





Article

Blockchain-Based Traceability and Visibility for Agricultural Products: A Decentralized Way of Ensuring Food Safety in India

Deepak Prashar ¹, Nishant Jha ¹, Sudan Jha ¹, Yongju Lee ^{2,*} and Gyanendra Prasad Joshi ^{3,*}

¹ Department of CSE, Lovely Professional University, Punjab 144411, India; deepak.prashar@lpu.co.in (D.P.); nishant.11702196@lpu.co.in (N.J.); sudhan.25850@lpu.co.in (S.J.)

² School of CSE, Kyungpook National University, Daegu 702-701, Korea

³ Department of CSE, Sejong University, Seoul 05006, Korea

* Correspondence: yongju@knu.ac.kr (Y.L.); joshi@sejong.ac.kr (G.P.J.); Tel.: +82-53-950-7285 (Y.L.); +82-2-6935-2481 (G.P.J.)

Received: 1 March 2020; Accepted: 22 April 2020; Published: 24 April 2020



Abstract: The globalization of the food supply chain industry has significantly emerged today. Due to this, farm-to-fork food safety and quality certification have become very important. Increasing threats to food security and contamination have led to the enormous need for a revolutionary traceability system, an important mechanism for quality control that ensures sufficient food supply chain product safety. In this work, we proposed a blockchain-based solution that removes the need for a secure centralized structure, intermediaries, and exchanges of information, optimizes performance, and complies with a strong level of safety and integrity. Our approach completely relies on the use of smart contracts to monitor and manage all communications and transactions within the supply chain network among all of the stakeholders. Our approach verifies all of the transactions, which are recorded and stored in a centralized interplanetary file system database. It allows a secure and cost-effective supply chain system for the stakeholders. Thus, our proposed model gives a transparent, accurate, and traceable supply chain system. The proposed solution shows a throughput of 161 transactions per second with a convergence time of 4.82 s, and was found effective in the traceability of the agricultural products.

Keywords: Ethereum smart contracts; blockchain; traceability; visibility; throughput; supply chain; IPFS

1. Introduction

It is well known that India has a large population and an enormous potential for the growth and success of food companies. India is expected to be the country with the highest GDP very soon. At the same time, however, agriculture, now contributing about 13.7 percent of the GDP in India and contributing 50 percent of the workforce, is gradually declining. Food security concerns have received a great deal of scrutiny lately. Food supply chain is among the high-value businesses globally. However, the scenario states that the supply chain yields some faults. In December 2017, it was claimed by a report that around 65 out of 72 food testing labs of the Food Safety and Standards Authority of India (FSSAI) were illegal [1]. In October 2018, famous food companies like Zomato, Swiggy, and Food Panda removed 10,500 hotels and restaurants from food service, as the food quality in these restaurants and hotels was not up to the mark as per the standard prescribed by the Food Safety and Standards Act of 2006 (FSS Act) [2]. In addition to this, these places did not have the basic food safety controller approvals from the Indian government [3]. The food service industry, such as restaurants

in most of the cities of India, were found serving stale food to the customers. As an example, one of the famous noodle varieties was found to contain lead in the product, which can cause serious health issues [4]; most recently, a reputed chocolate company found plastic in their chocolate product [5]. The worldwide food recalls are increasing every year, and the cost of recalls is also rising for the suppliers [6]. Since the farm-to-fork traceability is hard, the cost of failure is even harder. Apart from keeping an eye on the ingredients only, it has become the utmost necessity to mitigate and monitor the pockets of the consumers too. The reason for this is process visibility, which plays an important role in food safety. Recent developments in food traceability management (TMS) are expected to allow the quality and safety of food to be tracked throughout the cycle from farming to consumption [7,8]. Globally, one in five deaths, equivalent to 11 million deaths per year, is associated with poor diet [8]. The conventional and existing processes combined do not give an efficient end-to-end traceability.

Blockchain technology has the potential to solve this problem efficiently. In real time, food can be traced to its origin, thus improving the health of customers significantly. Using blockchain technology, agricultural products, from farmers to dealers, can be tracked and controlled using maximum visibility throughout the entire process in the chain. A seller selling mutton knows exactly when the animal is slaughtered, when the mutton is shipped, how long the mutton is in shipment, and how long it can be used before it gets spoiled. The retailer has this knowledge in his possession and can follow the product on a trip through the supply chain in real time. The risk of food poisoning can also be traced; thereby, knowing the first details of the food product, doctors can quickly determine where the foulness has occurred, thus saving lives and, ultimately, resources. Eventually, traceability means that consumers and their customers who pay more for checked organic and non-genetically modified organism (GMO) products receive quality products in accordance with their standards. Blockchain technology can eliminate the middlemen involved, allowing farmers to function and connect directly with retailers to raise both parties' sales and profits. This structure can make a huge impact on smaller farms that can serve local sellers. If the seller runs out of potatoes and that void is recognized in the blockchain, then the local farmer can ship potatoes to the seller immediately, filling the gap quickly. Blockchain enables the deal to occur without any telephone call or order form. The farmer notes the shortage of potatoes and updates the blockchain to show that they will fill the order, the blockchain is updated, and the seller is ready to receive the potatoes. New markets can open for small farms, including local firms and marketplaces. Blockchain technology will greatly improve performance by replacing multiple stages in the negotiation phase so that demands are addressed precisely. One of the technologies is actually being used to eliminate the need for a centralized network in the supply chain for livestock, providing full exposure of the market for the goods of the field, and allowing quicker accomplishment and increased sales for both sellers and producers.

The food business has been revolutionized in recent years despite several challenges; some of them are the non-transparent and inefficient non-communicating networks formed of processes, products, and data. Lack of transparency makes it difficult to achieve fair pricing and quality of products. The need for data-driven agriculture has resulted from regulating pressure, fraud, and food crisis. Blockchain offers both merchants and customers an open platform to haggle for the best prices for their items. This helps distributors to make payments directly to consumers, avoiding traders and intermediaries. Concerned parties can transfer funds to agri-business partners in other countries without worrying about scamsters. A supply chain is more transparent and visible with blockchain. It will help in food visibility and safety responsiveness, as all details of food origin are at the fingertips. Therefore, this provides the exact information that the consumer is expecting, from digital ID management to payment systems. Blockchain-based solutions push the new era in inclusive growth. Although blockchain technology can solve the major problems in the food and agriculture sector more efficiently, the limitation of this technology is connecting feasible business models and captivating cases. It is believed that in the near future, blockchain technology will play an important role in the agricultural sector.

The main objective of the work suggested in this paper is to enable the real-time monitoring of the supply chain by using blockchain, thus bringing transparency in the food supply industry. The contributions of the proposed approach in this paper are as follows.

- This study explores the design and implementation of a traceability system using blockchain at its core.
- It introduces Ethereum-based smart contracts for the agricultural supply chain.
- It investigates and defines the roles of major contributors involved in the system.
- It analyzes and evaluates the performance of the system based on various factors, such as hardware, block timing, network latency, and bandwidth.

The rest of the paper is organized as follows. Section 2 deals with the related work. Section 3 deals with the prerequisites for designing the system, followed by Section 4, which deals with the system design and architecture. Section 5 deals with the implementation of the system. Section 6 deals with the analysis of the system. Section 7 deals with the results, followed by Section 8, which concludes the paper.

2. Related Work

In this section, we review the works reported in the literature on the implementation of blockchain in food supply chains. While research work on blockchain technology has been growing slowly in the banking and financial industries, research on the supply chain for food and agriculture is limited and has just begun to gain recognition. Dai et al. [9] investigated the integration of Internet of things (IoT) and 5G technology with blockchain. Reference [10] represents a detailed analysis of Blockchain of Things (BCoT), including the challenges they faced in implementation of the technology. The authors pointed out the major challenges, including security vulnerabilities, network complexity, and heterogeneity of IoT systems. In the proposed system, we have addressed some of the challenges indicated by the authors.

The food safety chain traceability schemes are based on risk recognition and significant levels of management, and are discussed by Ustundaga and Tanyasb [10]. At the Echelon point, the authors analyzed ways of influencing the efficiency of the interconnected radio-frequency identification (RFID) supply chain in terms of cost considerations through product value, time, and market uncertainties. In order to achieve the anticipated advantages of using an RFID-based system in the supply line by efficiency improvements in control, protection, accuracy, and visibility, they implemented a simulation model. Tian [11] proposed a food supply chain with hazard analysis critical control points (HACCP)-based blockchain and IoT technologies. In another research by the same author, Tian [12], an RFID- and blockchain-based agri-food supply chain traceability system and the process of this system are described.

Some early blockchain pilot implementation instances of the food and agriculture offer chain include wheat trading enabled by Agridigital blockchain maltreatment technology in Australia. Kamath [13] described how Walmart is tackling food safety in the supply chain using IBM's blockchain solution based on Hyperledger Fabric. Tripoli and Schmidhuber [12] address distributed ledger technology (DLT) equipment and sensible contracts to improve control and provide traceability in agriculture. The authors recognize any technical challenge and hurdle to acceptance to infer that DLTs have critical potential to achieve property development objectives. For our study, we explored the framework of Aung and Chang [14] to position the various ideas and characteristics of food traceability with respect to the various food supply chain players, as shown in Figure 1. In the framework, the supply chain comprises different stakeholders and describes the method of delivery via internal traceability and external traceability. Mao et al. [15] advocate a syndicated blockchain solution to the economic food trading system. The authors presented an integrated Byzantine Fault Tolerance (iPBFT) definition to boost the food supply chain's commodity exchange portfolio.

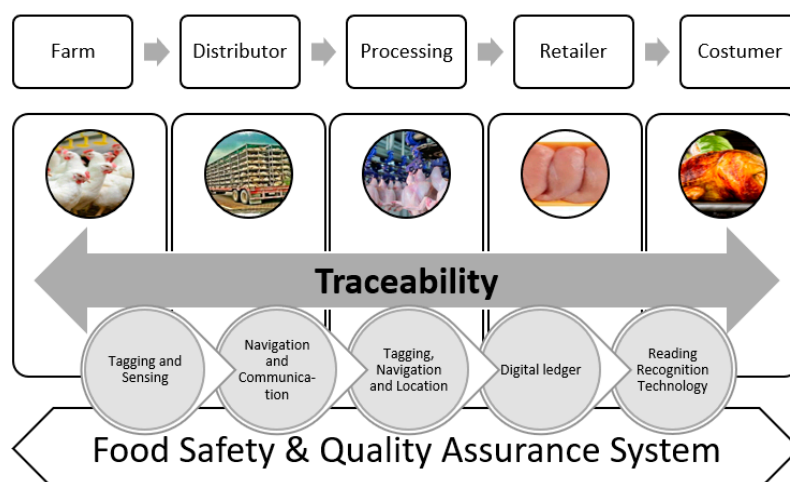


Figure 1. Structure of a food traceability system.

Arena et al. [16] presented a blockchain based application, “BRUSCHETTA”, for certification and traceability of the extra virgin olive oil (EVOO) supply chain in Italy. It provides a blockchain-based solution to implement the certification of EVOO by tracking the whole production process. However, the process is a bit complicated compared to our proposed solution. Aich et al. [17] highlighted the variations between a traditional supply chain and blockchain-based supply chain. The authors explained the advantages of using blockchain technology in various sectors, including automobile, pharmaceutical, food industry, and retail. They also discussed the problems faced by these industries. This study only discusses the theoretical aspects of blockchain technology instead of focusing much on the implementation part. However, in this work, we focus on the practical framework and implementation of blockchain in food supply chain industry. Shakhbulatov et al. [18] have shown the implementation of blockchain for tracking carbon footprints of food production and stages of transportation. This work resembles our proposed solution in this manuscript very much. However, the above-mentioned solutions are much more complicated than our proposed system.

Lucena et al. [19] presented the application of blockchain technology in real-world contexts to ensure monitoring of grain efficiency. The results of their study support that a blockchain-based certification can add a 15 percent valuation for GM-free soy in the grain exporter business network in Brazil. The decentralized smart contract system known as Hawk is introduced by Kosba et al. [20]. The system Hawk does not record financial transactions on the blockchain directly, and therefore retains general transactional anonymity. In Hawk, programmers can write a private smart contract without implementing cryptography. The system automatically generates cryptographic primitives. Yang et al. [21] developed a traceability system for vegetable products based on the .net framework. It integrates pre- and postproduction management, a short messaging system, feedback, and health guidance for consumers. A simulation-based prototype was developed by Sari [22] for studying the impact of RFID technology on supply chain performance. The study results show that the implementation of RFID in a supply chain provides significant advantages when collaborating intensively with stockholders. An article by Foroglou et al. [23] shows that blockchain technology can be used in various areas, including banking, contracts, elections, intellectual property rights (IPR), and smart properties. Nonetheless, they dealt with all aspects of supply chain management and noticed only the distribution and storage processes.

Sayogo et al. [24] identified the challenges and motivating factors for data openness faced by data producers. Such threats and driving factors are grouped into four main areas that affect intelligent communication. The results show that marketers should realize that consumers know the product they purchase. As discussed above, there is a growing trend in the use of blockchain technology for more data protection, accountability, verification of various food requirements, and supply chains for

agricultural products. Much of the literature addresses the theoretical use of blockchain in agricultural supply chains concerning a certain application method or strategy. Blockchain is being used rapidly to improve data security, efficiency, and authentication of food supply chains with different criteria.

A large portion of the literature discusses the possible use of blockchain technology with a certain strategy in agricultural supply chains. This paper concentrates on the different ways by which the blockchain can be introduced in the agricultural sector for the benefits of farmers, consumers, and distributors collectively.

In this paper, we present a framework that is less complex and requires less expertise to operate.

3. Preliminaries

3.1. Blockchain Technology

After the launch of the first cryptocurrency, the Bitcoin, developed based on blockchain in 2009, a number of new blockchains have been created. The common denominator in all of these blockchain solutions is that they have an underlying connected block data structure and are distributed in varying degrees. The original vision of Bitcoin was a decentralized, distributed system that was unregulated, i.e., everyone can participate in the system, and safe enough to be used as a payment network. This required trade-offs in terms of processing speed, processing latency, and ensuring final state adjustments. Such trade-offs are not appropriate in certain cases, so other approaches were developed that deal with these drawbacks, but not without having their own trade-offs. Such trade-offs often revolve around the decentralized and unregulated existence of blockchain technologies, essentially making them more similar than the original Bitcoin blockchain to existing distributed networks. All of these attributes and trade-offs are derived from consensus algorithm preference.

3.2. Consensus Algorithm

Consensus algorithms are an important component of blockchain technology. These algorithms are used when interconnected computers want to interact and cooperate for a common purpose in a distributed system, also called distributed computing. In blockchains, the goal is typically to preserve and update an immutable shared state, mediate transactions, and provide a stable computing engine. The process of consensus building and maintaining among distributed entities, or nodes, can be divided into two components. The first component is where state machine replicas run on all nodes. It drives the essence of the service planned for delivery. The second component consists of the actual consensus algorithm, whose task is to propagate the requests between the nodes so that each node can execute them in the same order.

3.3. Network Model

To be considered feasible for a state machine replication method to be a consensus algorithm, it must guarantee protection and liveness. Protection means that the algorithm should always be right during normal execution and in the presence of a given fault model, and liveness indicates that at some point in time, every execution should be completed correctly. However, as Fischer et al. [25] found, this cannot be assured in an asynchronous network model, where messages can be delayed indefinitely. In a synchronous network model, where messages always arrive in the appropriate order, it is possible to guarantee that. For the blockchains, a synchronous network cannot be ensured because of the geographic distance between the distributed nodes; one power outage may result in a network partitioning, resulting in messages being unable to be transferred between partitions. To counter this problem, the eventual-synchrony network model is introduced by Dwork et al. [26]. This model assumes that asynchronous networks can inevitably become synchronous under a certain time limit. Nowadays, this model is commonly accepted in the design of scalable distributed systems.

3.4. CAP Theorem

The CAP theorem states that only two of the following three properties can be satisfied in a distributed system: Consistency (C), availability (A), and tolerance of partitions (P). Consistency is achieved if a program can guarantee that every read receives the most recent state or error, e.g., if performed concurrently, two clients will not receive two different values for the same request. Availability requires that every request always receives a non-error response. Partition tolerance states that even though messages cannot be transmitted to all nodes in the network, a machine must be able to continue running. Since the theorem states that only two out of three properties can be obtained simultaneously, a system can be either CA, CP, or AP. However, it is more complex than this, and the need for partition tolerance cannot be ignored most of the time, essentially making a CA scheme unfeasible. This is particularly true for blockchains, because one fundamental idea is to spread the nodes globally to increase decentralization. The same illustration with a power outage at the wrong position, as explained earlier, can also be used to highlight the need for partition resistance.

3.5. Smart Contracts

Smart contracts are a computerized transaction protocol that implements the terms of a contract. Smart contracts are used to convert contractual clauses into code and integrate them into hardware or software so that the code gets implemented by itself. This eliminates the need for trusted intermediaries. Smart contracts are the scripts stored in the blockchain defined by a unique address. Ethereum is the first blockchain of its type developed as a smart contract platform. In Ethereum, the Ethereum virtual machine (EVM) executes smart contracts and provides a complete programming language for Turing. A protocol is implemented to restrict the resources that each contract requires, because every single operation is performed by each node in the network. Therefore, each process has a cost calculated in gas, and each gas unit consumed by a transaction is charged in a cryptocurrency called Ether. Gas price refers to how much Ether it costs for each gas unit. Because each block used in the blockchain has a gas cap, in transactions using a lot of gas caps, the amount can be included in a block for such transactions.

3.6. Interplanetary File System

Interplanetary File System (IPFS) is a distributed blockchain file system that stores and shares data in a distributed file system. It requires an IPFS server. It is possible to duplicate data stored on IPFS automatically or based on user behavior, e.g., data can only be replicated after a user has accessed them. Although storing data in an off-chain network lacks a blockchain's fundamental properties, it is more cost-effective, performs better, and is more versatile than storing data in smart contracts. The downside of using such an off-chain storage is that the smart contracts do not directly access the data, making automating processes on the blockchain more difficult. Figure 2 shows the illustration of the relationship between smart contracts and IPFS.

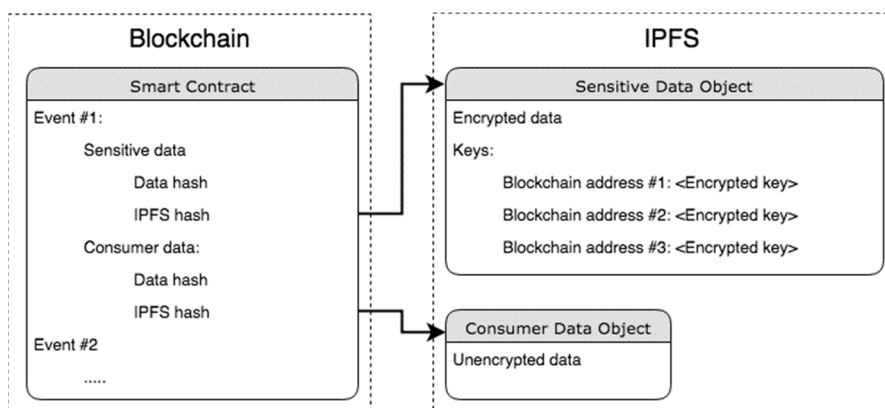


Figure 2. Illustration of relationship between smart contracts and Interplanetary File System (IPFS).

3.7. Security Analysis

Cryptography is an important computer security operation that converts data and information in an unintelligible form. There are two major categories to cryptography, i.e., symmetric and asymmetric encryption. Symmetric encryption is recognized as a single-key cryptography that is used for encryption and decryption. The data encryption standard (DES), advanced encryption standard (AES), and Blowfish are the three most popular symmetric key cryptography algorithms. Asymmetric encryption is known as cryptography with a public key, which uses two keys instead of one, a public key and a private key. The public key is used for encryption and the private key is used for decryption. Bitcoin and Ethereum use public-key cryptography to build wallets and sign transactions. This creates a unique signature when a contract is signed with a private key. When an owner signs a transaction, it is sent to the memory pool, where it remains until miners or validators process it.

4. System Design and Architecture

There are many sections of a full traceability framework. The uniqueness of this analysis is the use of a blockchain as the base layer. Therefore, choosing which blockchain technology to use and how to use it is the key to the entire system.

4.1. Permissionless vs. Permissioned Blockchain

The traceability framework discussed in this paper is for the purpose of verifying the origin and production process of a particular product. Since economic and legal information are processed, the process of consensus is vital. Therefore, blockchain technology plays an important role in maintaining the privacy of this information. In addition, the decentralized public blockchain Ethereum gives the output for every charged transaction, making it highly economical.

The openness of the consensus algorithm makes it highly vulnerable to attackers. On the other hand, the use of a licensed blockchain gives better preconditions with higher performance, because the fewer validators there are, the higher the performance will be. It does not allow general users to verify and trust the product, which is one of the key reasons for using the blockchain technology. If there are no external validators to track the activities on the blockchain over a certain span of time, then it will be very difficult to identify the manipulation done at the backend, maybe by a third-party user or the authorized party.

4.2. Selection of Consensus Algorithm

The next step is to decide the appropriate consensus algorithm and blockchain framework that can be used. In order to achieve a stable, working, and cheaper blockchain, a Byzantine Fault-Tolerant (BFT)-capable consensus algorithm is proposed, which supports validators efficiently to preserve the decentralized blockchain. The Istanbul Byzantine Fault-Tolerant (IBFT) algorithm is the best suited in this case. Figure 3 depicts the architecture of the IBFT algorithm.

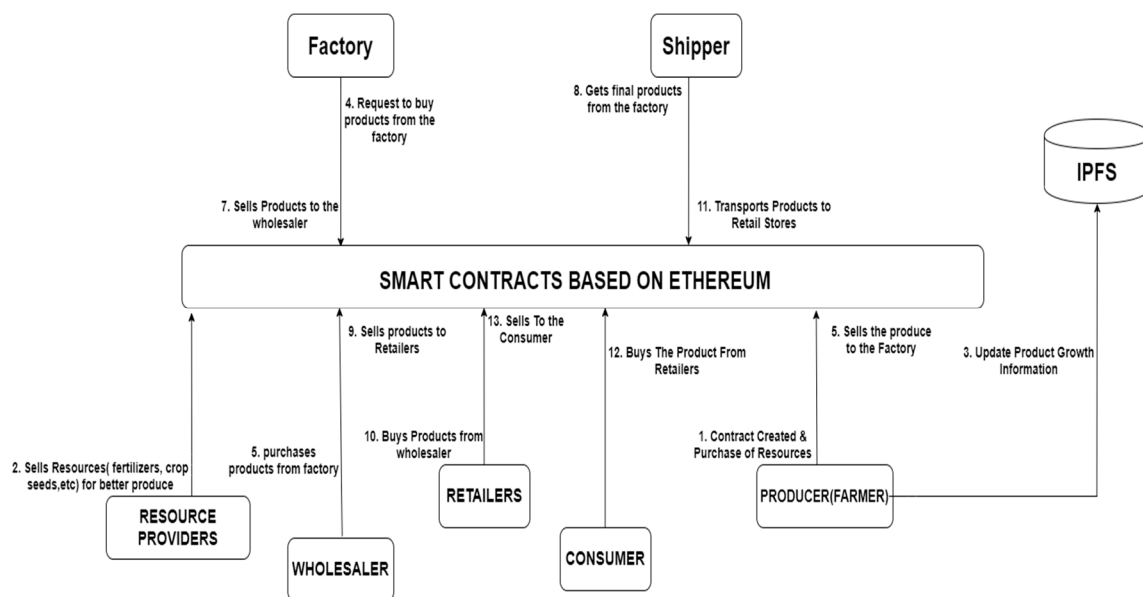


Figure 3. System architecture for product traceability and visibility using smart contracts based on Ethereum.

4.3. User Application

This is a software in which users, such as farmers or producers, retailers, shippers, and customers, can communicate. It is the client's duty to hash and encrypt event data and upload it to IPFS. It is also responsible for the developing and signing of transactions. Various key players participating in the supply chain are as follows:

- **Producer:** The supply chain starts from the firm, as the farmers are the key players in the agricultural supply chain. In plants, we monitor their growing parameters like temperature, soil moisture, and fertility. They are also responsible for processing the crop and for selling it to the food industry. The records that the producers must hold in the chain are location, height, product types, origins, irrigation, treatments, and pruning. The farmers are also responsible for daily tracking and documenting of the crop growth information, and storing them in the form of photographs or Moving Picture Experts Group (MPEG) files on the shared file system, for which Interplanetary File System (IPFS) is commonly used as a decentralized file system.
- **Supplier:** At each point in the product life cycle, suppliers have an important role. Organizations work closely with their suppliers to optimize their goods by acquiring raw materials and helping them to boost production while the market continues to saturate, and to find a better replacement for raw materials. On the other hand, the role of a retailer in a company can be difficult because of the quality level expected from the customers. The producers want suppliers to sell a lot of goods, due to which companies need to be flexible while deciding how to manage collaborations.
- **Factory:** Factories are an integral part of the production chain, and either produce or label goods. Food is processed at the host in different ways. Many of these products can be frozen, bottled, fried, roasted, processed, heated, pasteurized, or handed over. In wide factories, food production is primarily carried out by computers. Food processing personnel are active in food product manufacturing and storage. Knowledge of food safety is very important in every step of the food chain industry. Food processors need to be aware of food safety issues that are relevant to their role and to be prepared to act when problems arise.
- **Retailer:** The retailers have the dual roles of buying and selling goods. A retailer's duty is to find the most profitable route from the producer to buy the products and to carry on with the customer benefits. Often, they perform the functions of warehousing and delivery. They store the products in bulk and make them available according to the demand of the consumer.

Warehousing and recycling help to ensure that the products are available to consumers without delay. A retailer's primary function is to sell the products to customers, for whom the retailer adopts various techniques or business practices to meet the strategic goals.

- **Consumer:** Consumers decide which products and services they need, whether to buy for themselves, friends, family, or business clients. The media aid in persuasion, training, and awareness for customers on selecting and purchasing appropriate products and services. Customers are the main impetus of the economy, alongside corporations. Customers, once they get interested in a product, collect informative data about the product to decide to purchase the product. The information extracted from the data helps in deciding on its purchase. With the managed supply chain, investors harvest the benefit as they invest, and customers receive better products or services.
- **Other influencers:** In addition to the above-mentioned actors, the supply chain also includes several other members not specifically related to food processing or manufacturing. They are (i) raw material suppliers, responsible for providing all of the supplies needed for production, (ii) freight carriers, responsible for transporting goods to importers or other firms (distributors, wholesalers, retailers, etc.), and (iii) importers, responsible for buying goods from the producer, and selling and supplying the finished goods to the wholesaler or retailers of the destination country on the basis of the channels of delivery.

4.4. Controller

A service that lies between the user and Ethereum. The controller is used for scheduling and ordering transactions, load balancing, and managing errors.

4.5. Smart Contracts

There are two types of smart contracts: A smart contract representing information objects, and a smart contract representing ownership and information object contributors. The first smart contract is used for event registration and read-access control. The second is used to verify and/or pass ownership of goods, and to provide control overwrite access.

4.6. Validator

The validator runs the program for the blockchain node, Ethereum in this case. The validator communicates with the system's smart contracts to build a blockchain network. The validator is linked to many other validators. The validator validates incoming transactions and transfers them for processing to the blockchain network.

The validator is interested in the process of consensus within the blockchain network. The network will include thousands of client applications, as well as each user's smartphone or device that has the client application on it. The client will then submit transactions to a server. Each entity participating in the network will be able to provide its own controller, IPFS, and Ethereum node to improve decentralization. Figure 3 demonstrates how the components communicate among themselves.

A smart contract can represent an object of information (e.g., order, shipment, or product). When constructing one or more knowledge objects, the initial user (in this case, the producer) demands empty contracts from the smart contract pool that meet the user's requirements. Then, the smart contract transfers ownership to the person who requested it and returns to the farmer the contract addresses. When an event occurs and the user receives or generates it in the supply chain, the event data are hashed and encrypted with a special symmetric key. Then, the symmetric key is encrypted one or more times using each user's public keys with read access to the event data. The user program generates, and uploads to IPFS, two data objects, a confidential data object and a public user data object. The user asks the controller for a status update to decide which nonce to use before carrying out a new transaction. A nonce is a number specifying the order of the transactions. When one or more transactions are issued by the controller, then they are positioned in a scheduler. Later, the transaction is removed from the

scheduler and sent to Ethereum, depending on the transaction priority and current blockchain load. If the transaction is included in a block, it is deemed completed, and the reply is returned to the user.

5. Implementation of Food Traceability and Visibility System

In this section, we discuss the implementation of the proposed system. The key feature of using the traceability function in the proposed blockchain-based supply chain using smart contracts is to present the non-tamperable information to all the members of the supply chain. All information regarding the agreements between the members of the supply chain is stored in a database, starting from the agreement of the resources sold by the resource company to the farmers. The farmers constantly track the condition of the farming land and the growth of crops and upload the pictures through IPFS, which offers an e-database that can be utilized for verification of the agreements between the members. These digital transactions are good for tracking, but they also increase chances of swindling by some members. Therefore, the blockchain records the information as it is with verified identification of the real members. Scams can also be identified and mitigated by blockchain in this case. The system can give extra features to nullify the whole supply chain process and take desirable actions to impose penalties on scamsters. This will increase the authenticity of the information and make the chain more reliable. At the initial stage, the contract is created by the farmer; after that, the farmer is ready to buy resources in offline mode from the authorized resource companies. Here, the authenticity of the requesting producer is confirmed by the smart contracts.

If the payment for the resources is done as in Algorithm 1, all members in the supply chain are informed about all of the changes happening in the supply chain. The process terminates if it fails to inform the members of the changes. Product buying from the factory by the wholesaler is described in Algorithm 2. Product storage and processing are done in factories. Products are stored in the warehouse in a large quantity, and can only be sold if the buying request is made by the wholesaler. The products generally include the date of buying, date of shipping, and other related information about the product. Here, the *Product Requested* condition is created, resulting in the generation of a contract state, i.e., *Purchase from Factory*. If the product deal concurred and the product price is paid, then the agreement is successful, thereby alerting all of the members in the supply chain about the successful purchase of the product. If the status is that the product price is not paid, then the system should send an alert to all members in the supply chain about the failure of the agreement by changing the contract state from the *Product Requested* condition to the *Product Request Aborted* condition. Next in the chain, the product is shipped by the shipper to the retailers. Many important factors, such as the manufacturing date of the product, purchase date, and the total quantity of the product sold, are verified. Each of the shippers and the retailers are identified through their Ethereum Inscription (EI), as shown in Algorithm 3. Only the authorized retailers and shippers are allowed in the contract; if the product agreement and the payment of the product are done, then the chain moves forward to the shipment of the product by the shipper. Upon successful shipment of the product to the retailers, the contract notifies the retailer of the success with a message; otherwise, it notifies about the failure of the shipping of the product, and the message of failure is sent to all of the members in the supply chain.

Lastly, the customer purchases from the retailers, which is explained in Algorithm 4 and described below.

The customers are the final members in the supply chain and the smart contract only allows the customers to purchase products from the authorized retailers. The major factors to be considered here are the universal ID (UID) of the product to be purchased and the date of purchase. Initially, the customer is *Ready to Purchase*, and on successful payment for the purchase of the product, the nature of the contract is changed to *Product Dispatch Successful* from the retailer's end and *Bought Successfully* from the customer's end. If the payment is not successful, then the contract changes to *Product sale Aborted*, resulting in the failure of purchase, and the contract changes to the *Abort purchase* state, thus sending an alert message of the failure of the agreement to all of the members in the supply chain.

Algorithm 1. Resource company sells resources (fertilizers, crop seeds, pesticides, etc.) to the farmer (producer).

Input: ‘X’ is the number of authorized producers

Ethereuminscription(EI) of producers

Ethereuminscription(EI) of Resourcecompany

Quantity, ResourceType, ResourcePrice.

1. ContractCondition is **Generated**
 2. Condition of a producer becomes **ResourceRequested**
 3. Resource company condition is **Prepared**
 4. Limit entry to only $x \in X$, i.e., authorized producers
 5. **If** *Producer = authorized & ResourcePrice = Paid*, **then**
 6. Contract condition modified to *ResourceRequestMade*
 7. Modify the condition of a producer to *HoldonforResource*.
 8. ResourceCompany condition is *ConsentToSell*
 9. Alerting message is generated specifying the Resource sale
 10. **End**
 11. **Else**
 12. Return to contract condition & display an error message
 13. **End**
-

Algorithm 2. Wholesaler buys product from factory.

Input: ‘Y’ is the number of registered wholesalers

Ethereuminscription(EI) of wholesalers

Ethereuminscription(EI) of Factory

Quantity, BoughtDate, ProductPrice

1. Contract condition is **PurchaseFromFactory**
 2. Condition of the wholesaler is **ProductRequested**
 3. Factory condition is **CropPurchasedFromProducer**
 4. Limit entry to only $Y \in Factory$
 5. **If** *ProductDeal has concurred & ProductCost = paid*
 6. **then**
 7. Contract condition modifies to *ProductRequestConcurred*
 8. Change the condition of the Wholesaler to *HoldonForProductFromFactory*
 9. The Factory condition becomes *SellProductToWholesaler*
 10. Generate an alert message showing the sale of the product to the requesting wholesaler
 11. **End**
 12. **Else**
 13. Contract condition changed to *ProductRequestAborted*
 14. Condition of the wholesaler is *RequestAbort*.
 15. Factory condition is *AbandonRequestOfWholesaler*
 16. Generate an alert message displaying the request abortion
 17. **End**
 18. **Else**
 19. Return to contract condition and display an error message
 20. **end**
-

Algorithm 3. Shipper transports product to retail stores.

Input: ‘rs’ is the number of authorized retail stores*Ethereuminscription(EI) of shipper,**Ethereuminscription(EI) of retail stores,**ManufacturingDate, QuantityObtained,**DateBought*

1. Contract condition is *ProductObtainedByShipper*
 2. Shipper condition is *ProductRecievedFromShipper*
 3. Retail stores’ condition is *ReadyToBuy*
 4. Limit entry to only $rs \in \text{retail stores}$
 5. **If** *sale = concurred & ProductPayment = success*, **then**
 6. Contract condition modifies to *ProductSoldToRetailStore*
 7. Retailer condition is *ProductDispatchedSuccessful*
 8. Generate an alert message of “successful completion”
 9. **End**
 10. **Else**
 11. Contract condition modifies to *SaleRequestRejected*
 12. Shipper condition updated to *RequestAborted*
 13. Retailer condition is *ProductDispatchAborted*
 14. Generate a request abortion alert message.
 15. **End**
 16. **Else**
 17. Return to contract condition and display an error message
 18. **end**
-

Algorithm 4. Consumer purchases from Retailers

Input: *Ethereuminscription(EI) of Retailer,**Ethereuminscription(EI) of Consumer,**PurchasedDate, Product UID,**Sales UID*

1. Contract condition is *SaleRequestConcurredSuccessful*
 2. Retailers’ condition is *ProductDispatchedSuccessful*
 3. Consumer’s condition is *ReadyToPurchase*
 4. Limit entry to Consumers only
 5. **If** *ProductTransaction = success*, **then**
 6. Contract condition updates to *ProductSoldToConsumer*.
 7. Retailer condition is *SaleofProductSuccessful*
 8. Consumer state is *BoughtSuccessfully*
 9. Generate a “Successfully completed a purchase” alert message
 10. **End**
 11. **Else**
 12. Contract condition updates to *ProductSaleRefused*
 13. Retailers condition is *ProductSaleAborted*
 14. Consumer condition is *AbortedPurchase*
 15. Alert with “Purchase Aborted” message
 16. **End**
 17. **Else**
 18. Return to contract condition and display an error message.
 19. **end**
-

6. Analysis of Effectiveness of the System

To know how performance indicators are affected by various parameters, we run several experiments. We used two efficiency matrices; the average throughput is measured in transactions per second (tps), and average total transaction processing time (ttpt) is measured in seconds. Here, throughput refers to how many transactions per second the machine can process. The system is designed to isolate the storage from the database; therefore, various processes can be performed in parallel on different machines. We measure parallel processes to determine the average transaction processing time. The total transaction processing time is calculated as (1).

$$ttpt = tct + tqt + tpt \quad (1)$$

where *tct* is the time for creating a transaction on the user side, *tqt* is the transaction queuing time, i.e., waiting for the controller, and *tpt* is the transaction proposal time, i.e., sending transactions to Ethereum, or processing and response time. The processing time on IPFS is less important than the processing time on Ethereum, because it is the recording of an occurrence in time on the blockchain, which is the key factor in providing its meaning to the blockchain-based traceability system. The following subsections deal with the analysis of the effectiveness of the proposed system on various parameters.

6.1. Development Environment

We created a testing environment that is like the real environment. We used Amazon Web Services (AWS) for the study. Because manual deployment is difficult and time consuming, an automated deployment system was used for streamlining the testing process. The system configuration and types of instances, i.e., hardware and number of nodes used for deployment, are discussed in Table 1.

Table 1. System configuration.

Component	Software	Version
Operating System	Windows 10	Home Single Language
Controller	Node.js	13.3.0
Blockchain	Ethereum	2.0.2
Distributed File System	IPFS	0.4.17

6.2. Hardware Analysis

The hardware validator and device controller affect the output of any program executed in the system. Our first experiment examined how efficiency changes with specific client hardware in a particular network.

- The average throughput increased to 290 when the instance was upgraded to t2.
- The use of RAM reached 75% and
- The load of the CPU increased to 60%.

Further upgrading of the instance specifications did not improve the performance. For the types of instances of c5, a small increase can be seen due to the higher clock speeds for those types of instances. RAM usage, percentagewise, decreased as the amount of RAM available increased with better instances, while the actual consumption remained the same. The additional vCPUs brought no additional output; however, the CPU load remained at 60% to the validator. The varying parameter is the hardware of the system, and the parameters were average transaction processing time and total transaction processing time combined.

The processing power (clock speed and/or number of vCPUs) and RAM were modified for each type of instance, as shown in Table 2. In all hardware measurements, a block time of one second was

used. Finally, in all of the tests conducted, each block's gas cap was set to a number that would not restrict the number of transactions that could be processed in order to find the highest possible load for the system. A setup of one client, one controller, and one validator were used in this experiment.

Table 2. Types of instances used of AWS.

Type of Instance	vCPU	Clock Speed (GHz)	RAM	Performance of Network (Mbps)
t2.nano	1	2.3–2.4	500 MB	Low
t2.small	1	2.3–2.4	2 GB	Low to medium
t2.medium	2	2.3–2.4	4 GB	Low to medium
c5.large	2	3.0–3.5	4 GB	Up to 10 Gb/PS
t2.large	2	2.3–2.4	8 GB	Low to medium
c5.xlarge	4	3.0–3.5	16 GB	Medium
c5.2xlarge	4	2.3–2.4	32 GB	Up to 10Gb/PS
t2.2xlarge	8	3.0–3.5	32 GB	Medium

The controller and validator hardware were set to the best form of instance possible, t2.2xlarge. The device hardware changed incrementally from t2.nano to t2.2xlarge. As shown in Figure 4, the least-performing instance type, t2.nano, yielded an efficiency of up to 158 tps. We observed the following.

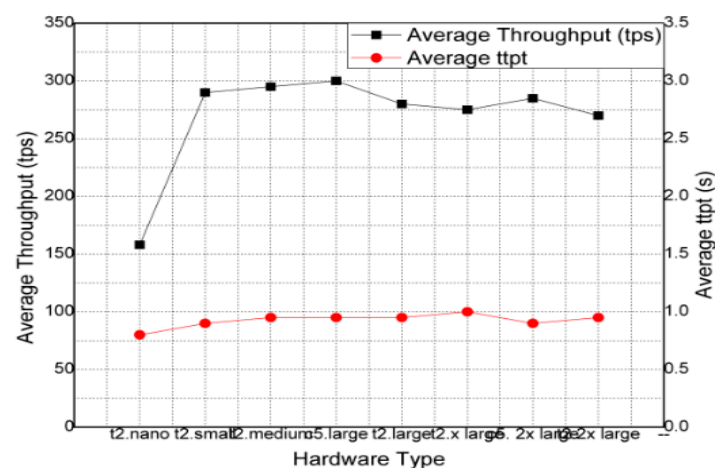


Figure 4. Effect of hardware instetting with one client (varying instance type), one controller (t2.2xlarge), and one validator (t2.2xlarge).

- RAM was used completely (100%) as soon as the test started.
- The CPU load during the test was steady at around 40%.
- The average throughput increased to 290 when the instance was upgraded to t2.small.
- The use of RAM reached 75% and the load of the CPU increased to 60%.
- The upgrading of the instance specifications did not improve the performance. For instance, c5, a small increase can be observed. However, this increment is due to the higher clock speeds. RAM use, percentagewise, decreased as the amount of RAM available increased with better instances, while the actual consumption remained the same.
- The additional vCPUs brought no additional output; however, the CPU load remained at 60%.

The above observations suggest that the controller or validator are the factors that influence the overall performance of the system. The reason is that the client did not affect the average ttp significantly. The processing speed of each individual transaction mostly depended on the CPU's clock speed and the relation between the network nodes. In this experiment, the speed of the clock remained

constant for all types of cases, and only the number of vCPUs varied. Figure 4 shows the effect of hardware insetting with one client and varying instance type.

6.3. Block Timings

The block time affects performance metrics adversely. An environment consisting of one client, one controller, and one validator is used in the test. They were executed on default settings, respectively: t2.small, t2.medium, and a t2.2xlarge case. Referring to Figure 4, when increasing the block time, output decreases. The increased time span in which a block is generated takes a longer time to process each transaction. The decrease in average throughput is not as easy to understand, however. The validators were supposed to be able to process more transactions for each block, despite an increased block time. Because one of the goals is to minimize the average ttpt and maximize the average throughput, the block time for the remaining tests is kept at one second. Figure 5 shows the effect of block timings in a setting with one client (t2.small), one controller (t2.medium), and one validator (t2.2xlarge).

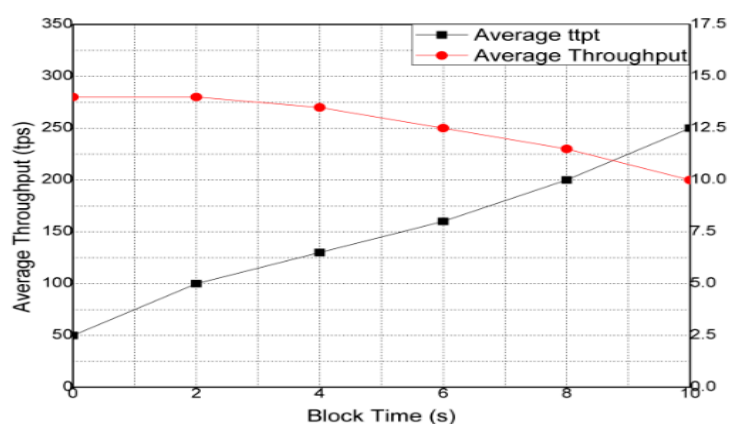


Figure 5. Effect of block timings in a setting with one client (t2.small), one controller (t2.medium), and one validator (t2.2xlarge).

6.4. Latency and Bandwidth of the Network

Here, the latency is evaluated for ten clients, five controllers, and ten validators with the usual types of instances: t2.small, t2.medium, and t2.2xlarge, respectively. In addition to the already existing latency in the network, the latency is applied in this experiment. The native latency, however, is very low (< 6 ms) so that it could be omitted. Therefore, it is worth stressing that it is the additional latency that is seen in Figure 6 and not the real latency.

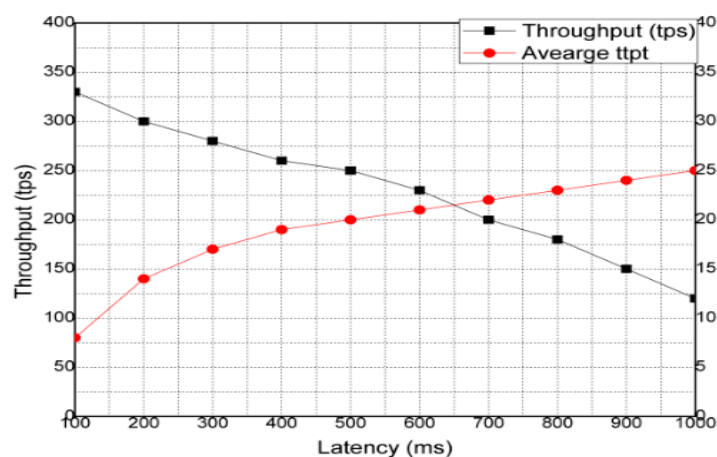


Figure 6. Effect of latency of the network in a setting of ten clients (t2.small), five controllers (t2.medium), and ten validators (t2.2xlarge).

The latency of the network has affected the average ttpt as anticipated. Due to these time constraints, the network bandwidths' impacts on the quality metrics were not checked separately. The rate at which data are transmitted and received is registered, which made general bandwidth requirements easily accessible. Both the rate of transmission and reception of the data increases with a higher load and more validators. The increase is, however, only seen on the nodes of the validator. Therefore, an increase in validator numbers does not impact the controller or the client. On the other hand, the validator has more transaction data to send and receive, with increased load and validator numbers. With more validators, the number of messages also increases. Figure 6 shows the effect of latency of the network in a setting of ten clients (t2.small), five controllers (t2.medium), and ten validators (t2.2xlarge).

6.5. Evaluation of IPFS

Here, we analyzed the efficiency of the upload times for 10 KB and 1 MB files, which were 7.0 and 15.0 ms, respectively. We see that the upload time for IPFS is smaller in comparison to the entire network, thus resolving the bottle-neck problem for the network. Here, the number of IPFS servers can also be scaled up to some extent. Figure 7 shows the analysis of the performance of IPFS when uploading files with increasing file size from 1 KB to 1 GB.

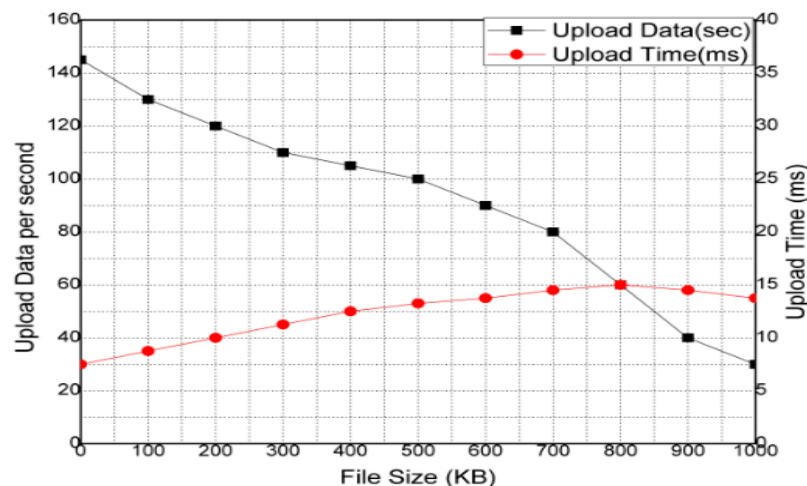


Figure 7. Performance of IPFS when uploading files with increasing file size.

7. System Simulation and Testing

Our analysis is based on behavior during runtime. We set the default instance types to thirty clients, ten controllers, and thirty validators: t2.small, t2.medium, and t2.2xlarge, respectively. The number of validators is a significant feature of our proposed design because it affected the decentralization of the system, which is highly important for the high number of validators, thus achieving the blockchain features. To gain from the blockchain functionality, the number of validators is desired to be as high as possible.

Thirty clients and ten controllers are used due to time and cost limitations. We used a block time of one second and implemented a latency of 250 ms. This analysis resulted in, on average, 161 tps throughput and an average of 4.82 second ttpt. The latency remained almost constant at the specified value of 200 ms, and the use of the network bandwidth for both upstream and downstream transmissions remained at around 26 Mbps. The load is sustained, on average, every minute between 20% and 60% per vCPU. Additionally, the load increases gradually over time. Though the behavior is interesting, the increased load decreases the stability, as shown in Figure 8. The use of memory or RAM, as shown in Figure 9, is continuously growing. This is anticipated because much of the increasing blockchain state is required to be stored in memory for easy access.

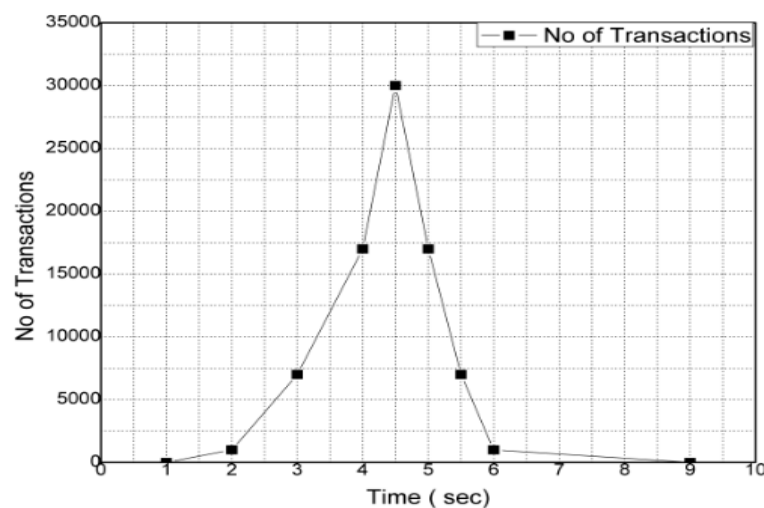


Figure 8. Variation of the total transaction processing time (ttpt) for all transactions sent during the simulation.

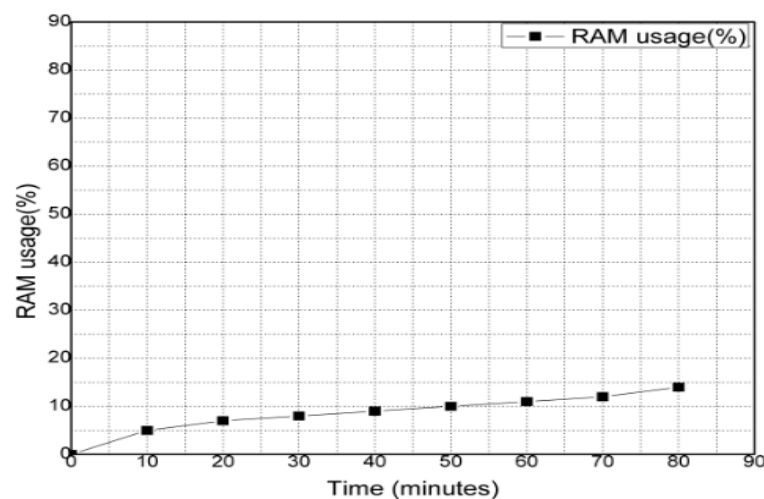


Figure 9. Usage of RAM during the simulation.

8. Results

Once the supply chain starts, farmers generate the genesis block and add the required information, as defined in Figure 3. Almost all miners inspect the block in the network before the next block is added to the chain. Every person in the system is identified with an identification number, whereas the batch number indicates the production batch provided by the producer. This is important, as there may be more than one organization delivering the same types of commodities to agricultural producers. The shipper, wholesaler, retail stores, and other individuals adopt this process to include their dealings into the chain.

A specific secret key is pre-distributed among all individuals in the system to ensure the secrecy. In general, the wholesalers, packers/distributors, product shippers, factories, and retailers follow this mechanism to incorporate their transactions into the supply chain. Every product can be tracked. Each product is allotted with a UID, and by entering the UID into the system, the customer can see the complete data, as well as their flow and associated information. When the UID is accessed, the seller account is first collected by the device. Each contract contains the previous block lot UID and hash value, which are used to trace back all related information and the data flow from the seller to the producer. Since the records of sold products are registered in the blockchain, the same thing cannot be sold again; Figure 10 shows the outcome of the system. The implemented system had an average

throughput of 161 tps and an average ttpt of 4.82 during a simulation of one hour, which had 30 clients, ten controllers, and 30 validators combined.

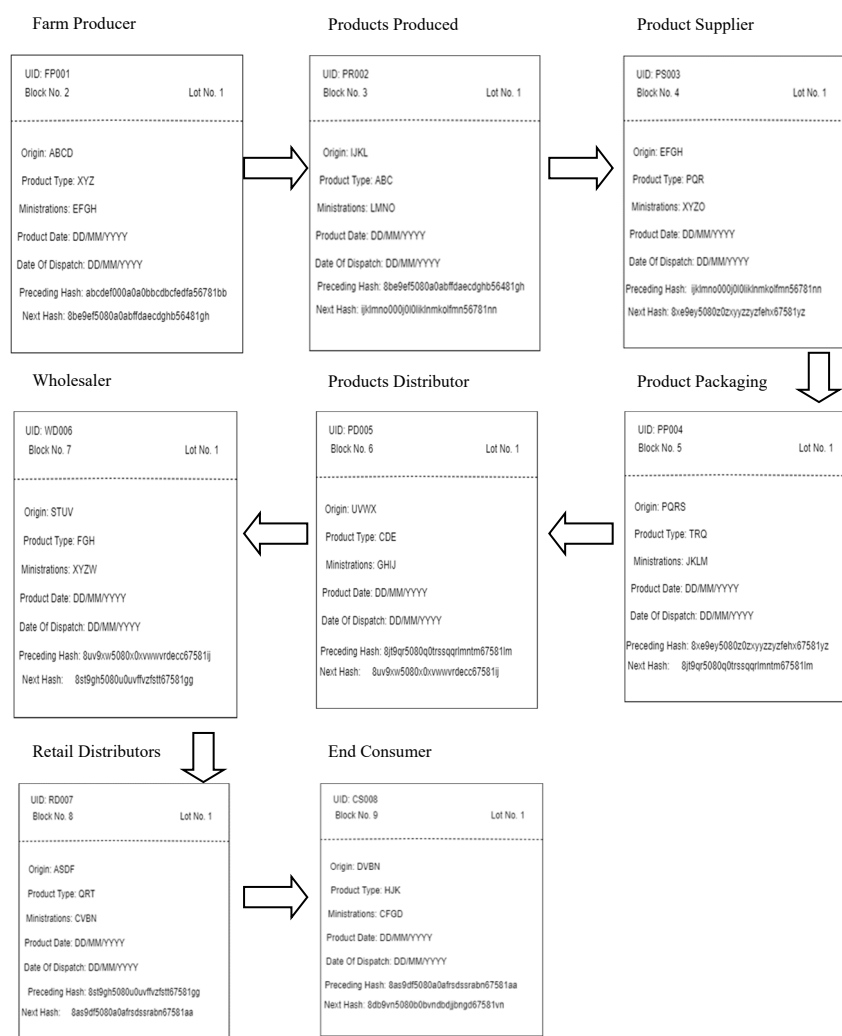


Figure 10. Illustration of the blocks generated by farm producer, products produced, product supplier, product packer, wholesale distributor, product distributor, retail distributors, and end consumer.

9. Conclusions

Blockchain has become one of the key aspects for most technologies, ranging from healthcare to farming. This paper proposes a blockchain-based traceability network that provides food supply chain identification, meaning, safety, and security. While food adulteration has risen on the global market, blockchain is an innovative solution to this problem. The proposed traceability system introduces a private blockchain with five major participants, thus allowing the participants to encrypt secret information. This methodology covers the entire data collection and information management process of each item in the supply chain for agriculture, which offers monitoring, visualization, and accountability for agri-food quality and safety, and can be extended to all food sectors. The potential benefit of enhanced traceability, performance, and public health safety encourages more research and development towards the advancement of an innovative business approach centered on blockchain and IoT. In this paper, we tried to design a more effective system to trace an agricultural product through the supply chain, with average throughput of 161 tps and an average ttpt of 4.82, over a simulation time of one hour per 30 clients, ten controllers, and 30 validators. Our future work will be

based on performing different experiments on a greater number of clients with increased controllers and validators through integrated cloud services.

Author Contributions: Conceptualization, methodology, investigation, writing—original draft preparation, N.J.; methodology, visualization, investigation, D.P.; writing—review and editing, supervision, S.J.; validation, resources, funding acquisition, Y.L.; writing—review and editing, supervision, funding acquisition, G.P.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2016R1D1A1B02008553).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ministry of Health and Family Welfare Report No. 37 of 2017; Implementation of Food Safety and Standards Act 2006; Union Government, Govt. of India: Delhi, India, 2017; pp. 1–102.
2. MLJ. FSSAI. Available online: <https://fssai.gov.in/cms/food-safety-and-standards-act-2006.php> (accessed on 10 April 2020).
3. PTI. Swiggy, Zomato, Others Remove 10,500 Restaurants for Violating Food Safety Law. Press Trust of India, 2018. Available online: <https://economictimes.indiatimes.com/small-biz/startups/newsbuzz/swiggy-zomato-others-remove-10500-restaurants-for-violating-food-safety-law/articleshow/67093466.cms?from=mdr> (accessed on 10 April 2020).
4. Sardar, D.R. *Maggi Stews in Lead and MSG Pot: Controversy Over India's Favourite Instant Noodles*; Social Science Research Network: Rochester, NY, USA, 2018.
5. Boukema, A.; Lahaye, R. *Advantages of Anomaly Detection Between the Controlling Unit and the Process Devices of an Industrial Control System*; University of Amsterdam: Amsterdam, The Netherlands, 2017; pp. 1–14.
6. Paramithiotis, S.; Drosinos, E.H.; Skandamis, P.N. Food recalls and warnings due to the presence of foodborne pathogens—A focus on fresh fruits, vegetables, dairy and eggs. *Curr. Opin. Food Sci.* **2017**, *18*, 71–75. [CrossRef]
7. Chinaka, M. Blockchain Technology—Applications in Improving Financial Inclusion in Developing Economies: Case Study for Small Scale Agriculture in Africa. Ph.D. Thesis, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, 2016; p. 43.
8. Afshin, A.; Sur, P.J.; Fay, K.A.; Cornaby, L.; Ferrara, G.; Salama, J.S.; Mullany, E.C.; Abate, K.H.; Abbafati, C.; Abebe, Z.; et al. Health effects of dietary risks in 195 countries, 1990–2017: A systematic analysis for the Global Burden of Disease Study 2017. *Lancet* **2019**, *393*, 1958–1972. [CrossRef]
9. Dai, H.-N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A Survey. *IEEE Int. Things J.* **2019**, *6*, 8076–8094. [CrossRef]
10. Ustundaga, A.; Tanyasb, M. The impacts of Radio Frequency Identification (RFID) technology on supply chain costs. *Transp. Res. Part E Logist. Transp. Rev.* **2009**, *45*, 29–38. [CrossRef]
11. Tian, F. A supply chain traceability system for food safety based on HACCP, blockchain & Internet of things. In Proceedings of the 14th International Conference on Services Systems and Services Management, ICSSSM 2017, Dalian, China, 16–18 June 2017; pp. 1–6.
12. Tripoli, M.; Schmidhuber, J. *Emerging Opportunities for the Application of Blockchain in the Agri-food Industry*; FAO: Geneva, Switzerland, 2018; pp. 1–29.
13. Kamath, R. Food Traceability on Blockchain: Walmart's Pork and Mango Pilots with IBM. *J. Br. Blockchain Assoc.* **2018**, *1*, 47–53. [CrossRef]
14. Aung, M.M.; Chang, Y.S. Traceability in a food supply chain: Safety and quality perspectives. *Food Control* **2014**, *39*, 172–184. [CrossRef]
15. Mao, D.; Hao, Z.; Wang, F.; Li, H. Innovative blockchain-based approach for sustainable and credible environment in food trade: A case study in Shandong Province, China. *Sustainability (Switzerland)* **2018**, *10*, 3149. [CrossRef]
16. Arena, A.; Bianchini, A.; Perazzo, P.; Vallati, C.; Dini, G. BRUSCHETTA: An IoT Blockchain-Based Framework for Certifying Extra Virgin Olive Oil Supply Chain. In Proceedings of the 2019 IEEE International Conference on Smart Computing (SMARTCOMP), Washington, DC, USA, 12–15 June 2019; pp. 173–179.

17. Aich, S.; Chakraborty, S.; Sain, M.; Lee, H.; Kim, H.-C. A review on benefits of IoT integrated Blockchain based supply chain management implementations across different sectors with case study. In Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea, 17–20 February 2019; pp. 138–141.
18. Shakhbulatov, D.; Arora, A.; Dong, Z.; Rojas-Cessa, R. Blockchain Implementation for Analysis of Carbon Footprint across Food Supply Chain. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 546–551.
19. Lucena, P.; Binotto, A.P.; Momo, F.D.S.; Kim, H. A Case Study for Grain Quality Assurance Tracking based on a Blockchain Business Network. In Proceedings of the Symposium on Foundations and Applications of Blockchain (FAB '18), Los Angeles, CA, USA, 9 March 2018; pp. 31–36.
20. Kosba, A.; Miller, A.; Shi, E.; Wen, Z.; Papamanthou, C. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In Proceedings of the 2016 IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, 23–25 MAY 2016; pp. 839–858.
21. Yang, X.T.; Qian, J.P.; Sun, C.H. Design and application of safe production and quality traceability system for vegetable. *Trans. Chin. Soc. Agric. Eng.* **2008**, *24*, 162–166.
22. Sari, K. Exploring the impacts of radio frequency identification (RFID) technology on supply chain performance. *Eur. J. Oper. Res.* **2010**, *207*, 174–183. [[CrossRef](#)]
23. Foroglou, G.; Tsilidou, A.L. Further applications of the blockchain. In Proceedings of the The Columbia University PhD in Sustainable Development 10 Year Anniversary Conference, Athens, Greece, 27 February 2014; Volume 1, pp. 1–8.
24. Sayogo, D.S.; Zhang, J.; Pardo, T.A.; Tayi, G.K.; Hrdinova, J.; Andersen, D.F.; Luna-Reyes, L.F. Going Beyond Open Data: Challenges and Motivations for Smart Disclosure in Ethical Consumption. *J. Theor. Appl. Electron. Commer. Res.* **2014**, *9*, 1–16. [[CrossRef](#)]
25. Fischer, M.J.; Lynch, N.A.; Paterson, M.S. Impossibility of distributed consensus with one faulty process. *J. ACM* **1985**, *32*, 374–382. [[CrossRef](#)]
26. Dwork, C.; Lynch, N.; Stockmeyer, L. Consensus in the presence of partial synchrony. *J. ACM* **1988**, *35*, 288–323. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).