with 250ms delay. The response time starts at 824ms when there are three servers network versus to 855ms when one server is responding to one client connecting the network to view the products provenance. Comparing the performance of the system without delay and with 250ms delay a slight change in performance can be seen. For instance, when we evaluated the system without delay the average response time was 864ms for three servers network versus to 870ms for one server. The values improved slightly to 824ms versus 855ms When the number of clients increases to 480, the average response time reaches 7345ms and 8810ms for a single and three servers network. Similarly, the response time improved from 7447ms versus 9072ms to 7345ms versus 8810ms.

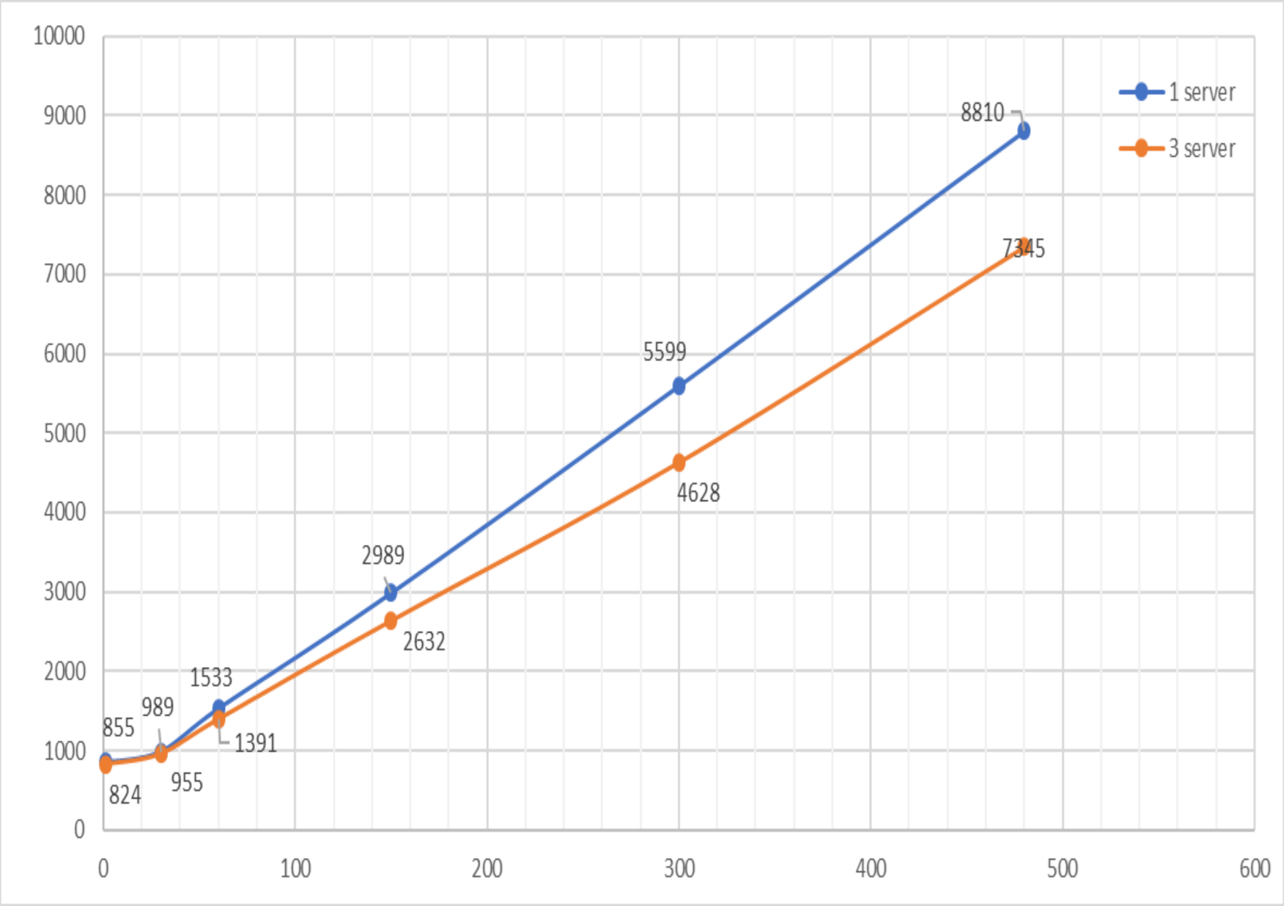


Figure 4.4: Average Response Time With 250ms Delay

4.3 Experiment Three: Average Response Time in the LAN With 500ms Delay

In the last experiment we measured the average response time with 500ms delay. As it is clear from the figure, the performance did not change very much compared to 250ms delay; it has a small yet positive impact on the performance. The average response time when there is only one client sending request is identical which is 827 for both one server and three servers. The difference between the two values gets larger when there are more number of clients; it took the three servers 6947ms to respond to 480 clients simultaneously and when there was only one server responding to the clients, it took 8785ms which is much higher than three servers. In short, comparing the result of the performance with 250ms and 500ms delay indicates that the higher delay resulted in little difference.

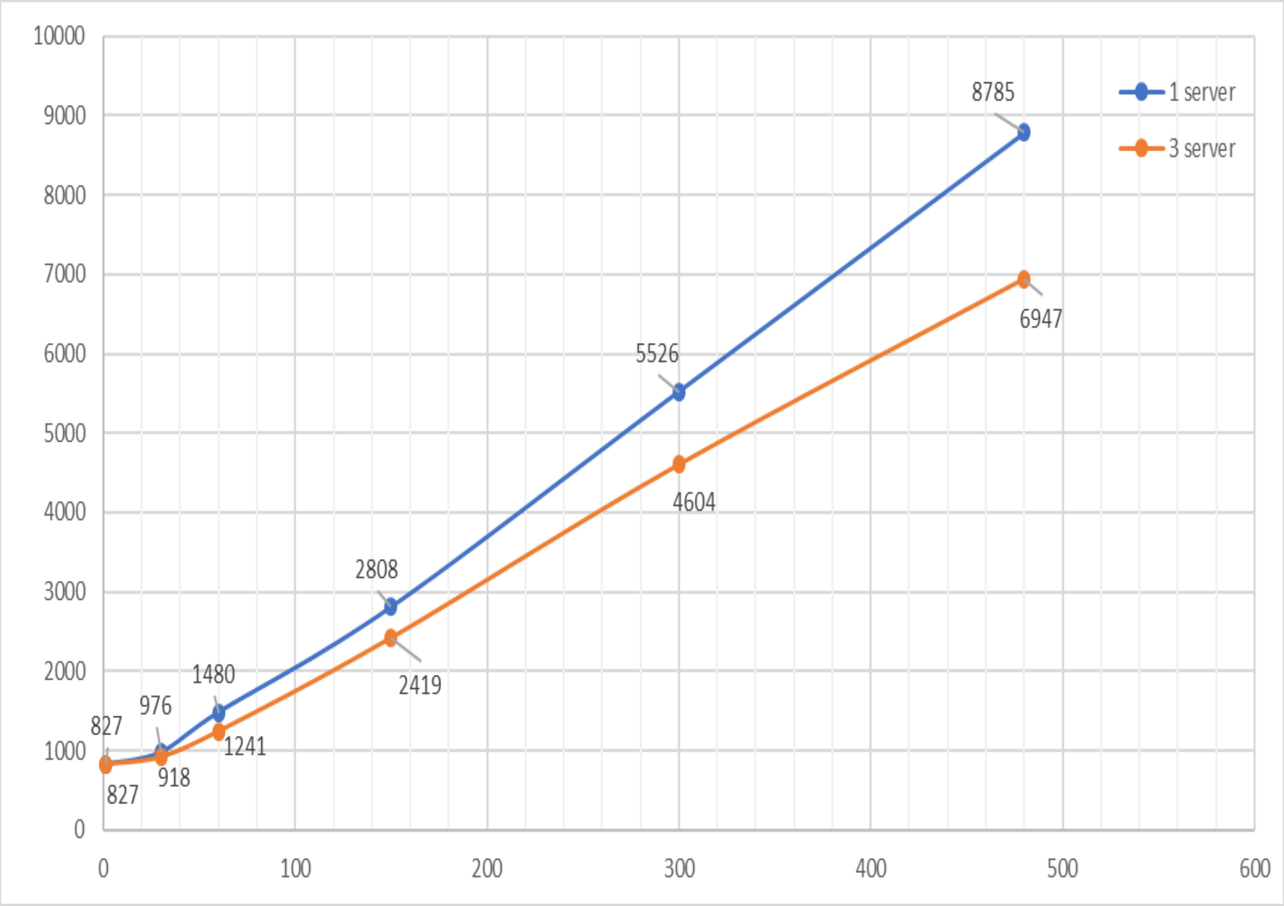


Figure 4.5: Average Response Time With 500ms Delay

4.4 Summary

We compared performance of the system when running in one server responding to the requests versus three servers. Based on the findings of both configurations, at low client loads servers respond better and when the number of clients grow it takes more time to respond to the requests. For instance, it takes 7447ms for three servers to respond to 480 clients sending requests simultaneously but it takes 864ms to respond to one client only. On top of that, we found out that more number of servers impact the scalability of the system effectively.

Chapter 5
Conclusion and Future Work

The most important element in Visualization is data. Using data that has been collected from traditional systems cannot be reliable since there is no guarantee that data has not been manipulated somehow. The whole data is in control of one single authority which can be considered as the bottleneck that can be get down easily. When it comes to visualization in supply chain, not having control over data has become a serious concern since all the involved parties dealing with data continuously. In a situation where every one in the network access their own data, data accuracy can be authenticated and that data can be reliable to be used in visualization. Technology evolves quickly and business owners are required to adopt new technologies to answer demands properly. Blockchain is still a new technology that can bring various benefits to create a meaningful visualizations:

- **Network is Decentralized** The need of third-party intermediary has been eliminated; so data is not in control of only one authority.
- **Data is Immutable** Data can be only entered; it cannot be altered or deleted by the participants in the network which makes the records permanent.
- **Network is Transparent** All the records that has been saved in the network are available to view which increases visibility and makes the whole system trustworthy and reliable.

In this thesis, a review has been done on relevant literature and we discussed the potential benefits of adopting blockchain to supply chain management and providing visualization. A solution was proposed to visualize provenance of products that were created in the system. The detailed architecture and its implementation were discussed afterwards and finally the performance was evaluated and discussed.

In this work we aimed to increase transparency by providing provenance of products. We mainly focus on transparency because it brings multiple benefits to all the entities in the supply chain. For instance, consumers are the end entity in the supply chain that only deal with store. By increasing transparency, the store is confident that products are acquired honestly without faking data. The customers also has this option that they can browse product history since the data is not private and it is available to all the customers.

5.1 Contributions

As explained in the literature in the chapter three, there are lots of successful and widely-used supply chain visualizations. However, they are mainly over centralized configuration networks and in those research that developed supply chain network over blockchain, visualization were missed out. The main goal of this research was to provide a transparent system which is in need in today's supply chain systems. Providing provenance along with reliable data can increase customer's trust as the end point in the supply chain. In this thesis a supply chain tool is provided to trade goods so the actors in the system can buy and sell products in a trustable manner. We used the system to record data about products permanently over blockchain to provide provenance visualization to allow user explore previous owners and location of a specific product. In order to visualize provenance of registered products in the system, multiple technologies were used. Infura API is one the technologies that has been introduced a few years ago. With the help of Infura we could connect to Rinkeby Network with emulates exact Ethereum behavior to save time, energy and cost. Smart contract or the core of the whole system is developed in Solidity, the specific programming language to develop decentralized applications in Ethereum. For user registration and authentication we used an extension called MetaMask that allows us to interact with the Ethereum network within the browser. Web3 is an important JavaScript library that is used to make a connection between the blockchain and the front-end. Finally, another JavaScript module named React, was easy-to-use library to generate interactive frond-end pages.

5.2 Limitations and Future Work

The current work was mainly focused on visualizing products provenance in a supply chain. We tried to suggest a potential solution using Ethereum Platform as a distributed network. There are multiple aspects in this work that could be added or improved for the future work that we mentioned below.

- **Re-implementing the Application Over Other Platforms** We created our decentralized application over Ethereum blockchain. There are various platforms that can be replaced with some changes including: Hyperledger and Multichain. Creating the same application over other platforms allows us to compare the efficiency of these platforms.

- **Using more than three server nodes** In the performance evaluation we assessed the system response over only three nodes due to resource limitations. Adding more servers can give us a better insight into scalability of the system.

- **Reputation** There are different parties involved in a supply chain that have a direct connection with each other. Adding reputation to the system can be an incentive mechanism for all of the network participants to improve service quality.

- **Analyzing Products' Information** In this system, there are different users that enter information about the product they want to sell. By having real data, analyzing data about different products can give organizations a direction for future production. For instance, popularity of a product can be examined; or price fluctuation in time. Another analysis can be done on location; to find out seller at which locations sold their products more.

- **Adding Information Authentication** Right now in this thesis, users enter the information themselves. By using RFID and barcodes on the product, entering information would be automatic and more trustworthy.

References

[1] A. Ashoor and K. Sandhu, "Blockchain infrastructure acceptance in the gulf cooperation council countries: An overview," *RJSH*, p. 55, 2014.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[3] S. A. Abeyratne and R. P. Monfared, "Blockchain ready manufacturing supply chain using distributed ledger," 2016.

[4] F. Tian, "An agri-food supply chain traceability system for china based on rfid & blockchain technology," in *Service Systems and Service Management (ICSSSM), 2016 13th International Conference on*, pp. 1–6, IEEE, 2016.

[5] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Untrusted business process monitoring and execution using blockchain," in *International Conference on Business Process Management*, pp. 329–347, Springer, 2016.

[6] J. Gao, K. O. Asamoah, E. B. Sifah, A. Smahi, Q. Xia, H. Xia, X. Zhang, and G. Dong, "Gridmonitoring: Secured sovereign blockchain based monitoring on smart grid," *IEEE Access*, vol. 6, pp. 9917–9925, 2018.

[7] J. T. Mentzer, W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith, and Z. G. Zacharia, "Defining supply chain management," *Journal of Business logistics*, vol. 22, no. 2, pp. 1–25, 2001.

[8] D. Galvin, "Ibm and walmart: Blockchain for food safety," *PowerPoint presentation*, 2017.

[9] S. New, "Mcdonalds and the challenges of a modern supply chain," *Harvard Business Review Digital Articles*, 2015.

[10] L. U. Opara, "Traceability in agriculture and food supply chain: a review of basic concepts, technological implications, and future prospects," *Journal of Food Agriculture and Environment*, vol. 1, pp. 101–106, 2003.

[11] P. F. Skilton and J. L. Robinson, "Traceability and normal accident theory: how does supply network complexity influence the traceability of adverse events?," *Journal of Supply Chain Management*, vol. 45, no. 3, pp. 40–53, 2009.

[12] I.-H. Hong, J.-F. Dang, Y.-H. Tsai, C.-S. Liu, W.-T. Lee, M.-L. Wang, and P.-C. Chen, "An rfid application in the food supply chain: A case study of convenience stores in taiwan," *Journal of food engineering*, vol. 106, no. 2, pp. 119–126, 2011.

[13] A. Kassahun, R. Hartog, T. Sadowski, H. Scholten, T. Bartram, S. Wolfert, and A. Beulens, "Enabling chain-wide transparency in meat supply chains based on the epcis global standard and cloud-based services," *Computers and electronics in agriculture*, vol. 109, pp. 179–190, 2014.

[14] C. N. Verdouw, J. Wolfert, A. Beulens, and A. Rialland, "Virtualization of food supply chains with the internet of things," *Journal of Food Engineering*, vol. 176, pp. 128–136, 2016.

[15] W. Wang, Y. Liu, and S. Wang, "A rfid-enabled tracking system in wire bond station of an ic packaging assemble line," in *RFID-Technology and Applications (RFID-TA), 2010 IEEE International Conference on*, pp. 170–175, IEEE, 2010.

[16] D. Folinas, I. Manikas, and B. Manos, "Traceability data management for food chains," *British Food Journal*, vol. 108, no. 8, pp. 622–633, 2006.

[17] C. Shanahan, B. Kernan, G. Ayalew, K. McDonnell, F. Butler, and S. Ward, "A framework for beef traceability from farm to slaughter using global standards: an irish perspective," *Computers and electronics in agriculture*, vol. 66, no. 1, pp. 62–69, 2009.

[18] V. Mattoli, B. Mazzolai, A. Mondini, S. Zampolli, and P. Dario, "Flexible tag datalogger for food logistics," *Sensors and Actuators A: Physical*, vol. 162, no. 2, pp. 316–323, 2010.

[19] E. Abad, F. Palacio, M. Nuin, A. G. De Zarate, A. Juarros, J. M. Gómez, and S. Marco, "Rfid smart tag for traceability and cold chain monitoring of foods: Demonstration in an intercontinental fresh fish logistic chain," *Journal of food engineering*, vol. 93, no. 4, pp. 394–399, 2009.

[20] M. Friendly, "A brief history of data visualization," in *Handbook of data visualization*, pp. 15–56, Springer, 2008.

[21] P. Shah and E. G. Freedman, "Bar and line graph comprehension: An interaction of top-down and bottom-up processes," *Topics in cognitive science*, vol. 3, no. 3, pp. 560–578, 2011.

[22] M. Galesic and R. Garcia-Retamero, "Statistical numeracy for health: A cross-cultural comparison with probabilistic national samples," *Archives of internal medicine*, vol. 170, no. 5, pp. 462–468, 2010.

[23] S. Neuner-Jehle, O. Senn, O. Wegwarth, T. Rosemann, and J. Steurer, "How do family physicians communicate about cardiovascular risk? frequencies and determinants of different communication formats," *BMC family practice*, vol. 12, no. 1, p. 15, 2011.

[24] I. M. Lipkus, "Numeric, verbal, and visual formats of conveying health risks: suggested best practices and future recommendations," *Medical decision making*, vol. 27, no. 5, pp. 696–713, 2007.

[25] Sourcemap, "End-to-end supply chain visualization white paper," tech. rep., [Accessed May 08, 2019].

[26] G. Bhosle, P. Kumar, B. Griffin-Cryan, R. van Doesburg, M. Sparks, and A. Paton, "Global supply chain control towers: Achieving end-to-end supply chain visibility," *Capgemini Consulting White Paper*, 2011.

[27] J. U. Min and H. Bjornsson, "Construction supply chain visualization through web services integration," tech. rep., CIFE TR, 2004.

[28] M. Barratt and A. Oke, "Antecedents of supply chain visibility in retail supply chains: a resource-based theory perspective," *Journal of operations management*, vol. 25, no. 6, pp. 1217–1233, 2007.

[29] A. Ilic, T. Andersen, and F. Michahelles, "Increasing supply-chain visibility with rule-based rfid data analysis," *IEEE Internet Computing*, vol. 13, no. 1, pp. 31–38, 2009.

[30] L. Bonanni, "Sourcemap: eco-design, sustainable supply chains, and radical transparency," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 4, pp. 22–26, 2011.

[31] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.

[32] T. H., "How blockchain works," [Accessed March 5, 2019].

[33] M. Mainelli and M. Smith, "Sharing ledgers for sharing economies: an exploration of mutual distributed ledgers (aka blockchain technology)," 2015.

[34] J. Bunge, "Latest use for a bitcoin technology: Tracing turkeys from farm to table." https://www.wsj.com/articles/latest-use-for-a-bitcoin-technology-tracing-turkeys-from-farm-to-table-1508923801, 2017[Accessed November 7, 2018].

[35] M. Samaniego and R. Deters, "Detecting suspicious transactions in iot blockchains for smart living spaces," in *International Conference on Machine Learning for Networking*, pp. 364–377, Springer, 2018.

[36] J. Liebkind, "Coca-cola and us state dept use blockchain to combat forced labor." https://www.investopedia.com/news/cocacola-and-us-state-dept-fight-forced-labor/, 2018[Accessed November 7, 2018].

[37] M. Samaniego and R. Deters, "Blockchain as a service for iot," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 433–436, IEEE, 2016.

[38] C. Group, "Carrefour launches europe's first food blockchain." http://www.carrefour.com/current-news/carrefour-launches-europes-first-food-blockchain, 2018[Accessed December 22, 2018].

[39] M. Samaniego and R. Deters, "Internet of smart things-iost: Using blockchain and clips to make things autonomous," in *2017 IEEE international conference on cognitive computing (ICCC)*, pp. 9–16, IEEE, 2017.

[40] C. Brannigan, "Introducing downstream the worlds first blockchain beer." https://www.irelandcraftbeers.com/blog/view/42/introducing-downstream-the-worlds-first-blockchain-beer., 2017[Accessed December 22, 2018].

[41] M. Samaniego, C. Espana, and R. Deters, "Access control management for plant phenotyping using integrated blockchain," in *Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure*, pp. 39–46, ACM, 2019.

[42] A. Mole, "Blockchain technology to fight food fraud." https://research.qut.edu.au/blockchain/2018/10/04/blockchain-technology-to-fight-food-fraud/, 2018[Accessed December 22, 2018].

[43] M. Samaniego, C. Espana, and R. Deters, “Smart virtualization for iot,” in *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 125–128, IEEE, 2018.

[44] G. R. F. Coop, “Transparency from pastures to plate grass roots farmers’ co-op.” `https://grassrootscoop.com/transparency-tools/`, 2018[Accessed December 22, 2018].

[45] M. Samaniego and R. Deters, “Supporting iot multi-tenancy on edge devices,” in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 66–73, IEEE, 2016.

[46] G. Zyskind, O. Nathan, *et al.*, “Decentralizing privacy: Using blockchain to protect personal data,” in *Security and Privacy Workshops (SPW), 2015 IEEE*, pp. 180–184, IEEE, 2015.

[47] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “Medrec: Using blockchain for medical data access and permission management,” in *Open and Big Data (OBD), International Conference on*, pp. 25–30, IEEE, 2016.

[48] G. Prisco, “The blockchain for healthcare: Gem launches gem health network with philips blockchain lab,” *Bitcoin Magazine*, 2016.

[49] M. Samaniego and R. Deters, “Zero-trust hierarchical management in iot,” in *2018 IEEE International Congress on Internet of Things (ICIOT)*, pp. 88–95, IEEE, 2018.

[50] P. Nichol, “Blockchain applications for healthcare,” 2016.

[51] S. Rouhani and R. Deters, “Performance analysis of ethereum transactions in private blockchain,” in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 70–74, IEEE, 2017.

[52] H. Natarajan, S. K. Krause, and H. L. Gradstein, “Distributed ledger technology (dlt) and blockchain,” *FinTech note*, 2017.

[53] L. A. Linn and M. B. Koo, “Blockchain for health data and its potential use in health it and health care related research,” in *ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST*, 2016.

[54] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in *Big Data (BigData Congress), 2017 IEEE International Congress on*, pp. 557–564, IEEE, 2017.

[55] J. Sousa, A. Bessani, and M. Vukolic, “A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform,” in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 51–58, IEEE, 2018.

[56] G. Konstantopoulos, “Understanding blockchain fundamentals, part 1: Byzantine fault tolerance,” 2017.

[57] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[58] C. Dwork and M. Naor, “Pricing via processing or combatting junk mail,” in *Annual International Cryptology Conference*, pp. 139–147, Springer, 1992.

[59] N. Shi, "A new proof-of-work mechanism for bitcoin," *Financial Innovation*, vol. 2, no. 1, p. 31, 2016.

[60] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International Workshop on Open Problems in Network Security*, pp. 112–125, Springer, 2015.

[61] J. Harm, J. Obregon, and J. Stubbendick, "Ethereum vs. bitcoin," *Creighton University, undated manuscript, retrieved*, vol. 1, 2017.

[62] J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye, and D. I. Kim, "Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks," *IEEE Wireless Communications Letters*, 2018.

[63] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain," 2018.

[64] B. Hill, S. Chopra, P. Valencourt, and N. Prusty, *Blockchain Developer's Guide: Develop smart applications with Blockchain technologies-Ethereum, JavaScript, Hyperledger Fabric, and Corda*. Packt Publishing Ltd, 2018.

[65] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[66] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[67] D. Vujicic, D. Jagodi, and S. Rani, "Blockchain technology, bitcoin, and ethereum: A brief overview," in *INFOTEH-JAHORINA (INFOTEH), 2018 17th International Symposium*, pp. 1–6, IEEE, 2018.

[68] S. Bragagnolo, H. Rocha, M. Denker, and S. Ducasse, "Ethereum query language," in *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, pp. 1–8, ACM, 2018.

[69] E. Team, "Solidity documentation," *Release 0.4*, vol. 11.

[70] N. P. Triantafyllidis and T. Oskar van Deventer, "Developing an ethereum blockchain application," 2016.

[71] M. Alessi, A. Camillo, E. Giangreco, M. Matera, S. Pino, and D. Storelli, "Make users own their data: A decentralized personal data store prototype based on ethereum and ipfs," in *2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–7, IEEE, 2018.

[72] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *Principles of Security and Trust*, pp. 164–186, Springer, 2017.

[73] V. Dhillon, D. Metcalf, and M. Hooper, "Unpacking ethereum," in *Blockchain Enabled Applications*, pp. 25–45, Springer, 2017.

[74] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *International Conference on Financial Cryptography and Data Security*, pp. 357–375, Springer, 2017.

[75] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on computing*, vol. 18, no. 1, pp. 186–208, 1989.

[76] L. Foundation, "Hyperledger." https://www.hyperledger.org/, [Accessed July 27, 2019].

[77] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, p. 30, ACM, 2018.

[78] Hyperledger, "Five hyperledger blockchain projects now in production." https://www.hyperledger.org/blog/2018/11/30/six-hyperledger-blockchain-projects-now-in-production, 2018 [Accessed July 27, 2019].

[79] M. Smith, "In wake of romaine e. coli scare, walmart deploys blockchain to track leafy greens." https://corporate.walmart.com/newsroom/2018/09/24/in-wake-of-romaine-e-coli-scare-walmart-deploys-blockchain-to-track-leafy-greens, [Accessed July 27, 2019].

[80] N. Technologies, "Niit technologies introduces chain-m, a blockchain powered solution for airlines and its partners." https://www.niit-tech.com/news-events/news/niit-technologies-introduces-chain-m-blockchain-powered-solution-airlines-and-its, [Accessed July 27, 2019].

[81] J. C. Blog, "Jd launches blockchain open platform." https://jdcorporateblog.com/jd-launches-blockchain-open-platform/, [Accessed Aug 2, 2019].

[82] A. A. of Insurance Services, "openidl (open insurance data link)." https://www.aaisonline.com/web/guest/openidl, [Accessed Aug 2, 2019].

[83] IBM, "Track pizza shipments." https://developer.ibm.com/tutorials/pizza-on-the-blockchain/, [Accessed July 27, 2019].

[84] C. M. Cap, "Cryptocurrency market capitalizations," *Retrieved on January*, vol. 21, p. 2018, 2018.

[85] A. Yakubov, W. Shbair, A. Wallbom, D. Sanda, *et al.*, "A blockchain-based pki management framework," in *The First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block) colocated with IEEE/IFIP NOMS 2018, Tapei, Tawain 23-27 April 2018*, 2018.

[86] C. Molina-Jimenez, I. Sfyrakis, E. Solaiman, I. Ng, M. W. Wong, A. Chun, and J. Crowcroft, "Implementation of smart contracts using hybrid architectures with on and off–blockchain components," in *2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2)*, pp. 83–90, IEEE, 2018.

[87] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[88] Rinkeby, "Rinkeby faucet." https://www.rinkeby.io/#faucet, [Accessed July 27, 2019].

[89] Next, "Next.js." `https://zeit.co/blog/next`, [Accessed January 09, 2019].

[90] React, "React a javascript library for building user interfaces." `https://reactjs.org/`, [Accessed January 09, 2019].

[91] Web3, "Web3." `https://web3js.readthedocs.io/en/1.0/getting-started.html`, [Accessed January 10, 2019].

[92] MetaMask, "Metamask." `https://metamask.io/`, [Accessed January 10, 2019].

[93] S. Goldfeder, J. Bonneau, R. Gennaro, and A. Narayanan, "Escrow protocols for cryptocurrencies: How to buy physical goods using bitcoin," in *International Conference on Financial Cryptography and Data Security*, pp. 321–339, Springer, 2017.

[94] A. Asgaonkar and B. Krishnamachari, "Solving the buyer and sellers dilemma: A dual-deposit escrow smart contract for provably cheat-proof delivery and payment for a digital good without a trusted mediator," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 262–267, IEEE, 2019.

[95] OpenSea.pro, "How can the shipping industry take advantage of the blockchain technology?." `https://opensea.pro/blog/blockchain-for-shipping-industry,`, [Accessed January 09, 2019].

[96] T. T. Daniel Jones, "Using blockchain to unblock the supply chain," [Accessed May 17, 2019].

[97] B. Cook and W.-N. Zealand, "Blockchain: Transforming the seafood supply chain," *World Wide Fund for Nature*, 2018.

[98] M. Yakubowski, "Walmart requires certain produce suppliers to deploy blockchain technology." `https://cointelegraph.com/news/walmart-requires-certain-produce-suppliers-to-deploy-blockchain-technology`, 2018[Accessed December 22, 2018].

[99] R. L. Blog and N. by Age, "Food safety: Blockchain technology by carrie dennett, mph, rdn, cd today's dietitian vol. 20, no. 6, p. 14," 2018.

[100] A. Chang, "The facebook and cambridge analytica scandal, explained with a simple diagram," *Vox, March*, vol. 23, 2018.