Traceability in Food Supply Using Blockchain

220611613

MSc in Advanced Computer Science,
School of Computing Science, University of Newcastle, U. K.

Prabhu Yogesh Vijayan

Abstract. Blockchain as a technology has gained much traction in recent years due to its decentralised and distributed ledger functionality, allowing transactions to be recorded and verified without relying on a central authority or trusted intermediary. One of the areas where blockchain has shown promise is the food supply chain. Since the reality of the food supply chain is extensive and involves many stakeholders from farm to fork, including farmers, distributors, and consumers, each of them has varying records and data storage that makes the whole process fragmented. Hence this project aims to develop a shared digital ledger which is established through blockchain technology to securely and immutably record and store transactions. Each transaction, including activities like goods movement, quality testing, or certifications, is represented as a block that is connected to the preceding block, creating a chain of information. This decentralised architecture guarantees that all participants possess identical information, eliminating the necessity for intermediaries and minimising the chances of fraudulent activities or tampering with data.

Declaration: I declare that this dissertation represents my own work except where otherwise explicitly stated.

1 Introduction

Progress in supply chain technology has opened access to a wide array of food items from around the world. However, the primary hurdle that the food and beverage sector confront is the preservation of product quality and prevention of contamination during the movement of food within the supply chain. This challenge also involves reducing occurrences of food fraud or tampering, all while aiming to maintain cost-effectiveness in production and operations.

The food industry faces a significant hurdle in the shape of food adulteration or fraud, which holds far-reaching consequences. Notably, situations have arisen where premium food products are swapped with more affordable components while maintaining identical labelling. An impactful incident occurred in Europe in 2013 when products labelled as beef were revealed to contain 80-90% horse meat [1]. Moreover, the responsible food supplier neglected to promptly notify local authorities about this matter.

Another significant scandal was the 2008 baby milk powder scandal in China was a catastrophic event that severely impacted the country's food industry and raised significant concerns about food safety and regulatory oversight [2]. The incident revolved around the deliberate adulteration of infant formula with melamine, a toxic compound used in plastics and fertilizers. This adulteration was carried out to artificially boost the protein content of the milk, leading to false readings in protein content tests. The scandal exposed a range of systemic issues within China's food industry. It highlighted lapses in quality control, inadequate regulatory oversight, and a lack of transparency in the supply chain.

Another significant challenge is maintaining specific temperature levels during the transportation of food products. An extensively documented study sheds light on a noteworthy incident that occurred in May 2013. During this period, there was an outbreak of hepatitis A virus infection in the United States and several European countries [3]. The investigation revealed that the source of this outbreak was linked to pomegranates and frozen berries that had been transported from Turkey and certain Gulf nations, including Morocco.

Due to such incidents, consumers are worried about the authenticity of the food products they purchase. This hesitation is partially due to the lack of an alternative that assures them of receiving a product as advertised. Additionally, the presence of information asymmetry within the fragmented supply chain makes the process of tracking problematic for buyers. Even though manufacturers and companies claim a farm-to-fork method there is no definite means at present for consumers to follow the product's origin, backtrack to the specifics of the food processing centre, validate the food distributor's hygiene standards, or access any information that would provide assurance of the product's authenticity. The absence of transparency within the supply chain network obstructs consumers from verifying the legitimacy of the received product.

1.1 Research Motivation

At its core, the impetus for integrating blockchain into supply chain management arises from the imperative to confront immediate challenges. These challenges span from ensuring the authenticity and traceability of products to combating fraudulent activities, heightening transparency, and optimizing operational efficiency. The concrete cases presented earlier vividly illustrate the potential of blockchain to reshape the dynamics of supply chains. By addressing these challenges, blockchain has the capability not only to transform the fabric of supply chains but also to safeguard consumer welfare, foster collaborative endeavours, and inaugurate a phase marked by transparency and accountability. As various industries mobilize to surmount these hurdles, the adoption of blockchain technology emerges as a transformative avenue for cultivating a supply chain ecosystem that is both more secure and streamlined, placing consumers at its core.

Blockchain technology is significantly transforming the complex field of food supply chain management. Its diverse and impactful applications address concerns about transparency, traceability, safety, and authenticity in the production, distribution, and consumption of food items. At present, blockchain is being harnessed in various ways throughout the food supply chain.

One prominent application involves establishing traceability and verifying origins. The decentralized and unchangeable nature of blockchain allows the recording of every stage of a food product's journey as immutable transactions [4]. This transparency empowers consumers to independently verify the authenticity and source of products, gaining insights into their production processes and ethical standards from farm to store.

Another crucial role is its capacity to combat food fraud and counterfeiting. By creating an enduring record of a product's history, blockchain ensures that data linked to food products remains unaltered [4]. This robust feature serves as a deterrent against fraudulent activities, bolstering consumer confidence in the accuracy of purchased goods.

In the realm of food safety, blockchain plays a vital role in facilitating swift and precise recalls in instances of foodborne illnesses or contamination. By identifying the origin of the issue within the supply chain, blockchain reduces health risks and financial losses for companies. This real-time tracking enhances the efficiency of the recall process.

Furthermore, blockchain streamlines supply chain management by providing a transparent shared ledger. Administrative processes are streamlined, minimizing paperwork, conflicts, and operational costs. Management of the cold chain, which is essential for preserving perishable goods, benefits from blockchain's ability to monitor temperature and conditions during transportation and storage.

Blockchain fosters direct interaction between producers and consumers through mechanisms like QR codes and RFID tags [5]. Scanning these codes offers consumers comprehensive details about a product's history, ingredients, and ethical practices. Such transparency cultivates consumer trust and loyalty, especially among those advocating for sustainably and ethically produced goods.

1.2 Aim and Objectives

1.2.1 Aim

The aim of this project is to analyse the security vulnerabilities faced by food supply chains and develop an effective blockchain-based food supply traceability system.

1.2.2 Objectives

3.1 To analyse the existing systems and identify the security challenges and vulnerabilities faced by food supply chains and understand the limitations of existing approaches in addressing these issues.

3.2 To explore the fundamental concepts and features of blockchain technology and assess its suitability for enhancing security in food supply chains.

3.3 To analyse the potential applications of blockchain in improving traceability, quality assurance, supply chain optimisation, and counterfeit prevention within the food industry.

3.4 To develop a comprehensive blockchain security framework specifically tailored for food supply chains, incorporating design principles, consensus mechanisms, identity management, and data encryption techniques.

2 Related Work

In October 2016, a ground-breaking collaborative initiative was launched, bringing together retail giant Walmart, technology leader IBM, and Beijing Tsinghua University. This ambitious project was driven by the shared goal of transforming the way food traceability is managed by leveraging the capabilities of blockchain technology [6].

At its core, the project aimed to create an innovative and advanced model for tracking the journey of food products throughout the supply chain. Traditional methods of tracing food items often involve complex, fragmented systems that can result in inefficiencies, delays, and potential gaps in transparency. The introduction of blockchain technology sought to address these challenges by establishing a decentralized, secure, and transparent system for tracking food products from their source to the consumer's table.

This research holds significance for shedding light on several factors that need to be considered when strategizing the implementation of blockchain solutions within a complex, multi-tier, and globally interconnected food supply chain network. In the business context, transparency emerges as a critical element that profoundly impacts operational efficiency, fostering enhanced collaboration among team members and facilitating informed and harmonious decision-making processes.

An illustrative example by Tian [5] explores an agro-inventory system built upon RFID technology and blockchain, with a specific focus on China. Tian examines the development of a model that combines the functionalities of RFID and blockchain technologies to enhance traceability within the agri-food supply chain and ensure the accuracy of records related to food, including information about warehouses and distributors. This study is notable for its innovative utilization of both RFID and blockchain technology. However, it's important to acknowledge that active RFID tags, although effective, come with certain drawbacks due to their higher cost and complexity, stemming from components like antennas radio batteries and transceivers.

In response to the traceability challenges inherent in the supply chain sector a work by Zhang et al [7] underscore the limitations of traditional methods due to their time-intensive nature. To counteract this, they introduce a decentralized framework that merges consensus mechanisms, IoT technology, and fuzzy logic techniques for the management of food product shelf life. Particularly significant is their utilization of fuzzy logic for decision-making and the precise determination of product deterioration dates, enhancing the system's dependability.

Concurrently addressing the intricacies of IoT technology with the work of Zhang et al who documented a traceability system that merged blockchain and NFC technologies for agri-food tracking. This amalgamation was anticipated to deliver heightened levels of transparency and security within the system. Collectively, these initiatives underscore the ongoing exploration of innovative technologies such as blockchain, IoT, and fuzzy logic, aimed at reshaping supply chain traceability, transparency, and overall operational efficiency.

Another such work was tracing wine production using blockchain [8]. The implemented system guarantees that transactions can be accessed by participants across multiple phases of the wine production process. This inclusivity encompasses stages like grape cultivation, wine processing, logistics, and eventual consumption. As a result, the system establishes a robust, open, and accurate structure for exchanging information. This transparency ensures that all stakeholders involved have a clear view of the transactions, enhancing security and trust throughout the entire wine supply chain.

3 Background

3.1 Blockchain

Blockchain is a technology that uses a decentralized, distributed system to securely and openly record transactions digitally [9]. It is the primary technology that roots the cryptocurrency of Bitcoin which it was developed for specifically [10]. Although it's most well-known for its connection to Bitcoin, its potential uses transcend beyond virtual money.

Blockchain is fundamentally a decentralized system. A blockchain network, in contrast to traditional databases maintained by a single person, is made up of numerous computers called nodes. These nodes work together to keep the network running smoothly and verify transactions. Since every node has a complete copy of the entire blockchain, there is no need for a central authority and backup copies are always available.

The primary unit of a blockchain is the "block." Transactions, which encompass a wide variety of data beyond financial exchanges, are grouped into these blocks. As transactions occur, they are verified by participants in the network and organized into a new block. Once a block is filled with transactions, it becomes a part of the existing blockchain.

The concept of cryptographic hashing is pivotal in guaranteeing data integrity and protection against tampering within the blockchain. Every block contains a cryptographic hash—a unique and complex identifier—derived from the data of the previous block. This hash serves as a digital signature, representing the content of the block. Importantly, any alteration to the data within a block would result in a change in its hash, requiring changes in all subsequent blocks. This mechanism establishes a secure connection between blocks, making it extremely challenging to manipulate historical data.

Consensus mechanisms play a crucial role in the blockchain's operation. Before a new block is added to the blockchain, the network must reach an agreement that the transactions within it are valid. Different consensus mechanisms, like Proof of Work (PoW) and Proof of Stake (PoS), facilitate this agreement. In PoW-based blockchains such as Bitcoin, miners compete to solve complex mathematical puzzles, with the first successful miner adding the new block.

In PoS, validators are chosen to create new blocks based on the amount of cryptocurrency they hold and are willing to "stake" as collateral.

Once a block is added to the blockchain, the data it contains becomes highly resistant to modification due to the interconnectedness of blocks and the consensus achieved across the network. This immutability is a crucial feature that enhances the security and reliability of blockchain systems.

Transparency is another key aspect of blockchain. All transactions and blocks are visible to all participants in the network. This transparency enables independent verification of information, fostering trust and accountability.

In certain cases, disagreements within the network can lead to forks, where the blockchain splits into two paths. This can arise from differences in consensus rules or software upgrades. A hard fork results in the creation of a new separate blockchain, while a soft fork maintains compatibility with the existing one.

3.2 Decentralization

Decentralization, within the context of blockchain technology, refers to the core concept where control and authority are dispersed across a network of participants, eliminating the need for a single central entity to oversee transactions and data [10]. This concept marks a departure from conventional systems that often rely on centralized control, yielding notable implications and advantages.

One key benefit of decentralization involves eliminating the vulnerability of a sole point of failure. In centralized systems, the failure of the central authority or server can disrupt the entire system. Conversely, in a decentralized blockchain network, authority is distributed across multiple nodes. Thus, even if one node encounters a failure or compromise, the rest of the network remains unaffected, contributing to a more robust system.

Security is also elevated within decentralized blockchains. Traditional centralized systems are susceptible to hacking attempts and data breaches, given that a successful attack on the central server can expose a substantial amount of sensitive information. In contrast, a decentralized network fragments data across numerous nodes, significantly heightening the complexity for malicious actors to undermine the entire system.

Decentralization also carries implications for resisting censorship. In centralized systems, a central entity can control and manipulate the flow of data. However, within a decentralized blockchain, participants can directly engage and carry out transactions without intermediaries, thereby curtailing the potential for censorship since no single entity can exert control or halt transactions.

Transparency emerges as another consequential outcome of decentralization. Transactions are logged on an open ledger accessible to all network participants. This transparency fosters trust and accountability, as anyone can independently verify transactions and data.
An integral concept linked with decentralization is the idea of "trustless transactions." Traditional transactions often necessitate trust in intermediaries like banks. In a decentralized blockchain, participants aren't obligated to trust one another directly. Instead, they depend on the inherent security provided by the blockchain's consensus mechanisms and cryptographic methods, enabling secure transactions without the necessity of intermediaries.

Moreover, decentralization promotes community governance. Several decentralized blockchains operate on a consensus-driven basis, where decisions are collectively made by network participants. This approach nurtures community-led development and decision-making, reducing the influence of a single controlling entity and facilitating a more inclusive and democratic approach.

However, decentralization poses challenges. The distributed nature of blockchain can lead to slower transaction speeds and greater resource demands due to the requisite consensus mechanisms and data replication. Additionally, resolving conflicts or implementing changes in a decentralized environment can be intricate due to the absence of a central authority.

3.3 Distributed Ledger

A distributed ledger, integral to the concept of blockchain, functions as a digital system for recording data across a network of connected computers or nodes [10]. This differs from conventional centralized ledgers controlled by a single entity. In a distributed ledger setup, each participating node possesses an identical version of the complete ledger. Changes to the ledger are collectively agreed upon through consensus mechanisms, ensuring transparency, security, and the absence of centralized authority.

The core attribute of the distributed ledger lies in its decentralized nature. Each network participant maintains an identical copy of the ledger, updating it with new transactions and information. These transactions are grouped into blocks, creating a sequential chain. This series of interconnected blocks forms the blockchain. Significantly, the data contained in each block includes a reference to the previous block, creating a secure and tamper-resistant connection between them.

The integrity of the distributed ledger is maintained through consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS). Before a new block is added to the chain, network participants must collectively agree on the validity of the transactions. In PoW, participants, often referred to as miners, solve complex mathematical puzzles. Conversely, in PoS, validators are selected based on their stake in the network. Once consensus is reached, the new block is appended to each participant's version of the ledger.

The advantages of a distributed ledger are multi-fold. Most notably, its decentralized nature eliminates the requirement for a central entity to oversee transactions, reducing the risk of single points of failure or manipulation. This significantly enhances the security and resilience of the system. Moreover, transparency is a pivotal aspect as all participants have access to the same data. This fosters a trustless environment where verification is achievable without relying on intermediaries.

Beyond cryptocurrencies, distributed ledgers find utility in diverse domains. They are employed in supply chain management to transparently track the movement of goods and ensure precise records. They also power smart contracts, which are self-executing agreements triggered by predefined conditions. Furthermore, the ability to maintain tamper-resistant records is invaluable in sectors like healthcare, real estate, and voting systems, where data accuracy and transparency are of utmost importance.

3.4 Smart Contract

A smart contract is a ground-breaking concept made possible by blockchain technology. It acts as an automated and self-executing digital agreement, eliminating the need for intermediaries. Essentially, a smart contract is a coded program designed to enforce the terms and conditions of an agreement between parties [11].

Fundamentally, a smart contract is a piece of code written using specialized programming languages tailored for blockchain platforms. For instance, Solidity is a well-known language used for crafting smart contracts on platforms like Ethereum. This code outlines specific rules, conditions, and actions that will unfold when particular criteria are met. This coding process intricately maps out the sequence of events that will unfold throughout the contract's lifespan.

After development, the smart contract is deployed onto the blockchain. Each unique contract receives a distinct address within the blockchain network. This address serves as the point of interaction and execution for the contract. Crucially, the contract's address is open to all participants in the network, ensuring transparency and accountability.

When parties aim to engage in a transaction governed by a smart contract, they initiate interactions with the contract's address by sending transactions to it. These parties, often referred to as "signers" or "users," could be individuals, organizations, or entities. Transactions can encompass a range of actions, such as transferring digital assets, cryptocurrency, or even triggering specific functions within the contract.

The distinguishing feature of smart contracts lies in their autonomous execution. Once conditions set in the contract's code are met, the contract independently carries out the predetermined actions. This self-automated process eliminates the need for manual intervention or dependence on external entities to enforce the contract's terms. Consequently, transaction outcomes are determined by the code itself, enhancing accuracy and consistency. Additionally, data processed within a smart contract is transparent and verifiable. As the contract operates on a blockchain, interactions and actions are permanently recorded in the blockchain's distributed ledger. This ledger serves as an auditable record, visible to all participants, ensuring that the execution process is traceable and accountable.

The principle of trustlessness is a central tenet of smart contracts. Participants engaging in transactions via smart contracts need not trust each other. Instead, they place their trust in the code's integrity and the consensus mechanisms of the blockchain network. This aspect substantially reduces the risks linked with fraud and manipulation.

Another important quality of smart contracts is their irreversible nature. Once conditions specified in the contract are met, the execution is final and cannot be altered. This attribute provides participants with certainty and conclusiveness, as transactions cannot be easily reversed or modified afterward.

It's important to acknowledge that executing smart contracts may involve costs, commonly referred to as "gas fees," particularly in platforms like Ethereum. These fees cover the computational resources required to process and execute the contract. Users initiating contract actions are responsible for paying these fees, which can vary based on the contract's complexity and network conditions.

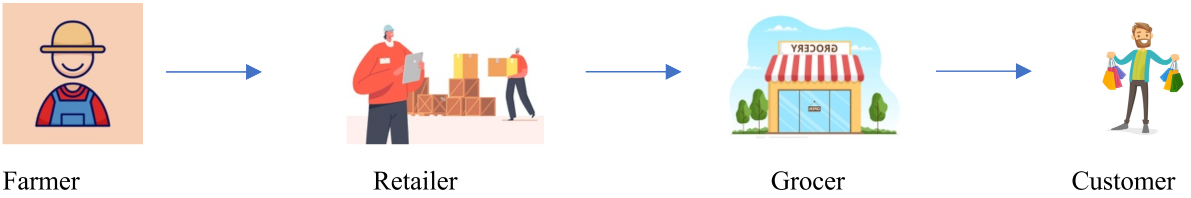
4. Design and Implementation

4.1 Structure

The process initiates as the farmer generates a product and makes it available for purchase by the retailer. Once the product is completed, the farmer organizes its transportation to the retailer 's location. Upon receiving the product, the retailer takes on a range of responsibilities. These tasks encompass processing the product, which can involve activities like sorting, cleansing, and evaluating quality. Subsequently, the retailer meticulously packages the product, ensuring it's ready for retail presentation, and subsequently makes it accessible for sale.

At this point, the grocer enters the equation. The grocer obtains and procures the product from the retailer with the intent of presenting it to their customer base. The retailer then manages the product's transportation to the grocer's premises, ensuring it reaches the grocer securely and in the anticipated condition. Upon arrival, the grocer displays the product and offers it for purchase to customers.

Ultimately, the customer becomes the final recipient of this supply chain journey. The customer, showing interest in the product, conducts a purchase from the grocer. This transaction marks the culmination of the cycle, with the customer acquiring the product that initially commenced its journey with the farmer. This series of events underscores the complex and interlinked nature of the supply chain process, involving diverse stakeholders, transactions, and logistical phases to facilitate the progression of a product from its inception to the customer's possession.



4.2 Technology and Reasons for Use

4.2.1 Solidity

Solidity is a specialized programming language exclusively designed for crafting smart contracts and decentralized applications (DApps) on blockchain platforms, with Ethereum being its primary use case. Its core purpose is to empower developers in defining the rules, logic, and interactions of smart contracts, which operate as self-executing agreements in the form of code. These contracts automatically execute predefined actions when specific conditions are met, eliminating the need for intermediaries, and boosting transparency and credibility in transactions.

A central function of Solidity revolves around enabling the creation of smart contracts. These contracts serve as the fundamental building blocks of blockchain applications, enabling secure and automated execution of processes without relying on a central authority. By encoding contractual terms within these smart contracts, parties involved can ensure that terms are fulfilled automatically based on predetermined criteria.

Expanding beyond smart contracts, Solidity is also instrumental in the development of DApps. These applications operate on a decentralized network of computers, often referred to as a blockchain, rather than relying on a centralized server. The use of DApps aims to provide solutions characterized by security, transparency, and resilience to censorship across various domains, from financial applications to supply chain management.

A noteworthy aspect of Solidity is its accessibility to developers familiar with other programming languages. It shares resemblances with languages like JavaScript and Python, making it approachable for programmers and enabling them to work efficiently within the Solidity environment.

Solidity employs static typing, requiring explicit declaration of variable types. This feature enhances code clarity and helps prevent certain programming errors. Additionally, Solidity places a strong emphasis on security, incorporating mechanisms to strengthen the resilience of smart contracts against vulnerabilities and malicious attacks. This includes features such as input validation and safeguards against re-entrancy attacks.

Although Solidity is primarily associated with Ethereum, it also finds application in other Ethereum-compatible blockchains like Binance Smart Chain and Polygon. By using Solidity, developers can define smart contract behaviour, specify functions, manage data storage, and outline interactions with external contracts and users. The Solidity code is then compiled into bytecode, a format that can be executed by the Ethereum Virtual Machine.

In essence, Solidity plays a crucial role in shaping the decentralized landscape by equipping developers to create self-executing contracts and DApps, establishing a foundation for enhanced trust, automation, and integrity within digital agreements and transactions.

4.2.2 Ganache and Truffle

The Ganache Truffle Suite is a comprehensive package of tools designed to simplify and enhance the development of Ethereum-based applications and smart contracts. This suite integrates two key components: Ganache and Truffle.

1. Ganache: Ganache serves as a private Ethereum blockchain emulator tailored for local development and testing purposes. It replicates the behavior of the Ethereum network, allowing developers to simulate blockchain interactions without needing to connect to the actual network. This proves invaluable during development and testing phases, enabling developers to experiment and iterate on their applications within a controlled environment.

2. Truffle: Truffle is a widely used development framework for Ethereum that streamlines various tasks related to building and deploying decentralized applications (DApps) and smart contracts. It encompasses features such as compiling, deploying, and testing smart contracts, along with facilitating the management of the entire development lifecycle of Ethereum projects. Truffle's suite of tools simplifies coding and deployment processes, assisting developers in crafting efficient, thoroughly tested, and secure blockchain applications.

In essence, the Ganache Truffle Suite merges Ganache's local Ethereum blockchain simulation capabilities with Truffle's development framework. By utilizing this suite, developers can enhance their efficiency, identify, and address issues early in the development cycle, and construct robust and dependable blockchain solutions.

4.3 Defining Roles for Users

The contract named "Roles.sol" has the purpose of overseeing roles on a platform, offering functionalities for role addition, removal, and role-related information retrieval. This contract is designed to be utilized in conjunction with other smart contracts that involve the management of roles.

To commence, the contract begins by introducing a library called "Roles." This library encapsulates a range of functions related to managing roles. Within this library, a structure called "Role" is defined. This structure incorporates a mapping that links addresses with Boolean values, designating them as role bearers.

Continuing, the library proceeds to implement internal functions aimed at handling various role-related actions. The "add" function, for instance, allows the inclusion of an account into a role. It ensures that the provided account address is not null and that the account isn't already associated with the role.

Similarly, the "remove" function is constructed to eliminate an account from a role. This function verifies the validity of the given account address and confirms the existence of the account within the specified role. Subsequently, it alters the role value stored in the system to "false," effectively detaching the account from the role.

Lastly, the library concludes by introducing a "has" function, which assesses whether an account possesses a specific role. By accepting an account address as input, this function produces a Boolean outcome indicating whether the account holds the designated role.

In essence, the "Roles.sol" contract presents an organized approach to overseeing roles within a blockchain-based platform. Through its capacity to facilitate role addition, removal, and querying, the contract contributes to maintaining structured access rights and permissions across the platform. This contract can be integrated into other smart contracts to ensure consistent and secure role management throughout the application.

4.3.1 Individual Roles

All the roles have similar function, for the sake of explanation let’s look at the Grocer role.

In the realm of managing roles for Grocers within a smart contract environment, the procedure involves utilizing the previously mentioned contract known as "Roles," which functions as a library to handle various role-centric tasks. The objective is to establish a specialized contract named "GrocerRole," utilizing this library to define and oversee roles for individuals serving as Grocers within a particular system.

The initiation of the "GrocerRole" contract is achieved by inheriting functionalities from the "Roles" library and incorporating its fundamental structure referred to as "Role." This structure serves as the foundational element for managing roles within the contract, providing functionalities for the addition, removal, and querying of role-related data.

To ensure transparent communication within the contract, two distinct events are introduced. These events, denoted as "GrocerAdded" and "GrocerRemoved," are designed to capture and broadcast notifications whenever Grocers are successfully added or removed from the role.

This event-driven approach ensures a clear and accessible record of changes pertaining to the composition of the role.

Within the confines of the "GrocerRole" contract, a private role named "Grocers" is defined using the inherited "Role" structure from the "Roles" library. This role acts as a container, holding addresses associated with the Grocer role within the system.

The process of assigning roles commences during the deployment of the contract. The constructor function, executed upon the creation of the contract, ensures the automatic addition of the deploying address as the initial Grocer. This step is pivotal in establishing the first participant possessing the Grocer role.

To enforce actions specific to the role, a modifier known as "grocerExists" is introduced. This modifier validates whether the invoking address possesses the requisite Grocer role. By doing so, it restricts the execution of certain functions solely to individuals holding the Grocer role, augmenting the security and reliability of role-based operations.

For simplified role verification, a function called "grocerPresent" is implemented. This function enables external callers to inquire about whether a given address holds the Grocer role. The function's output is a boolean value that signifies the presence or absence of the role for the provided address.

Ultimately, the "GrocerRole" contract establishes a framework for managing Grocers within a broader ecosystem of decentralized applications or blockchain networks. By combining the capabilities of the "Roles" library with specialized functionalities tailored to Grocers, this contract creates a comprehensive mechanism for role assignment, verification, and controlled access within the sphere of individuals assuming the role of Grocers within the system.

4.4 Supply Chain Smart Contract

1. Role-Based Access Control: The contract employs role-based access control to differentiate the participants in the supply chain. These roles include farmers, distributors, grocers, and customers. Each role is associated with specific permissions, and the modifiers

(onlyFarmer, onlyGrocer, etc.) ensure that only participants with the appropriate role can execute certain functions. For instance, only farmers can call the farmerShipsProduct function, and only retailers can call the itemShippedByRetailer function.

2. Enum State and Lifecycle Management: The State enum defines different states that a product can be in as it progresses through the supply chain stages. Each state represents a particular step in the product's lifecycle, from production to consumption. The modifiers (harvestedByFarmer, readyForSaleByFarmer, etc.) ensure that actions are only allowed when the product is in a specific state. This state management ensures that the product follows a predefined sequence of steps in the supply chain.

3. Event Emission: The contract emits events at various stages of the supply chain process. These events provide a way to notify external systems or interfaces about important actions taking place within the contract. For instance, the ProduceByFarmer event is emitted when a farmer produces a new item, and the PurchasedByDistributor event is emitted when a distributor purchases an item.

4. Tracking Product Information: The Item struct is used to store comprehensive information about each product. This includes data like the product's stock keeping unit (SKU), universal product code (UPC), owner's address, farmer's address, farm details, product notes, price, state, and more. This structured data storage enables accurate tracking of each product's attributes as it moves through the supply chain stages.

5. Product History and Transactions: The contract also maintains a Txblocks struct that records block numbers for various stages of the supply chain. This information allows stakeholders to verify the history of a product's journey through the blockchain. For instance, the FTD field stores the block number when the product is purchased by a retailer from a farmer.

6. Function Sequencing and State Transitions: The contract enforces a specific sequence of function calls and state transitions to ensure that the supply chain process follows a logical and secure flow. For example, a product must be produced by a farmer before it can be sold, purchased by a retailer, processed, packaged, sold by a retailer, purchased by a retailer, and

finally purchased by a customer. Each function call updates the product's state and triggers the appropriate events.

7. Data Retrieval Functions: The contract includes functions to retrieve various data about the items, including their attributes and historical block information. These retrieval functions provide transparency and traceability, allowing stakeholders to verify product information and history on the blockchain.

4.4 Flow of Supply Chain

The Ethereum-based 'SupplyChain' smart contract serves as the backbone of an open and transparent supply chain management system. It is designed to trace the movement of products, beginning from farmers, and ending with customers. To ensure secure access, the contract collaborates with distinct access control contracts that define roles such as 'FarmerRole,' 'RetailerRole,' 'GrocerRole,' and CustomerRole.' This setup guarantees that only authorized entities can execute specific actions within the contract.

At the heart of the system lies the concept of an 'Item' structure, which holds essential details about each product's journey. This includes critical information such as the stock unit, product ID, current owner, original farmer, farm-related information, product notes, price, and its current stage within the supply chain. This stage is represented using the 'State' enumeration, which encompasses different phases such as production, sale, shipment, and purchase.

The product's voyage starts with farmers creating items using the 'produceItemByFarmer' function. During this step, farmers provide details like the farm's name, relevant information, product notes, and pricing. Once this information is captured, the product is stored within the 'items' mapping. Additionally, a new entry is made within the 'itemsHistory' mapping to begin documenting the product's journey.

After creation, farmers can choose to sell their products by employing the 'farmerSellsItem' function. This action changes the product's state to "For Sale by Farmer," indicating its availability for purchase.

As the products move through the supply chain, retailers play a crucial role by purchasing items from farmers. This is achieved through the 'retailerBuysItem' function, wherein payment is made to the farmers. As the ownership shifts, the product's state is updated to "Purchased by Retailer."

Subsequent to purchase, farmers ship the products using 'farmerShipsProduct,' transitioning the state to "Shipped by Farmer." This signifies that the item is en route to its destination. When retailers receive the shipped items, they use the 'retailerReceivesProduct' function, marking the product's state as "Received by Retailer."

Retailers further contribute by processing and packaging items through 'retailerPackagesProduct,' advancing the state to "PackagedbyRetailer." The packaged product is then marked for sale using 'retailerSellsProduct,' indicating its availability for purchase by setting a price and changing the state to "ForSalebyRetailer."

As the supply chain advances, grocers enter the picture. They join the process by purchasing products from retailers with the 'productPurchasedByGrocer' function. This ensures proper payment, ownership transition, and an updated state of "Purchased by Retailer."

The interaction between retailers and grocers also includes logistical steps such as shipping and receiving. These steps are facilitated by the 'itemShippedByRetailer' and 'receivedItemByGrocer' functions, respectively. Grocers continue by selling the received items through 'sellItemByGrocer,' which sets a price and changes the state to "For Sale by Grocer."

Finally, customers participate by buying products from grocers using the 'customerBuysItem' function. The process involves verifying payment, transferring ownership, and changing the state to "Purchased by Customer."

Throughout this entire journey, the contract emits events for each significant action, contributing to transparency and accountability. By capturing the product's entire lifecycle and recording it on the blockchain, the 'SupplyChain' contract establishes trust and confidence in the supply chain management process, ensuring a dependable and transparent flow of goods from producers to end consumers.

5 Evaluation and Conclusion

5.1 Evaluation

The 'SupplyChain' smart contract is structured using Solidity and utilizes role-based access control to ensure that authorized participants can perform designated actions. Roles such as 'Farmer,' 'Retailer,' 'Grocer,' and 'Customer' are defined through separate access control contracts, providing a clear mechanism for assigning and verifying roles. This architecture guarantees that participants can only execute actions pertinent to their roles.

The heart of the 'SupplyChain' contract lies in its ability to transition products through distinct states that reflect their lifecycle stages. These states are defined using an 'enum' called 'State.' The contract allows for actions such as production, sale, shipment, and purchase, with each state change meticulously tracked and recorded on the blockchain. This ensures an accurate representation of the product's journey and provides a transparent audit trail.

The smart contract enhances transparency by emitting events at key stages of the supply chain process. These events, such as 'ProduceByFarmer,' 'ForSaleByFarmer,' 'PurchasedByRetailer,' and others, serve as notifications of significant actions. This feature fosters accountability and offers stakeholders a real-time view of the product's status and movement.

5.2 Conclusion

When comparing centralized traceability systems to blockchain-based traceability systems, the latter demonstrates significant efficiency gains. Blockchain technology, commonly applied in contexts where trust is a concern, such as supply chains, has ushered in a new level of transparency within these systems.

The implementation of tracking and tracing mechanisms has brought about comprehensive product details that are easily accessible to consumers. This includes information like manufacturing and purchase dates. Moreover, this approach ensures that retailers deliver products on time, contributing to maintaining product quality and reducing instances of counterfeit products.

In contrast to traditional systems that often involve a third party as an authoritative intermediary, blockchain technology promotes a more equitable arrangement. All participants are treated equally, and the technology enforces decentralization, eliminating the need for central authority intervention.

An essential feature of blockchain-based traceability systems is their automation, which reduces the need for manual intervention. This not only streamlines supply chain processes but also minimizes the potential for human errors, improving overall efficiency.

In summary, blockchain-based traceability systems offer advantages over centralized alternatives, including enhanced efficiency, transparency, detailed product information, quality assurance, elimination of third-party intermediaries, data security, and comprehensive automation. This technology transforms supply chain management, optimizing operations and building trust and reliability.

6 Future Work

In terms of scalability and efficiency, while the current smart contract showcases the viability of transparent supply chain management on the Ethereum blockchain, there's a need to address potential scalability challenges. As the number of participants and transactions increase, the Ethereum network might face congestion, leading to elevated gas fees and slower transaction processing times. Future work could concentrate on exploring layer 2 solutions, such as sidechains or state channels, to alleviate these concerns. This proactive approach aims to ensure that the smart contract remains cost-effective, swift, and reliable even as it scales to accommodate a larger network.

Integration with existing legacy systems emerges as another focal point for future enhancements. Combining the 'SupplyChain' smart contract with conventional supply chain management software requires creating standardized interfaces or APIs that facilitate seamless interaction. This synergy enables enterprises to adopt the blockchain-based contract without disrupting their current operations. Developing these integration mechanisms aligns with the practicality of assimilating the benefits of blockchain into established workflows while maintaining business continuity.

While the smart contract framework effectively manages product lifecycle states, addressing exceptional scenarios is equally vital. In this context, future exploration could focus on implementing mechanisms to handle unexpected occurrences, ranging from incorrect function calls to failed transactions. These automated fail-safe measures would bolster the contract's resilience, reducing the impact of disruptions and ensuring a consistent supply chain flow.

Considering real-world regulations and compliance is a crucial facet. Supply chains navigate industry-specific and regional regulatory frameworks. To align with these standards, future iterations could incorporate features that automatically enforce labelling requirements, certifications, and quality mandates. This evolution would transform the smart contract into a tool that not only enhances transparency but also ensures adherence to legal and industry-defined prerequisites.

Expanding participant involvement beyond the existing roles is also an avenue for growth. While the contract currently caters to farmers, retailers, grocers, and customers, supply chains encompass a wider array of stakeholders. These include suppliers, logistics providers, and auditors. Future advancements could broaden the scope to accommodate these participants, enriching the contract's representation of the entire supply chain ecosystem.

Enhancing usability and user experience is essential for broader adoption. Simplifying interaction with the contract requires intuitive interfaces and applications that mask the technical complexities. User-friendly dashboards, notifications, and alerts can significantly enhance the experience for participants with varying levels of technical expertise, encouraging wider adoption of the contract.

The security of the smart contract remains paramount. Regular security audits and continuous monitoring are necessary to identify and rectify vulnerabilities. Future efforts should focus on ensuring the contract's robustness through frequent security assessments and prompt updates. This approach safeguards the contract against evolving threats and underscores its reliability.

References

1. Van Boxstael S, Habib I, Jacxsens L, De Vocht M, Baert L, Van de Perre E, Rajkovic A, Lopez-Galvez F, Sampers I, Spanoghe P, De Meulenaer B (2013) Food safety issues in fresh produce: bacterial pathogens, viruses and pesticide residues indicated asmajor concerns by stakeholders in the fresh produce chain. Food Control 32(1):190–197

2. Gossner CM, Schlundt J, Ben Embarek P, Hird S, Lo-Fo-Wong D, Beltran JJ, Teoh KN, Tritscher A. The melamine incident: implications for international food and feed safety. Environ Health Perspect. 2009 Dec;117(12):1803-8. doi: 10.1289/ehp.0900949. Epub 2009 Aug 6. PMID: 20049196; PMCID: PMC2799451.

3. CollierMG, Khudyakov YE, Selvage D, Adams-CameronM, Epson E, Cronquist A, Jervis RH, Lamba K, Kimura AC, Sowadsky R, Hassan R (2014) Outbreak of hepatitis A in the USA associated with frozen pomegranate arils imported from Turkey: an epidemiological case study. Lancet Infect Diseas 14(10):976–981

4. Swan M (2015) Blockchain: blueprint for a new economy. O'Reilly Media, Inc, USA

5. Tian F (2016) An agri-food supply chain traceability system for China based on RFID & blockchain technology. IEEE: In 2016 13th International conference on service systems and service management (ICSSSM). 1–6

6. Kamath R (2018) Food traceability on blockchain:Walmart's pork and mango pilots with IBM. J British Blockchain Assoc 1(1):47–53

7. Tsang YP, Choy KL, Wu CH, Ho GTS, Lam HY (2019) Blockchain-driven IoT for food traceability with an integrated consensus mechanism. IEEE Access 7:129000–129017

8. Iansiti, M., & Lakhani, K. R., 2017. The truth about blockchain. Harvard Business Review, 95(1), 118-127.

9. Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008): 28.

10. Swan M(2015) Blockchain: blueprint for a new economy. O'Reilly Media, Inc, USA

11. Chen T, Li X, Luo X, Zhang X (2017) Under-optimized smart contracts devour your money. IEEE: In 2017 IEEE 24th International Conference on Software Analysis, Evolution, and Reengineering (SANER). 442–446 February

12. Majdalawieh, M., Nizamuddin, N., Alaraj, M. *et al.* Blockchain-based solution for Secure and Transparent Food Supply Chain Network. *Peer-to-Peer Netw. Appl.* **14**, 3831–3850 (2021). https://doi.org/10.1007/s12083-021-01196-1

13. Manos, Basil, and Ioannis Manikas. "Traceability in the Greek fresh produce sector: drivers and constraints." British food journal 112.6 (2010): 640-652.

14. Dabbene, F., Gay, P., and Tortia, C. 2014. "Traceability issues in FSC management: A review,"

15. Regattieri, A., Gamberi, M., and Manzini, R. 2007. "Traceability of food products: General framework and experimental evidence," Journal of Food Engineering (81:2), pp. 347–356.

16. Abeyratne, Saveen A., and Radmehr P. Monfared. "Blockchain, ready manufacturing supply chain, using a distributed ledger." (2016)

17. Liao, Pei-An, Hung-Hao Chang, and Chun-Yen Chang. "Why is the food traceability system unsuccessful in Taiwan? Empirical evidence