Using Smart Contracts to Automate a Food Hub

Rui Mira Gomes Gama da Costa

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisors: Prof. Kevin Christopher Gallagher
Prof. Miguel Nuno Dias Alves Pupo Correia

Examination Committee

Chairperson: Prof. António Manuel Ferreira Rito da Silva
Supervisor: Prof. Kevin Christopher Gallagher
Member of the Committee: Prof. Miguel Filipe Leitão Pardal

November 2023

Declaration
I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I would like to express my heartfelt gratitude to my parents for supporting me throughout my life, especially during my thesis. I am truly thankful for all the sacrifices you have made for me.

To my family, I want to thank you for your understanding and encouragement during this difficult process.

I am deeply grateful to my girlfriend for standing by me during the ups and downs of this journey.

I would also like to extend my appreciation to my dissertation supervisors, Prof. Kevin Gallagher and Prof. Miguel Correia, for their guidance, shared knowledge, and dedication to my thesis.

To everyone who has helped me in some way or another, thank you.

Abstract

Blockchain technology has been used for several purposes in the context of a supply chain, including traceability, transparency, and real-time monitoring systems. A supply chain is a complex process that requires a large number of intermediaries to deal with. Blockchain technology can be used to enable disintermediation, reduce transaction costs, and improve payment efficiency. This work creates a solution that uses smart contracts to automate the role of a food hub. The system automatically pairs multiple farmers in order to fulfill larger orders, thus integrating the farmers into the supply chain. It does it in a way that is fair, where we make sure that a farmer will eventually participate in an order. This work deploys three types of smart contracts that the actors, small and mid-sized farmers as well as buyers, can interact with. This system also automates the negotiation process with the buyer for farmers and effectively reduces the number of intermediaries. The system is evaluated using both qualitative and quantitative methods.

Keywords

Blockchain · Smart Contracts · Food Hubs · Aggregation · Disintermediation · Small and Mid-sized Farmers

Resumo

As tecnologias Blockchain têm sido utilizadas para vários propósitos no contexto das cadeias de fornecimento, incluindo rastreabilidade, transparência e sistemas de monitorização em tempo real. A cadeia de fornecimento é um processo complexo que requer a interação com um vasto número de intermediários. As tecnologias Blockchain podem ser utilizadas para reduzir o número de intermediários, reduzir custos de transação e melhorar a eficiência nos pagamentos. Este trabalho cria uma solução que usa contratos inteligentes para automatizar o papel de um Food Hub. O sistema proposto neste documento agrega automaticamente múltiplos agricultores para atender a pedidos de maior dimensão, integrando assim os agricultores na cadeia de fornecimento. O sistema agrega os agricultores de uma forma justa, onde é garantido que todos os agricultores acabarão por participar numa ordem. Este trabalho implementa três tipos de contratos inteligentes com os quais os intervenientes, pequenos e médios agricultores, bem como, compradores podem interagir. O sistema também automatiza o processo de negociação com o comprador e reduz eficazmente o número de intermediários, sendo avaliado tanto por métodos quantitativos como por métodos qualitativos.

Palavras Chave

Blockchain . Contratos Inteligentes . Food Hubs . Agregação . Desintermediação . Pequenos e Médios Agricultores

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

API	Application Programming Interface
CID	Content Identifier
DAO	Decentralized Autonomous Organization
IDE	Integrated Development Environment
IoT	Internet of Things
IPFS	InterPlanetary File System
VRF	Verifiable Random Function

1

Introduction

Contents

Blockchain technologies have been used for several purposes in the context of a supply chain. Currently, the main focus has been on traceability, not just for supply chains as a whole [1], but also in the food supply chain [2]. The reason behind this is that blockchain is a distributed ledger that stores information in a distributed way. This, together with cryptographic mechanisms, such as hashes, and replication, makes tampering with information difficult and thus provides greater trust compared to centralized storage systems. Consumers increasingly want more information about the origin of the products they consume and the manufacturing steps [1]. In addition to that, by providing better traceability systems, it also increases the level of food safety because it makes it easier for products not safe to consume to be recalled.

Although the use of blockchain technology has increased in recent years, the support of small farmers is still an area where little work has been done [2]. Small and mid-sized farmers don't have enough production capabilities to fulfill larger orders, thus having difficulties entering some supply chains. Besides this, the negotiation process for this kind of transactions is still very inefficient, where farmers and buyers need to trade several messages until an agreement is met. Current solutions for this problem are based on the use of intermediaries [3], [4]. Farmers either sell their product through farm gate sales or food hubs. In a farm gate sale, farmers can sell their product directly to consumers, but they can also sell to an intermediary. Another solution for farmers is to sell their products to food hubs. Food hubs are intermediaries that buy from small and mid-sized farmers and sell to wholesale buyers or consumers. Food hubs also aggregate multiple farmers to be able to sell to their buyers. Using farm gate sales or food hubs, farmers still need to use intermediaries that eat part of the profits that could have been directed to the farmers.

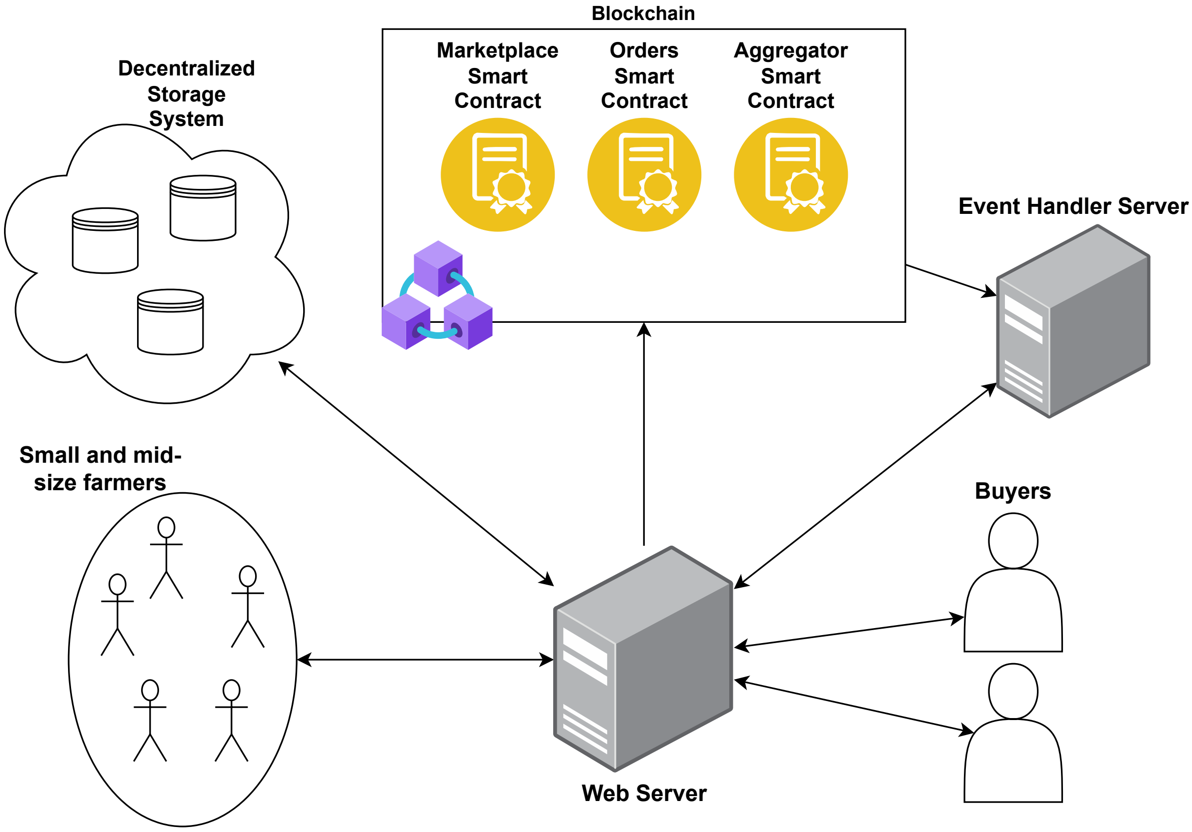

Figure 1.1: System architecture

This work creates a solution that uses smart contracts to automate the role of a food hub. Smart contracts pair multiple farmers together to fulfil larger orders from buyers, which can be restaurants, hospitals, etc. Figure 1.1 shows the proposed general architecture. This work deploys to the blockchain three types of smart contracts that the actors, small and mid-sized farmers as well as buyers, can interact with through the Web Server. The first type is the Marketplace Smart Contract, through which farmers are able to update their stocks. The second type, the Aggregation Smart Contract, is where buyers are able to place orders. The last type of smart contract acts as an Escrow Smart Contract. Every time an order is placed and the buyer agrees with the proposed offer, this smart contract holds the money of the order. This smart contract is then used to automatically pay the farmers who participate in each order. In addition to smart contracts, the system also makes use of a decentralized storage system to store the order history and other information provided by the farmers. This is done not just to pay less gas fees per transaction with the smart contracts but it also allows us to store information in a manner that no single entity controls it. This information is used to help in the aggregation process of the farmers, but it will be explained later in the document. The system also has a Web Server that is responsible for interacting directly with the smart contracts and the decentralized storage system to fulfil farmer and buyer requests, as well as another server, the Event Handler Server, that listens to important Blockchain events emitted by the Aggregator Smart Contract.

1.1 Objectives

This work seeks to create a smart contract system that **removes the need for intermediaries**, such as food hubs, by aggregating multiple farmers so that they can together fulfil larger orders. Furthermore, this work provides farmers not only with a gas fees consumption system that encourages them to participate in the supply chain through our system, but also with a **fairness factor** in the aggregation process. With this work is also expected that farmers will be able to obtain better prices for their crops, since the revenue will not have to be shared with an intermediary. Moreover, this work also presents a modular decentralized storage system that stores data separately about each product and only retrieve data about the desired product whenever necessary. The system also contributes with a way that automatically takes care of the negotiation process, provided by the implementation, and automatic payment, provided by both the use of blockchain technologies and the implementation.

1.2 Overview

The remainder of the document is structured as follows. Chapter 2 of this document will talk about blockchain technologies and more specifically smart contracts in supply chains, with more focus on the food supply chain. Chapter 3 describes in detail the solution for the presented problem. Chapter 4 defines the evaluation methods for the system. Finally, Chapter 5 concludes.

2

Background and Related Work

Contents

2.1 Background

A blockchain is a distributed ledger that stores records of information and transactions [5]. The nodes inside a blockchain network store a copy of the ledger and can verify its validity. Blockchain works by creating a chain of blocks that are linked together by the hash of the previous block. A hash function is a function that transforms the original data into a string that identifies the original data but that, at the same time, is impossible to obtain the original data having the output string. Using a consensus mechanism, this chain of blocks is then agreed upon by the miners in the Blockchain network. This makes it very hard to alter the data that is stored inside the blockchain because an attacker needs to alter not only the targeted block but also all the blocks that follow. The attacker then needs to convince the majority of nodes in the blockchain network to accept these changes. With the consensus mechanism, the hash of the previous blocks, and digital signatures, blockchain is able to provide trust, authentication, integrity, and non-repudiation [6]. Blockchain also enables smart contracts. Smart contracts are executable programs that run on the blockchain and can automatically enforce the agreed terms when the agreed predefined conditions are met [7].

2.2 Blockchain in Supply Chain

2.2.1 State of The Art

Blockchain technology has been used alongside other technologies such as Internet of Things (IoT), RFID, NFC, QR codes, and third-party sensors to get real-world information to improve agriculture and food supply chains ([2], [8]). Kamilaris et al. divide projects into 6 categories: food security, food safety, food integrity, support of small farmers, waste reduction and environmental awareness, and better supervision and management of the supply chain. Almost 50% of the projects studied in their survey are aimed at food integrity. Only 8% is aimed at supporting small farmers, and here the authors describe a smart contract solution that would automate a transaction between a cooperative and a distributor. Not much of the work was done to support small farmers and that is where this work comes in. This work plans to use smart contracts to integrate small farmers into the supply chain by eliminating the middleman.

Chang and Chen conducted a literature review of the current state and potential applications of blockchain technology in supply chain management. They found that the main topics of research, or issues that blockchain research is trying to solve, are the following. First, traceability and transparency, where blockchain technology has been used together with IoT devices, sensors, and smart contracts to provide traceability, transparency, and real-time monitoring systems. Second, stakeholder involvement and collaboration, where trust problems among parties and coordination among legacy systems are the main things to solve. Third, supply chain integration and digitalization, where blockchain technology can be used to enable disintermediation, reduce transaction costs, and improve payment efficiency. Lastly, common frameworks on blockchain-based platforms, where smart contracts have been playing an important role in the disintermediation of the supply chain. In addition to that, blockchain systems need to be objective-oriented, meaning that if a system aims to provide transparency a public blockchain is an adequate choice. But if the system is dealing with more sensitive information a private or a consortium chain is the preferred choice.

In [10] the authors summarize the state of the art in terms of blockchain implementations and its technologies in supply chains. They start by dividing the papers into two groups, one where the approach was more theoretical and less technical, and the other where the work was focused on the implementations or use cases. Most of the works reviewed fit into the first group. Regarding the areas where blockchain technologies have been proposed, Gonczol et al. found that much of the work done is not focused on a specific area, but, from the works focused on a specific area, agriculture is the area where most papers have been published. The most frequent use cases for blockchain technologies are traceability, transparency and counterfeit detection, trust management, and systems that integrate blockchain and IoT. The most used blockchains are Hyperledger Fabric, a permissioned blockchain that best suits use with sensitive data, and Ethereum, a permissionless blockchain. Some issues arise when implementing a system with blockchain technology, namely, data security and user privacy. These issues will not arise in this work since the information needed is information that farmers already make available in other types of marketplaces. Two benefits that the authors identify in their study and are relevant to this work are that through blockchain farmers can obtain a better price and transactions are facilitated. Most of the work did not explain system implementation details or the technical aspects of the designed system.

In [11] the authors divide the uses of blockchain technologies in the agricultural sector into 4 categories: traceability and food authentication, smart farming data management, trade finance in the supply chain, and other information management systems. As seen in other surveys, traceability is by far the goal for which blockchain technologies are used. They also describe the different stages of the smart contract implementation life cycle along with the challenges of each stage. In the creation stage, there is the need to create readable and optimized code. In smart contract deployment, correctness and control flow are the most important factors. In the execution stage, the smart contract must be executed efficiently. Finally, in the completion stage, smart contract security is the most important factor. They describe three main problems. Scalability problem, where lots of information needs to be stored in the blockchain, and they suggest the use of systems like an InterPlanetary File System (IPFS), a mix between centralized databases and blockchain. A problem in integration with legacy systems where they believe the solution relies on the use of cloud-based services. Finally, a security and privacy problem is addressed in which they suggest the use of a private blockchain, and only data that needs to be shared is stored in a consortium chain.

Pournader et al. analyzed the current state of the literature regarding blockchain applications in supply chains, transport, and logistics. They divided the literature into four groups: Technology, Trust, Trade and Traceability, and Transparency. The most researched group is Technology, and the least researched groups are Trade and Traceability, and Transparency. Blockchain technologies have been used to increase synchronization among supply chain entities, proof of identity in transactions, increase trust, traceability and safety, especially in the food supply chain, and facilitate and automate payments. One of the areas that they believe blockchain technologies can address that has not been addressed yet is procurement, meaning, discovering goods and agreeing on terms to purchase those goods. This work addresses this idea by aggregating multiple food producers automatically to fulfill larger orders.

In [1] the authors conducted a literature review of current applications made with blockchain technologies in supply chains. They analyzed 106 papers and with those they identified research themes, research methodologies, and industries where blockchain technologies have been applied. They divided the research themes into three categories: Impact, Function, and Configuration. The most researched theme, with more than half of the papers, was in the Function category. From those, most of the papers focused on developing a traceability capability and only a few papers focused their research on logistics and coordination. Regarding the research methodologies, the authors divided the articles into four categories: Conceptual, Empirical, Modelling, and Technical. The two largest research methodologies were Conceptual, explained by the fact that blockchain is an emerging topic, and Empirical, with most of the studies carrying out qualitative research. Few studies have focused on system implementations. Ethereum and Hyperledger are the most widely used technologies. In terms of industries, agriculture is the field in which most of the research has been focused. This is because consumers want more information about the products, traceability of products, and food safety reasons.

2.2.2 Uses, Potential and Challenges

Chang et al. proposed a private blockchain framework that promotes transparency and distribution collaboration in the supply chain process. They created what they called a blockchain-based business process reengineering framework and evaluated the benefits and values of their approach. Their proposed framework is based on six types of smart contracts that they divide into six processes. The transaction process, where are the contracts that have anything to do with logistics (shipping and inspection processes), suppliers, and buyers, one for each element mentioned. The payment process where the smart contract is designed in a way that ensures payment completeness. And finally, the data accessing process where there are two processes responsible for communications with off-chain information. This smart contract communicates with another smart contract that acts as a load balancer and a gateway. They evaluate their framework concerning traceability, data storage, privacy, cost reduction, cash liquidity, payment, and automation. They found that payment via a digital currency was feasible, blockchain can be used to track logistics progress and cash flows. Although smart contracts have been used to disintermediate a transaction, the aim of the study is shifted more towards tracking an order. The pairing of buyers with suppliers is not addressed in their study.

In [14] the authors describe the benefits of digital technologies for smallholder farmers, identify challenges that are still unresolved, and the impact of hyper-transparency in global value chains. They extend the definition of inclusion in the smallholder market so that it also includes the competitiveness of farmers in local and international markets. They argue that hyper-transparency has six characteristics: data collection of mobile devices and sensors instead of humans, data collection in real-time, decision-making by algorithms, materiality of information (digital information), disintermediation, and accessibility to information. They summarise the potential benefits of hyper-transparency with various digital technologies in use, from drones and satellites to blockchain. In particular, with the blockchain technologies used, they say that one of the benefits is access to services such as obtaining credit or facilitating trade finance. Another, and more relevant benefit of this work, about the usage of blockchain, is that with it small farmers can lower transition costs by cutting or reducing the middleman. Together with IoT devices, blockchain can be used for traceability, certification, and food safety. There are also some limitations to the use of digital technologies for small farmers. Often access to such technologies is difficult due to lack of infrastructure or low digital literacy rates. In addition to that, sometimes farmers are reluctant to adopt new technologies.

In [8] the authors analyzed six companies from the agricultural sector that used blockchain technologies concerning four components: Technique component, Knowledge component, Organization component, and product component. They studied companies through whitepapers, reports, and interviews when possible. Köhler and Pizzol found that non-blockchain technologies, such as RFID, are usually used together with blockchain technologies. They also found that there were companies that used external databases and only stored the hash of the information inside the blockchain. To take full advantage of blockchain technology, education in the area is still needed. They state that by introducing the use of blockchain technologies it's possible to reduce the need for intermediaries due to direct communication between different supply chain actors. The use of blockchain technologies increased transparency, traceability, authenticity, and trust, especially in the short food supply chain.

The impact of the adoption of blockchain technology in three different food supply chains of a European retailer was studied by Stranieri et al.. The adopted system had the objective to provide more transparency to the consumers and traceability to the supply chain actors. In their study, three processors and one retailer participated. The authors did first a series of face-to-face interviews to identify the project's relevant actors in the implementation of the blockchain system, then those actors were interviewed to understand the impact that the system implementation had. Their results show that profits increased as a result of an increase in sales. One of the actors also reported a decrease in product loss. Customer satisfaction increased because customers had access to information such as the producers' history and product production chain. The implementation of blockchain technologies also improved the accessibility and availability of information to suppliers and retailers. Another thing that the implementation of the system provided was better collaboration and management of vertical relationships. The interviewees also reported that production and distribution costs didn't suffer significant changes.

In [16] an architecture for quality management in the short food supply chain is proposed. The authors, by doing a literature review, start by identifying the short food supply chain stakeholder and quality-related requirements. Stakeholder related requirements were, for example, equality, fair trade, value added, location, convenience, reduced risks, and traceability. Quality-related requirements were, for example, fairness, ecological sustainability, origin, product and process quality, and freshness. After identifying these requirements a multi-level architecture for quality management is explained. The architecture consists of six layers. The first layer defines the stakeholders. The second layer, IoT and communication layer, is where all the devices and sensors collect information are. The third layer is where the blockchain enters. Important information is stored inside the blockchain which provides immutable data. The fourth layer, where smart contracts can also be used to enable automation in payments for example, is for services and analysis, such as data analysis and forecasting. The last two layers are for trust mechanisms and applications. Based on these ideas, and especially on the requirements identified by their literature review, the system proposed in this work will try to be fair, reduce risks, and add value to the short supply chain by reducing intermediaries. Besides that, since the system will be implemented with blockchain technologies it will be easily improved to meet other requirements, such as traceability, that have been researched with blockchain.

In [17] the authors use the PPT model (People, Process, Technology) to try to create a framework for the food supply chain that classifies opportunities and constraints into 4 categories: people, process, technology, and performance. This framework is focused on the use of blockchain in the food supply chain, which has the potential to eliminate intermediaries, reduce transaction time, reduce cost, reduce risk fraud, real-time monitoring, data verification, and others. Much of this is done with the help of other technologies such as IoT devices, RFID, etc. Their findings suggest that most opportunities are in the people category, where blockchain can be used to redistribute food surplus, be used for traceability and transparency, disintermediation, and others. Some of the impediments they found in this category is that a transformation takes time because there is the need to have skilled persons and it is needed some additional investment. In the process category, they found that there is still a need for regulation and legal acceptance. The category they found that had the most impediments was the technology category where there are security concerns, high implementation costs, and lack of standards. Lastly, they claim that it is still too early to be able to evaluate the performance category and that more studies and time are needed, but they claim that balancing people and technology elements as well as process and technology elements would improve performance.

2.3 Smart Contracts in Supply Chain

2.3.1 Management and Traceability

Hasan et al. propose a blockchain-based solution to efficiently manage the supply chain when shipping goods. Using a set of smart contracts on the Ethereum Blockchain, the authors propose a system that together with IoT sensors is capable of monitoring shipment conditions, automating payments, and ensuring that the shipment is received by the legitimate buyer. Their system provides tracking of shipments through GPS. There are also sensors for temperature measurement, a pressure sensor, and a sensor for detecting falls or drops. Whenever the smart container detects a violation of the agreed conditions based on the information provided by the sensor, the system can notify, via smart contracts, the sender and the receiver. They have proved that smart contracts can be used to create trust between suppliers and buyers without the need for a third party. This work will use the same technology, smart contracts, to connect multiple suppliers, that alone couldn't fulfill a big order, with a buyer, without the need of someone who collects food from the multiple suppliers.

A smart contract based system to provide traceability, by storing transaction history information, of products in the smart chain has been proposed [19]. The proposed system consists of several smart contracts. The first is the Product Registration Contract. This smart contract serves for product registration by the owner of the products. After registration this smart contract automatically deploys a new smart contract, the Batch Addition Contract, where the product owner can call a function to add information about the product batch. This smart contract also deploys a new smart contract, the Transaction Update Contract, which provides the sellers of a product batch with a function to update the transaction history of that batch. Their system is implemented using the Solidity programming language, a programming language provided by the Ethereum blockchain. The system also contains a series of web pages that interact with the described smart contracts. Since the transaction history is stored inside the blockchain when a consumer wants to get information about the product history, he simply needs to provide a specific web page with the product code and batch number, and the history will be retrieved from the smart contract. Since the system is implemented using blockchain technologies it provides the stakeholders with data accessibility, data immutability, and system autonomy, because batch and transaction contracts are deployed automatically. This work will implement some ideas presented in this paper. For example, the idea of having multiple smart contracts, each with its purpose and functionality.

A system that uses blockchain technologies to provide data integrity to a fish farm [20]. The farm is made up of several IoT sensors and actuators that measure and control environmental variables in the fish farm, such as the water level. They use two smart contracts in their system, a smart contract for data and device monitoring and management, and a policy smart contract that controls access to each resource by containing a set of rules for farmers, farm owners, sensors, and actuators. Their system is implemented using Hyperledger Fabric, which is a permissioned blockchain, to allow only authorized users to interact with the system. Each node in their permissioned blockchain network has an off-chain storage system that stores all the current state values of the environmental values and only the history of transactions is stored inside the blockchain network. The interactions between the legacy fish farm system and the blockchain network are carried out through an Application Programming Interface (API). A simulation is carried out taking only into account the water level. They measured the performance of their system by evaluating the throughput, arriving at the conclusion that the system was scalable, being able to handle up to 2500 transactions per second, the optimal number, and latency, concluding that after the optimal number of transactions per second is reached, the latency would start to increase.

In [21] the authors propose a framework that uses smart contracts to execute transactions and provide traceability to all the interested parties. In their proposal, the smart contracts are created by the farmers. Farmers can update smart contracts and an IPFS with relevant crop information. The farmer and the processor then negotiate the conditions of the sale, and a set of functions are called on the smart contract to complete the transaction. The remaining transaction in the supply chain occurs similarly. The entity that pretends to buy food calls a specific function in the smart contract, which then notifies the seller, and when the transaction occurs, all relevant participants are notified. The traceability of their framework is provided by a QR code attached to the package that the consumer can read to obtain the desired information.

However, there is still room for improvement. In their framework, if a processor or distributor needs to buy food from multiple sources, they have to contact each of the sources and negotiate with them, making the process extremely inefficient and time consuming. The system described in this work will not only aggregate farmers but will also automate the negotiation process.

2.3.2 Potential and Challenges

De Giovanni propose a simple supply chain game in which there exist only two firms, one supplier, and one retailer. They initially propose two scenarios. One is a traditional marketplace and the other is where the supply chain is managed on the blockchain. They found that when the firms used blockchain, the retailer would order larger quantities and charge higher prices due to the transaction efficiency compared to the traditional online platform. Inefficiency of the transaction of the traditional online platform/marketplace also reduced profits and discouraged sales. To cope with some negative effects that can be introduced by the adoption of the blockchain the authors also propose two smart contracts. A smart contract that determines a wholesale price so that the supplier has an incentive to carry the transaction on the blockchain. And a smart contract that determines a shared quantity of profits so that all supply chain members benefit from the implementation of blockchain technologies. The implementation of these smart contracts both increased retailers' profits and incentivized suppliers to participate in the supply chain through the blockchain, which also increased their profits.

In [7] the authors conducted a survey on smart contracts and their challenges. Smart contracts are a piece of code that represents contract clauses that do not rely on a trusted third party to be executed. A smart contract automatically executes when certain predefined conditions are met. Smart contracts have the benefit of reducing risk, cutting administrative costs and service costs, and improving the efficiency of business processes. They say that a smart contract has 4 phases, each with its own challenges. Creation phase, with readability issues and functional issues like overcharging. Deployment phase, with contract correctness issues and control flow issues because it is not immutable. Execution phase, with Trust issues in third-party devices that smart contracts need to obtain real-time information, order dependency of transactions, and execution efficiency. Completion phase, with privacy and security concerns. They say that there are six major categories to which smart contracts can be applied. Internet of things, Distributed system security, Finance, Data provenance, Public Sector, and Sharing economy. In the last one, they say that some of the benefits of using smart contracts are the reduction of consumer costs and the reduction of environmental impacts. Therefore, this work plans to use smart contracts as a way to reduce agricultural supply chain logistics, which can also be applied to food hubs and effectively cut transportation costs by half.

2.4 Food Hubs

2.4.1 State of The Art and Concepts

Berti and Mulligan extended previous definitions and defined food hubs as organizations that act as intermediaries between small farmers and wholesale buyers or consumers and provide a platform to aggregate and distribute small farmers' products. Food hubs have five main functions: Logistics, such as aggregation and distribution of products, Marketing, such as searching for new markets, Product services, such as providing storage facilities, Consultancy Services, such as providing guidelines, and ”Web of practices”, acting as a sharing platform of knowledge and experiences. Food hubs have also taken advantage of digital technologies and expanded in this area. The authors identify two types of online food hubs. The first type is simple online platforms that connect buyers and producers but don't handle transactions or logistics. The second type acts as an online market and can carry out payments and coordinate logistics.

In a more recent survey [23], the authors summarize the state of the art in Food Value Chains and Food Hubs. Food value chains are defined as collaborative networks where players help each other in the production, logistics, distribution, and marketing of products. They found that Food Value Chains and Food Hubs have gained popularity in the academic community. Most of the work done in these areas has been evaluated recurring to three methods: Phenomenology, Mixed Methods, and Statistical analysis. Most of the research is focused on the Support Services function of Food Hubs, with less focus on other functions such as marketing. Another conclusion from this paper is that food hub studies lack the use of simulations and mathematical models, which the authors say are essential to designing optimal networks.

In [24] the authors highlight the importance and the role of food hubs in the supply chain. According to their study, food hubs together with digital technologies, such as e-commerce platforms, have the potential to overcome constraints and facilitate communication, aggregation, and organization. There are three models of food hubs: the Direct Consumer Model, the Wholesale Operation Model, and the Hybrid Model. The first refers to the type of food hub that only serves as a platform that connects farmers and final consumers directly. The second refers to food hubs that sell to actors that operate in the wholesale market by collecting food from multiple farmers and distributing it. The last model refers to food hubs that implement both of the previous models. There are also two types of online food hubs. The first type, ”virtual food hub”, is just an online platform that serves as a meeting place for farmers and buyers. The second type, ”online food hub networks”, is a platform where it is possible to place orders and process payments. Although in the United States food hubs are already established systems, in Europe there are very few of them. Digital food hubs could be a way to rapidly implement a system that helps small farmers not just in Europe but all around the world.

2.4.2 Case Studies

In [25] a case study of a Santa Barbara food hub is presented. The authors focused their study on understanding how the food hub operated, and their requirements to work with farmers to provide an alternative food system that the health of people, the environment impact, and the local economy. Like other food hubs, the products they buy are stored in a warehouse and only then are delivered to the buyer. This food hub chooses to work with farmers based on chemicals used, prices, quality of the products, and the fact that farmers don't compete to do business with the food hub, organization, and the food hub doesn't discard the farmer if they found a lower price offer. In some cases, the food hub would directly connect farmers with buyers making the purchase more efficient and reducing transportation costs. From the farmer's perspective, they choose to work with the food hub based on trust that comes from personal relationships and reliability and flexibility. Factors that prevented farmers from doing more business with the food hub were the fact that the food hub had limited purchasing capacity and that the food hub did not accept all types of crops. By implementing a system that automatically aggregates multiple farmers, this work eliminates the need for intermediaries. Furthermore, by implementing such a system with blockchain technologies, this work provides immediate trust among all actors that otherwise would only be possible with long-lasting personal relationships.

Marusak et al., through several case studies, studied the adoption of best logistic practices in response to the disruption caused by the COVID-19 pandemic. They interviewed 25 regional food supply chain actors during May to July 2020, and of those, only seven demonstrated the implementation of best practices in logistics to respond to pandemic challenges. The practices implemented the most were horizontal collaboration, four out of seven, and transportation outsourcing, three out of seven. The reasons, according to the authors, for which farmers were able to overcome the pandemic challenges were their willingness to adapt and the way they leveraged communication technologies to facilitate the implementation of best practices. Implementing these best practices allowed the reduction of costs and the increase in logistics capabilities. Regional food supply chains are also suitable for the sustainable development of communities where they are present. The use of communication technologies facilitated the collaboration between regional food supply chain actors, which in turn allowed the implementation of best practices in logistics that would be difficult to implement alone.

2.4.3 Challenges and Models

In [27], through a review of the literature, the authors identify the challenges of regional food supply chain logistics and identify best practices and recommendations to solve these challenges, as well as the current practices implemented by regional food supply chains. The identified challenges are related to transportation, warehouse, and inventory management. They found that among the most recommended practices are food tracking and traceability, a topic that has been researched with the use of blockchain technologies, horizontal collaboration, which means collaboration between competitors, and improved supplier reliability. Although these are the recommended practices their findings indicate that the most frequently implemented practices are not those. The most frequently implemented practice is the use of an organization that handles logistics, such as transportation and warehousing, followed by efficient vehicle utilization and vertical collaboration. A research gap identified by the authors in their study is the lack of logistical solutions that are not financially viable for small farmers. This work will provide a way for small farmers to enter the supply chain without the need for an intermediary and with minimal costs. In addition to that, the proposal in this document will also make it easier for farmers to implement the best recommended practices, namely tracking and food traceability.

Ioannis et al. propose a conceptual food hub model based on the ideas of community and innovation. There are two main approaches for food hubs, a value-based approach and a sustainability approach. In the first approach, the focus is on the economic aspects, such as logistics and marketing operations, intending to achieve better profits and efficiently access the market. In the second approach, the focus is on the sustainable aspects with the goals more focused on food security and the reduction of environmental impacts. Their proposed model combines both of these approaches and creates a new one. Their proposed model introduces the concept of an e-community and has the goals of sharing knowledge, food sustainability and security, and cooperation among all stakeholders in the supply chain. The introduction of an e-community facilitates knowledge sharing with other small farmers, as well as the use of remote and financial services, and facilitates market access thus eliminating the middleman. The proposed solution will take advantage of this idea of e-community and introduce an e-marketplace that automatically negotiates for farmers based on their inputs.

In [4] the authors suggest a series of economic, environmental, and social indicators and those evaluate several types of short and long food supply chains. They choose 10, of the most common types of supply chain, 6 short supply chains, and 4 long supply chains. Some of the indicators proposed are price premium, which means the price that a farmer can get compared to the average retail price and the chain value added for economic indicators, the distance traveled per kilogram of product, and the carbon footprint for environmental indicators, bargain power, and chain evaluation, a self-evaluation made by the farmer for social indicators. Their finding suggests that short food supply chains provide farmers with better prices and increased value added to the supply chain. In terms of environmental indicators, their study suggests that long food supply chains are better because it is possible to use transportation systems more efficiently. Lastly, in terms of social indicators, their findings suggest that short food supply chains are better in terms of price achieved and payment security, and farmers have greater bargaining power. Their study demonstrates that by farmers intervening in the short food supply chain they achieve the objectives of the system proposed in this document.

A hybrid model for food warehouse operations that tries to achieve logistical and operational efficiency has been proposed [29]. Farmers deliver food to food hubs at the time that is most convenient for them. As a result, queues are formed in the food hub warehouse. To solve this problem, the authors propose a system with agent-based modeling, for decisions based on farmer delivery schedules, and discrete-event simulation, for the simulation of warehouse operations. First, the authors collect answers to a survey from several farmers, and with the answers they divide farmers into four groups. This data is given to the agent-based modeling, which then produces the farmer schedule and arrival times, as well as warehouse personnel assignments. This is then given to the discrete-event simulation that simulates every stage of warehouse operations. To encourage farmers to schedule their arrival, the authors propose a scenario in which the food hub manager offers money to farmers. Results show that a monetary incentive didn't produce substantial changes. They then study a scenario where a food hub manager uses the model to schedule personnel and they found that this scheduling of personnel can improve operational efficiency. In this proposed model there is still the need for a central warehouse that receives all the farmer's harvests and then distributes them to buyers. In the system proposed in this document, there will be no need for this intermediary, since the farmer will directly connect and send his harvest to the buyer.

2.5 Online Blockchain Marketplaces

In [5] the authors explain blockchain concepts and how it works. They then proceed to explain a proof-of-concept market for the energy sector based on blockchain technology. In their market, producers publish their energy offers on the blockchain. A consumer would then read the offers and accept the one that most suited his needs, according to the paper, the cheapest offer. Although a simple idea, the authors show that blockchain technology can be used to connect suppliers with consumers/retailers. Furthermore, their concept together with smart contracts can be used to make more complex transactions, such as trying to bundle multiple small producer offers together to fulfill bigger consumer/retailer orders.

The system proposed in this paper is not entirely new. Kumarathunga et al. proposed a system that aggregates multiple farmers in a group and makes available the farmers' crops to be purchased. Their system aggregates farmers into groups based on the expected harvesting date and the location of the farmers and the quantities are then made available to be purchased through a web site. A buyer proposes a price for the crops and farmers can choose to accept or negotiate the price until an agreement is reached. At this point, their system uses smart contracts to enforce the agreed conditions between farmers and the buyer. Farmers transfer 10% of the agreed price to a smart contract, and the buyer transfers the total amount to the smart contract. At the due date of the contract, and if the agreed conditions are met, the smart contract automatically transfers the money to the farmers. Through this collective marketing strategy, farmers can get better prices. Although the system just described is similar to what this work proposes, some things can be improved. For example, the aggregation mechanism is not described and the aggregation itself is done outside of the blockchain. The reasons for this are not explained. By doing the aggregation inside the blockchain, the solution proposed in this document brings trust not only between the farmer and the buyer but also between the farmers themselves. Besides, the negotiation process is still made manually and is very time consuming. By asking the farmers with a wanted price and a threshold this process can be made automatic and almost instantaneous.

In [31] the authors propose BeIMP, a platform based on blockchain technology that enables manufacturers to form contracts and to perform transactions. Their proposal integrates the participation of producers/sellers, buyers, financial institutions that provide loans if necessary, insurance companies, and regulatory agencies. Their system is implemented using a consortium blockchain where the actors participate by joining with three Ethereum nodes. In their system, blockchain is used as a distributed database. On the blockchain, they only store important information, such as contracts and orders, for security reasons. Other information is stored off-chain. Their system is composed of six types of smart contracts. One to manage user data, another to manage contract specifications, another to manage order tasks, such as placing the order or order confirmations, another to deal with orders that the buyer wants to resell, another for shipment functionalities, and the last one to manage financial responsibilities such as insurance coverage and loan responsibilities. A producer starts by creating a new contract specification so that a buyer can place an order. Banks confirm the order when they receive the deposits from the buyer and the seller. The seller can also request loans from the bank to maintain operations if needed. The bank can take out insurance to ensure loan payment.

The rest of the payment process is carried out outside of their designed system. They evaluate their system on gas fees consumption by calling several functions confirming that the more storage the function modifies the higher the gas fee is. They also evaluated the throughput of the functions of their system to see the number of transactions per second handled.

FarMarketplace [32] is a digital marketplace based on blockchain technologies for agricultural products. Their proposed system is composed of three main components. An app, from which farmers can publish their offers, contracts, or bids, the digital platform where the trades happen, and the blockchain itself, together with the smart contracts and an off-chain storage system, only the hash is stored inside the blockchain. Buyers and deliverers first inform, via one smart contract, that they are interested in receiving, respectively, notifications about new offers and new pending deliveries. To publish a new offer, farmers interact with a smart contract through the app and then create a new smart contract, for that specific offer. This new smart contract is always up to date with the current state of the process. Farmers and possible buyers are then notified. The buyer can then purchase an offer, and at this point, provided that the purchase conditions are met, the deliverers are notified. When a deliverer accepts the delivery, and the state of the smart contract is valid, the farmer and buyer are notified. When delivered, the farmer and the buyer confirm reception, the smart contract state is updated, and every actor in the process can use a function to rate the other actors. Their system was implemented using Ethereum. They do a pre-experimentation phase to determine the number and duration of the tests. They evaluated their system for latency and throughput and found that when the emission rate of new transactions was bigger than the maximum number of transactions per block was reached, latency would increase and the throughput would be constant. Before this point, latency was equal to the time of mining a block and throughput was equal to the emission rate of new transactions.

A system, based on five smart contracts, has been proposed that connects multiple actors in the healthcare supply chain [33]. Group Purchasing Organizations are organizations that merge multiple orders from healthcare providers and negotiate with manufacturers to obtain better selling prices through economies of scale. Their proposed system is composed of five smart contracts. The organization initiates a registration smart contract where all the actors (manufacturers, distributors, and healthcare providers) go to register themselves. A second smart contract is used when an order needs to be placed, purchase negotiation smart contract. The organization proposes a price with a quantity to the manufacturer that can make a counter-offer. This occurs until an agreement is met. After an agreement, healthcare providers interact with another smart contract, the Purchase Order smart contract, to purchase the final order and assign a distributor to the product delivery. The last two contracts are for rebate settlements due to differences in the distributor price and the negotiated price, and loyalty rewards that the organization pays to healthcare providers. They evaluated their system in terms of gas fees that need to be paid for the execution of each smart contract function and found that their system was economically feasible. This idea can be extended to the problem this work is trying to solve with the necessary modifications, such as making the negotiations benefit small farmers and reducing intermediaries.

2.6 Summary

Paper	Area	Blockchain Type	Storage System	Market Linkage	Analysis
[5]	Energy Sector	Private	Not specified	One-to-One	Qualitative
[30]	Agricultural Sector	Not specified	Not specified	Many-One-Many	Qualitative
[31]	Undefined	Consortium	Mixed	One-to-One	Qualitative and Quantitative
[32]	Agricultural Sector	Public	Off-chain but hash stored on-chain	One-to-One	Qualitative and Quantitative
[33]	Healthcare Sector	Public	IPFS	One-to-Many	Qualitative and Quantitative

Table 2.1: Blockchain Marketplaces Synthesis Table.

Most of the marketplaces that use some kind of blockchain technology do not address the problem this work will try to solve. Table 2.1 summarizes several marketplaces relevant to this work. Only the work presented in [30] tries to aggregate multiple farmers together, but their work lacks implementation details, does not use blockchain technologies for all the transaction steps, and only performs qualitative analysis. Additionally, the goal is for the system to be open to any small or mid-size farmer and buyer, so the system needs to be implemented on a public blockchain.

Other works ([31], [32]) use some system that stores information both on-chain and off-chain. In [31] the system stores all the information in a centralized database and the most important information is also stored in the blockchain. In [32] the system stores all information in a centralized database and stores the hash of that information in the blockchain. The only system that uses an IPFS is the one proposed in [33]. The system is also implemented in a public blockchain. The work described in this document is also very similar to the one done in that document, but its focus is on helping customers in the healthcare sector, whereas the work described in this document will focus on helping producers in the food supply chain.

Similarly with other works ([31], [32], [33]) the plan is to evaluate this work both with qualitative and quantitative methods, such as gas fees consumption, throughput, and security properties.

3

Solution

Contents

3.1 Description

The main problem that this project solves is for small and medium-sized farmers to be able to fulfill larger client orders that otherwise they could not. Currently, the best solution for these farmers is to sell their products to food hubs that then sell to manufacturers or consumers. Inevitably, this leads to lower prices for farmers due to intermediaries in the food supply chain. This work develops a smart contract system that dynamically pairs multiple farmers. The system also automates the negotiation process with the buyer them and effectively reduces the number of intermediaries, thus achieving better pricing for farmers. To be able to accomplish this, the system makes use of three distinct types of smart contracts: Marketplace Smart Contract, Aggregation Smart Contract, and Orders Smart Contract. The Marketplace Smart Contract stores the Content Identifier (CID)s of the Stocks and the Orders of each product. The Aggregation Smart Contract is the smart contract that checks if an order is possible and performs the aggregation of farmers. The Orders Smart Contract holds the funds and pays farmers when he is told to do so. The initial idea was to have one Orders Smart Contract, or at least one smart contract with the same purpose, to be deployed for each new accepted order. It is easy to see that in the long run, this wouldn't be feasible because after every farmer's order has been paid, the smart contract would stop having any utility. This would be very expensive to do without the need for it. The next idea to be discussed was to have several smart contracts of this type and keep reusing them, one for each order too, but trying to always use one smart contract that wasn't holding the funds of any other order. This idea would have the same problem as the previous one and so the final decision was for the system to deploy only one Orders Smart Contract. The implementation of the system consists of only one Orders Smart Contract, that holds all the funds, for every order, independent of the product, and transfers the money to the farmers whenever it is requested. By doing this only one smart contract of this type is deployed, no smart contract is left unused after a while and a lot is saved in gas that would be consumed to deploy all of the unnecessary smart contracts. The decision to separate the smart contracts into three types was not just for maintainability but also for auditable and gas consumption reasons. The system is auditable. It implements mechanisms that make it so that someone can look at the aggregation process and reason if the process is accomplishing the objectives or not. If it is not possible, a new Aggregation Smart Contract can be deployed with a new aggregation process that better satisfies the objectives of the system. Check Section 3.4 for more details about the auditable aspect of the system and the aggregation process that was implemented. If the aggregation process is inside the same smart contract as the other functionalities, the new smart contract being deployed would contain more code, and more instructions, and thus would consume more gas. This gas consumed in excess would be for functionalities that were already implemented and did not change.

In a Blockchain Network used in this work, a price is paid, called gas fees, for each executed instruction, and the higher the amount of information stored inside this blockchain, the higher the amount paid in gas fees. Because of that the system makes use of an IPFS. IPFS is a peer-to-peer data storage system that stores data in a distributed manner [11]. This also keeps information stored in a decentralized manner where no single entity controls it. The IPFS stores information on the farmer's stock and order history. When a farmer calls the stock update function, the smart contract stores in the IPFS the information provided by the farmer, setting the state of the stock to available. The smart contract separates stocks of different products, storing them separately. For example, a farmer tells the smart contract that he has both carrots and tomatoes available for purchase. The smart contract stores these stock updates in different files inside the IPFS. By doing this, upon the initiation of the aggregation process, it is possible to retrieve only the stocks in the IPFS that are from the product the buyer wants. This is an advantage because it also allows for less gas consumption. After all, there are fewer stocks to filter. Section 3.4 explains this in more detail.

The system also makes use of a Web Server through which farmers and buyers make their requests. The initial plan was not to use this Web Server but, because of a particular situation, the decision to include a central Server was made. If no Web Server existed the farmers and buyers would be the ones interacting with the decentralized storage system, IPFS, and the smart contracts. By being the ones exchanging the information between smart contract and the IPFS, there could be a case where one farmer, for example, could remove other farmers' stocks from the available stocks. This situation is not desired and that is the main reason for the use of the Web Server in this system. Although such a problem could be easily tracked, since it is possible to check the changes made to the CID in the Marketplace Smart Contract and the blockchain address that made them, the decision to create the Web Server was made to have a higher degree of certainty that no Stock was being unfairly removed. Another server, the Event Handler Server, is used in the system. This server is used after an order request is made to the Orders Smart Contract. This Aggregator Smart Contract emits an event that is caught by the Event Handler Server. This server then sends that event to be processed by the Web Server. Events were used in this case because that is the way to retrieve information from the Blockchain after a writing operation, or a transaction. Section 3.4 explains the writing that takes place in the aggregation process. The Event Handler Server was used so that we could retrieve information about the order request from the Blockchain.

During the making of a new order, there can be a problem with information that led to another important decision. If nothing is made, it is possible to have two order requests for the same product come approximately at the same time. One of these order requests is picked first and the Aggregator Smart Contract decides which stocks to use for that order and emits the event with those orders. The Event Handler catches this event and sends it to the Web Server to be processed. After that, the Web Server updates the Stocks CID and the Orders CID, for the product of the order, in the Marketplace Smart Contract after updating the IPFS with the new information. Before this information would be stored in the IPFS and the Marketplace Smart Contract the Web Server would process the other order request. This request then arrives at the Aggregator Web Server that receives the request to fulfill that new order with stocks that are no longer available. To solve this problem the Web Server implements a locking mechanism. For each supported product exists a lock. Whenever a transaction begins the Web Server acquires the product lock and after that, the lock is released. In functionalities that require two blockchain transactions, such as the case of making a new order, where one transaction requests the order and the other updates the IPFS and the Marketplace Smart Contract CIDs, the first transaction acquires the lock and the second one releases it. This way we can guarantee that there is only one functionality for each product happening at any given time.

In this work, we also assume the pre-existence of a Decentralized Autonomous Organization (DAO). A DAO is an organization composed of several individuals, in this case farmers, without a single leader. Instead, they need to vote for a change to happen. This assumption was made in order to prevent larger farmers from taking over the system and start fulfilling the orders all by themselves. The system only allows members of the DAO to interact with it. For simplification reasons and because it was not the focus of this work, the DAO is implemented as a simple list.

3.2 Farmer Interactions

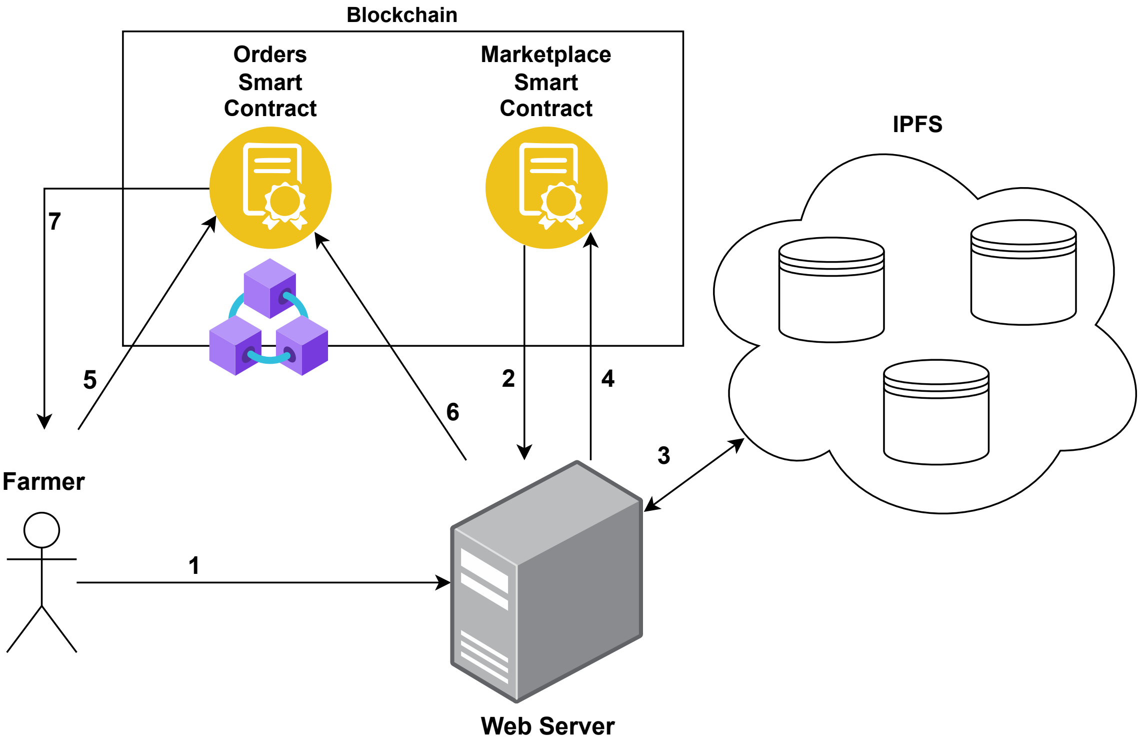

Figure 3.1: Farmer Interactions

Figure 3.1 shows the different interactions that a farmer can trigger in the system. In interaction 1, farmers provide the Web Server with information on the expected harvest period, the type of product, the quality of the product, the quantity of the product, the location of the farm, the price that he pretends to get, a threshold for the price he pretends to get and their Ethereum public address. The Web Server retrieves the current CID for the product specified by the farmer from the Marketplace Smart Contract, interaction 2. The Web Server retrieves it from the IPFS, appends the new information to the existing one, and stores it back in the IPFS, interaction 3, retrieving the new product stock CID in the process. After that, the Web Server updates the Marketplace Smart Contract with the new CID, interaction 4. Once an order with the stocks of a farmer has been accepted, the farmer is notified of how much he needs to transfer to the Orders Smart Contract. Farmers are then required to transfer a deposit, a percentage of the amount they will receive from that order, interaction 5. In this case, this deposit is set to 10% of the amount the buyer pays for that farmer's stock. The system does this to guarantee that the farmers comply with its obligations. After that, farmers send their products to the buyer. This part, which can include the tracking of shipments, is considered out of the scope of this work, but, according to the literature review, a lot of the work done with blockchain in the agricultural sector was done for traceability purposes. When a buyer informs the Web Server that the farmer shipment has arrived, the Web Server demands the Orders Smart Contract to pay the farmer and returns part of the deposit made by the farmer, interactions 6 and 7. A small percentage of the deposit is kept in the Orders Smart Contract. This percentage that is kept is used to finance the system. If for some reason there is the need to increase or decrease this percentage there is no need to deploy a new smart contract. The percentage is a parameter that is stored in the smart contract and is changeable. This was done to save on gas consumption. Changing the value of the parameter is cheaper than deploying a new smart contract.

3.3 Buyer Interactions

Figure 3.2 shows the different interactions that a buyer can trigger in the system. Buyers place orders by providing the Web Server, interaction 1, with information about their location, a radius from that location, this defines an area where the buyer wants products to come from and the aggregation is only done with farmers from inside that radius, the date at which they want the crops, the product they desire with the amount wanted and their Ethereum public address. After this, the Web Server interacts with the Marketplace Smart Contract, interaction 2, to fetch the CID of the desired product and gets the product stocks from the IPFS, interaction 3. After

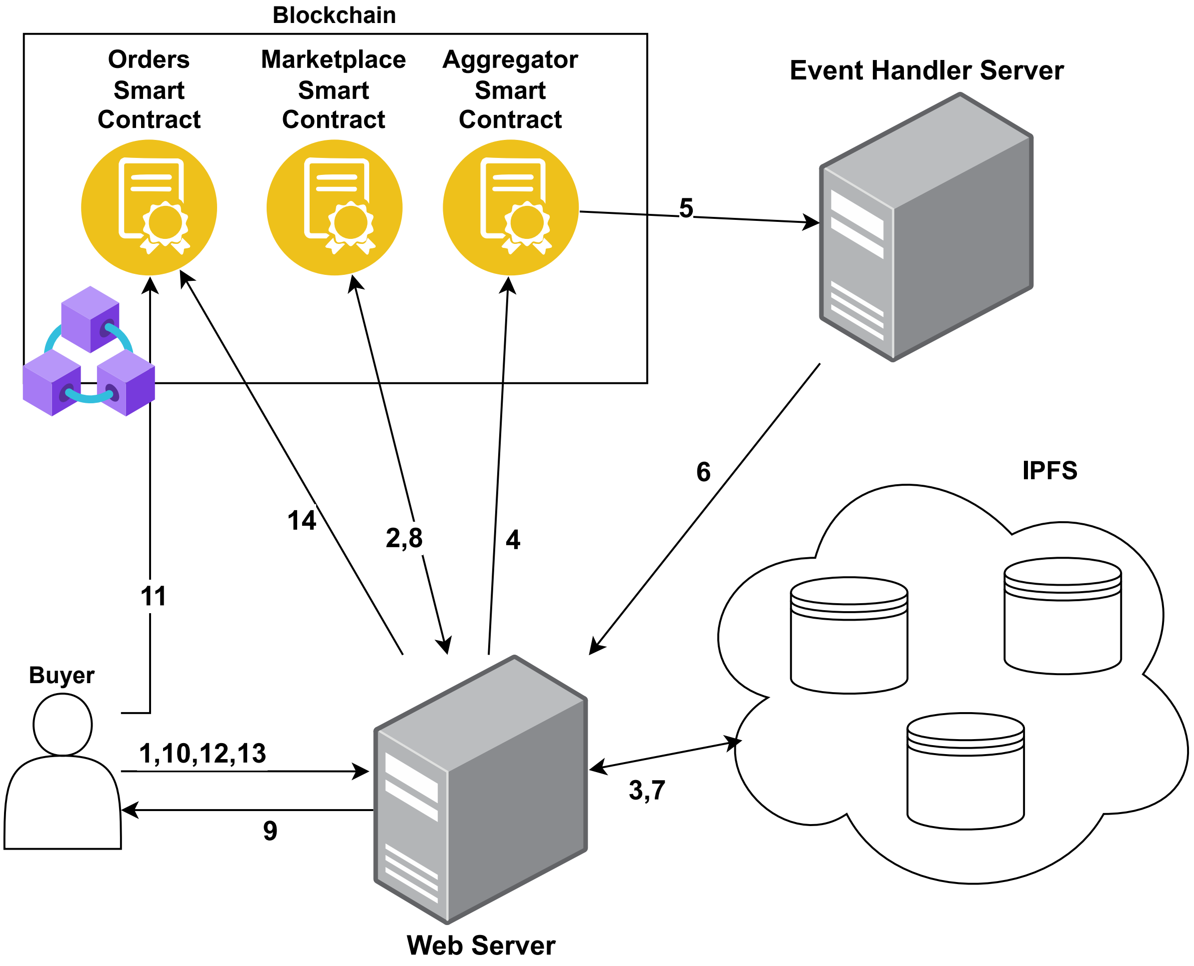

Figure 3.2: Buyer Interactions

that the Web Server requests the Aggregation Smart Contract, interaction 4, to perform the aggregation process; explained in Section 3.4. When the aggregation process finishes, the Aggregation Smart Contract emits an event, interaction 5, that is caught by the Event Handler Server. This event contains information on whether it is possible or not to fulfill the order. The Event Handler Server interacts with the Web Server, interaction 6, which processes the information and saves the new order in the IPFS, the CID in the Marketplace Smart Contract, and then informs the buyer with the proposed offer, interaction 7, 8 and 9 respectively. The buyer can accept the proposed price, decline it, or propose a different price, interaction 10. If the buyer chooses to propose a new price, the system can quickly check if it is possible to fulfill the order at that price based on the price threshold provided by the farmers. The buyer can only propose a new price one time. If the price is possible to fulfil then the order is automatically accepted. If the price is too low and farmers are not able to fulfil it then the order is automatically rejected. This is implemented so that buyers cannot abuse the system to consistently get the lowest possible price. If the number of new proposes was not capped, the buyer could perform something like a binary search, where he proposes a new price every time, checking if the order is possible to fulfill at that price or not, and always getting the lowest possible price that the farmers picked for that order are willing to also accept. When the buyer accepts the order the server notifies all the farmers and the buyer. The buyer then needs to transfer the total amount to the smart contract, interaction 11. The farmer's obligations at this stage are explained in Section 3.4. Shipments arrive individually to the buyer, one from each farmer, which then interacts with the system, interaction 12, to update the status of a specific farmer shipment. This involves reading the CID from the Marketplace Smart Contract, retrieving the information from the IPFS, storing the new one in the IPFS, and storing the new CID in the Marketplace Smart Contract. When the buyer notifies that a shipment arrives, interaction 13, the Web Server requests the Orders Smart Contract, interaction 14, to pay the farmer and to return the down payment made by the farmer. All these three interactions are done through the Web Server, which interacts with the Orders Smart Contract to automate the payment and with the IPFS and Marketplace Smart Contract to update the state of each farmer's stock. The stock of each farmer has several states: available, negotiating, shipping, and used. When a farmer updates his stock, it is stored in the IPFS as available. After the buyer places an order and before accepting it, the stock is in the negotiating state. When the buyer accepts an order, the state changes to shipping. If the order is rejected the stock comes back to the available state. The stock is kept in this state until the buyer says that a shipment has arrived, at which point the state is updated to ”used”.

3.4 Aggregation Process

The Aggregation Process is the core component of the system. It is where most of the project objectives are achieved. The aggregation of farmers is done by the following variables/properties:

- Expected harvesting period provided by the farmers - Only stocks that have a harvesting period that contains the date provided by the buyer are considered. This is done for obvious reasons. Only these Stocks are available at the time the buyer desires them;

- Geographical radius provided by the buyer - The aggregation process only considers farmers that indicate to the system that their farm is within the buyer's desired geographical radius;

- Product Quality provided by the farmers - The buyer indicates to the system both the product and the desired quality. The aggregation process only takes into account crops that the farmers indicated as having that quality;
- Fairness - In this work, fairness has several meanings. First, this work has the objective of not leaving farmers behind, in the sense that it does not want to fulfill orders always with the same farmers. Second, it is not desirable that a large farmer, capable of fulfilling large orders, can overtake the system and start fulfilling all the orders by itself. This is ensured by the existence of the DAO. This fairness factor is a mandatory requirement for all the farmers who want to participate in the supply chain via this system.

The initial idea for the aggregation process was to use some sort of score system. The idea was to attribute some weight to each variable that would enter the aggregation process. For the fairness factor, the idea was to use a variable that would indicate the number of orders that happened that the farmer didn't participate in. The distance from the farmer to the buyer and the quantity available would also be used. With that, every stock would have a score. The process would then continue by sorting the stocks by their score and picking stocks from the top until the quantity desired by the buyer was met. After some discussion, the conclusion was that implementing the aggregation process this way would consume a lot of gas and would be very expensive. The steps of this initial idea would be to filter the stocks, calculate the scores, sort the stocks by score, and finally pick the stocks. With that in mind, the decision was for the aggregation process to pick farmers' stocks randomly, only filtering the stocks first. This simplifies the process in terms of instructions needed, as sorting the Stocks would require a lot of instruction, and thus consumes less gas, which has a lower cost to the system. True randomness is hard to achieve inside a blockchain, which is a deterministic system. In order to implement a function that is as random as possible, the Aggregator Smart Contract picks the next stock to be used based on the previous block randao [34], the block timestamp, and the array of available Stocks. The randao is a pseudorandom generated value [34]. For any given block, the randao is defined by performing a XOR operation between the previous randao value and a randomly generated value by the block validator. In the case of the Blockchain used in this system, this randomly generated number is the validator signature of the current epoch number. If there is the need to pick more than one stock for a certain order using only the previous randao and the block timestamp would give always the same value. That is why the Aggregator Smart Contract also uses the array of available stocks. This array changes every time a new stock is picked. The Aggregator Smart Contract computes the hash of these values together, interprets it as an integer and uses it as a random value.

Algorithm 3.1 shows the pseudocode of the Aggregation Process. This work aims to encourage farmers to participate in the supply chain with the proposed system and not use an intermediary that requires a large commission or consumes part of the profits. To do that, the aggregation process must choose farmers who are probably not the best fit in terms of harvest date or location, but who have not participated in the aggregation process for some time. For the aggregation itself to happen, the Aggregation Smart Contract starts by filtering the stocks by the variables mentioned above and whether they are in the available state or not. After filtering, the stocks to use in the order are randomly picked. After the aggregation process is finished, the Aggregation Smart Contract saves in a Merkle Tree the farmer stocks input array, the buyer's request, and the aggregation decision. A Merkle Tree is a structure that stores hashs and where each node is the hash of the two child nodes. A Merkle Tree was used in this case because it allows us to store the same amount of information via the hashes of the information while using less space. Because space inside a smart contract is something that can be expensive, using a Merkle Tree allows for less gas consumption. After this, an event is emitted. The event is caught by the Event Handler Server that sends it for processing to the Web Server. The Web Server appends the new information to the existing one, stores it in the IPFS, and updates its CID in the Marketplace Smart Contract. The Web Server also stores logs of each new farmer stock, buyer request, and order output after the aggregation process has occurred. The reason behind this is that this way the system is fully auditable. A person or company can backtrack from any point and is able to understand the reasons that led to the aggregation decision of a specific order. It is then possible to take conclusions, understand if the fairness process is being in fact fair and how fair, and then the DAO can vote for an adjustment to the aggregation process if the members desire to do so. If, for some reason, it is not possible to trust the server logs, the person or company auditing the system can ask farmers and buyers what their inputs were to the system or use the information that is stored in the IPFS obtained with the Marketplace Smart Contract CIDs.

Algorithm 3.1: Aggregation Process Pseudo code

```
function MakeNewOrder $(S, O)$
Input  : An array S of the product Stocks,
         A structure O with the order request details
$OrderStocks \longleftarrow BeginAggregationProcess(S, O)$
$generateMerkleTree()$
$emitNewOrderEvent()$

function BeginAggregationProcess $(S, O)$
Input  : An array S of the product Stocks,
         A structure O with the order request details
Output: An Empty Array if an Order is not possible or an Array with the Product Stocks
         used in the Order
$Initialize\ Array\ to\ add\ Available\ Stocks$
$totalQuantity \longleftarrow 0$
foreach $s \in S$ do // Filter Stocks
    if $Stock\ s\ meets\ requirements$ then
        $totalQuantity \longleftarrow totalQuantity + s.quantity$
        $Add\ Stock\ s\ to\ Array$
if $totalQuantity < O.quantity$ then // Order Not Possible
    return $Empty\ Stock\ Array$
return $Aggregate(Available\ Stock\ Array)$ // Order Possible

function Aggregate $(A, q)$
Input  : An array S of the available product Stocks,
         The quantity q for the order
Output: An Array with the Product Stocks used in the Order
$Initialize\ Array\ to\ add\ the\ Order\ Stocks$
$quantity \longleftarrow 0$
while $quantity < q$ do
    $s \longleftarrow pickRandomStock()$
    $Add\ s\ to\ Order\ Stocks\ Array$
    $quantity \longleftarrow quantity + s.quantity$
return $Order\ Stocks\ Array$
```

4

Implementation and Evaluation

Contents

4.1 Implementation Details

The system is developed using the Ethereum Network. Ethereum is a public blockchain, which means that anyone can interact with or join the network at any given time. Besides that, Ethereum provides us with a Turing-complete language, Solidity. According to the literature review done, the Ethereum blockchain is widely used for systems that implement some sort of blockchain component. Solidity is also a language that has great documentation available online. Initial development started using Remix Integrated Development Environment (IDE). Remix is a Solidity IDE used for Web3 Development that allows developers to test their Smart

Contracts without the need to first deploy them to the Ethereum Main Network or a Test Network. It deploys Smart Contracts locally and provides a useful interface that the developer can use to make function calls, check function outputs and emitted events as well as debugging features that the Ethereum Main Network or a Test Network don't provide. When that stopped being sufficient and there was the need to put the Web Server and the Event Handler Server interacting with the Smart Contracts, the deployment of the smart contracts was made to the Ethereum Sepolia Test Network, which, at the time of implementing the system and writing this document, is the recommended testing network by Ethereum. The deployment was also made using Remix IDE since it is also able to connect to a Network through a wallet. For that purpose, MetaMask was used. The Web Server is implemented in Python using the Flask framework to deploy an API. The Event Handler Server is also implemented in Python. To interact with the Web Server, and because it was not the main focus of this work, a simple Python console application was made.

4.2 Methodology

Based on similar works [31], [32], [33] the system is evaluated based on qualitative and quantitative methods. Regarding the quantitative analysis, the evaluation of the system is done in terms of gas fee consumption for each transaction and functionality as well as the average execution time. Regarding qualitative analysis, the system is evaluated by the security properties it provides to all the stakeholders. In [33] the authors evaluate their system concerning confidentiality, data integrity, availability, authorization, non-repudiation, and vulnerability to cyberattacks. This work is evaluated for the same properties. The randomness in the aggregation process also is evaluated since it provides an important objective of this work, the fairness factor.

During the implementation of this work, we contacted several cooperatives and made contact with a farmer who works directly with a cooperative to try to obtain a real database to test the system on. Unfortunately, all the responses we received were negative and it was not possible to get a real database with information on farmer production and/or orders. To create a database that was the most realistic, we searched for what is considered a small farmer by the European Union in terms of farm hectares and the average production in kg per hectare in Portugal. To generate the order requests, we randomly generated orders that would request 1.5 to 5 times the average production of the farmers from the generated farmers' stock database. This is done to guarantee that during our testing the aggregation process would be triggered and would aggregate multiple farmers some of the times. All of this was generated for a specific product. For location, we considered only randomly picked locations in Portugal. The quality field on each entry in the database was also picked randomly and followed the classification standards given by the United Nations Economic Commission for Europe [35]. All the dates were also picked randomly and between two dates that were 90 days apart. We first tried with a completely random date, without any restriction, but initial tests showed that it would be almost impossible to fulfill an order request with the chosen database size.

The system is tested in the Sepolia Test Network. When using a Test Network the amount of gas consumed is the same but the final price of one transaction or to deploy a smart contract is different because of the price that is paid per gas. For that reason, the results consider an Ethereum price of 1534.725 EUR, collected at 12:45:59 on 16 September 2023, given by the Coinbase API. When testing all functionalities, we also recorded the Proposed Gas Price on the Main Ethereum Network, given by the Etherscan API. During the testing, the average Gas Price given by the API was 9 GWEI, or 0.000000009 ETH, and that is the value used to calculate the final price presented on Tables 4.2, 4.3 and 4.1. Tables also show the average execution time for each transaction and functionality. Although this value is presented keep in mind that this is for reference and that execution times on the Main Ethereum Network can be different and varies a lot depending on the gas strategy used.

The plan was to test the smart contract function that allows one to make a new order using an input that contained 1000 stocks. With initial testing that was concluded to be impossible because the estimated consumed gas would exceed the Network's block size limit of 30000000 (30 million) gas. To be able to present a result for this functionality the function was tested with an input of 100 stocks instead of 1000.

4.3 Quantitative Analysis

In this section, we present the evaluation results of the system in terms of quantitative analysis. We present the cost of deploying smart contracts and the cost of transactions and functionalities. In this section, we also discuss the randomness of the Aggregation Process that we use to pick the stocks to be used in the order.

4.3.1 Smart Contracts

Table 4.1 shows the cost of deploying the Smart Contract that composes the system. The Final Price is calculated with the conditions referenced in Section 4.2. The cheapest Smart Contract to deploy is the Orders Smart Contract with a cost of 9.64€. Next is the Marketplace Smart

Contract with a cost of 11.53€. The most expensive, and also most complex, smart contract to deploy is the Aggregator Smart Contract with a cost of 25.67€. This higher cost is because this smart contract has more code compared to the other two and also more complex code. While the main purposes of the other smart contracts are to store strings or to automate payments, the aggregator smart contract executes a lot more business logic, such as filtering the data it receives and picking random stocks to be used.

Table 4.1: Smart Contract Deployment

Smart Contract	Gas Used	Price (EUR)
Marketplace	835070	11.534425
Aggregator	1858694	25.673257
Orders	697650	9.636308

4.3.2 System Functionalities

Any transaction in the Ethereum network must pay gas fees. A call to a Smart Contract Function is considered a transaction if there is any sort of write in the blockchain. In this work, every buyer or farmer interaction can be made only by calling four different functions. The results for each of these transactions can be seen in Table 4.2. In the Marketplace Smart Contract two functions are called, one to update a product stock CID and another to update both a product stock CID and the product orders CID. The function to update a product stock CID is called when a farmer informs that he has a new stock, when a farmer informs that he has sent the money to the Orders Smart Contract, and when a farmer's shipment has arrived at the buyer. The function that updates both the product stock and orders CID's is called when a buyer accepts, rejects, or tries to negotiate an order, when the buyer informs that he has paid the order, and after every order transaction. In the Aggregator Smart Contract, the only function called by the Web Server is to try to make an order. This function triggers an event that is caught by the Event Handler Server and calls the function that updates both the product stock and orders CID's on the Marketplace Smart Contract. Lastly, when a buyer informs that a shipment has arrived, he triggers a call to the Orders Smart Contract to pay the respective farmer. After these transactions are processed, the Web Server also calls the function to update the stock CID on the Marketplace Smart Contract as described above.

Table 4.2 shows the total Gas Used and the final price and the average execution time of each possible smart contract transaction. Two of the transactions, update stock CID and pay address, had a constant Gas amount consumed because they either store a string always with the same length or make a transfer, respectively. The transaction that updates the stock CID

Table 4.2: Results Table by Transaction

Transaction		Gas Used	Price (EUR)	Avg. Execution Time (s)
Update Stock CID		41703	0.576024	18.18
Update Order CID and Stock CID		53980 or 59580	0.745600 or 0.823075	13.50
Pay Address		46173	0.637766	14.67
Make New Order	Impossible to fulfil Order (Avg.):	1815165	25.072012	18.08
	Pick New Stock (Avg.):	456112	6.300058	

costs the system 0.58€ and has an average execution time of 18 seconds. The transaction that pays to the farmer costs the system 0.64€ and has an average execution time of 15 seconds. The transaction that updates the Order CID and the Stock CID has two possible Gas consumption because only one of the CIDs may be updated. For example, when making an order it is possible that after filtering the stocks we conclude that there is not enough quantity to fulfill the order. In this case, the system records that there was a new order and that it was impossible to fulfill. The system writes this change to the IPFS, getting a new CID for the orders, and the Stocks CID remains the same because no Stock needs the state updated. When performing the transaction with the Marketplace Smart Contract, only one of the CID's is changed in this situation. In the case that only the Orders CID is updated the transaction costs the system 0.74€. In case both CID are updated the transaction costs the system 0.82€. This transaction had an average execution time of 13.5 seconds. The transaction to make a new order doesn't have a constant gas consumed as the others do. This is because the transaction has a step in which one stock is picked randomly until the desired quantity is met. This can lead to a new stock being picked any number of times. Besides that, the transaction begins by filtering the stocks. In this filtering process, there is an if statement with 5 conditions. It checks if the stock is available, if it has the desired quality, if it is within the desired distance to the buyer, and if the desired date is compatible with the stock harvesting date. The smart contract checks these conditions one by one and stops checking the stock conditions when it finds a false condition. This means that even the filtering process does not consume a constant amount of gas for the same input size. With that in mind, if an order is impossible to fulfill, meaning that the input is filtered and the available quantity is less than the one desired by the buyer, the transaction has a cost of 25.07€. This includes not just the filtering but also the emission of the event and the creation of the Merkle Tree for audit purposes. Besides that, an order can be fulfilled with more than one farmer's stock. Results have shown that for each new Stock that is picked, the cost of the transaction increases by 6.30€. This is expensive because it generates a random number based on the hash of the available stocks, which in turn can be a large array. In this case, the tests were made with an array that had a maximum length of 100. This transaction has an average execution time of 18 seconds.

Table 4.3: Results Table by Functionality

Functionality	Transaction(s) triggered	Total Gas Used	Total Price (EUR)	Avg. Execution Time (s)
Update Stock	Update Stock CID	41703	0.576024	18.18
Money Sent (Farmer)	Update Stock CID	41703	0.576024	18.18
Accept/Reject/Negotiate Order	Update Order CID and Stock CID	59580	0.823075	13.50
Money Sent (Buyer)	Update Order CID and Stock CID	59580	0.823075	13.50
Inform Shipment has Arrived	Pay Address + Update Stock CID	87876	1.213789	32.85
Make Order	Make New Order + Update Order CID and Stock CID	Starts at 1869145	Starts at 25.82	31.58

The results in Table 4.3 are already presented considering these situations. Taking into account the conditions described in Section 4.2 the total cost of the system to update a farmer's stock or to inform that system that a farmer has sent the money and the shipment is 0.58€. The cost of accepting, rejecting, or negotiating an order is 0.82 €. When the buyer informs the system that he has sent the money to the Orders Smart Contract, the Web Server updates the Order informing that it is paid for, and calculates the amount that each farmer picked is going to receive and the amount for the deposit to notify them. These changes are recorded to the IPFS which in turn returns a new CID for both the Orders and the Stocks. The cost of this transaction is 0.82€ because both CID are updated. Informing the system that a farmer's shipment has arrived has a cost of 1.21 €. This happens because this functionality involves two different transactions. First, the Orders Smart Contract transfers the money to the farmer, and only if and when this transaction is successful the transaction to update the Stocks CID happens. Making a new Order does not have a fixed cost, as seen before. In case it is impossible to fulfill the request then the total gas consumed is the one correspondent with that transaction, 1815165 gas, plus the gas consumed in the transaction where on updating the Order CID and the Stock CID only the Order one is updated, 53980 gas. This brings the cost to an average of 1869145 gas consumed, or 25.82€. In the case where it is possible to fulfill the order, the base cost increases to 1874754 gas consumed, or 25.90€, plus an average of 6.30€ for each stock picked. This small increase is due to the second transaction that occurs to update the Order CID and Stock CID, updating both CID. All of these costs are financed by the percentage that is retained when the farmer's payment and return of deposit is made. This percentage can be changed without the need to deploy a new smart contract.

4.3.3 Aggregation Process Randomness

To test how random our stock pick method was we performed a different test. For this test, we generate a new database consisting of 100 stocks of a product and 10 order requests. We generated these stocks and orders in a way that it would always be possible to fulfill the order, meaning that the quality was always considered the same, the order dates were always in the middle of every stock harvesting period, and the radius provided by the buyer was always set so that it would always be higher than the maximum possible distance. We then proceeded by performing the 10 order requests. This process was repeated several times until at least 1000 random stocks were picked. This way we could have a good sense of how random the aggregation process actually was. In total, 1145 stocks were randomly picked by the Aggregator Smart Contract. With this test, we expect to see that no stock has a higher probability of being picked than the others. In Figure 4.1 we can see how many times each Stock was picked in this process. In this figure, we can observe that every stock is picked and no Stock was picked several times which is much higher than the others. In order to see how random it is, and subsequently how fair the system is, a lot more stocks would have to be randomly picked. Nonetheless, this way we can have a good sense of how random and fair the Aggregation process is.

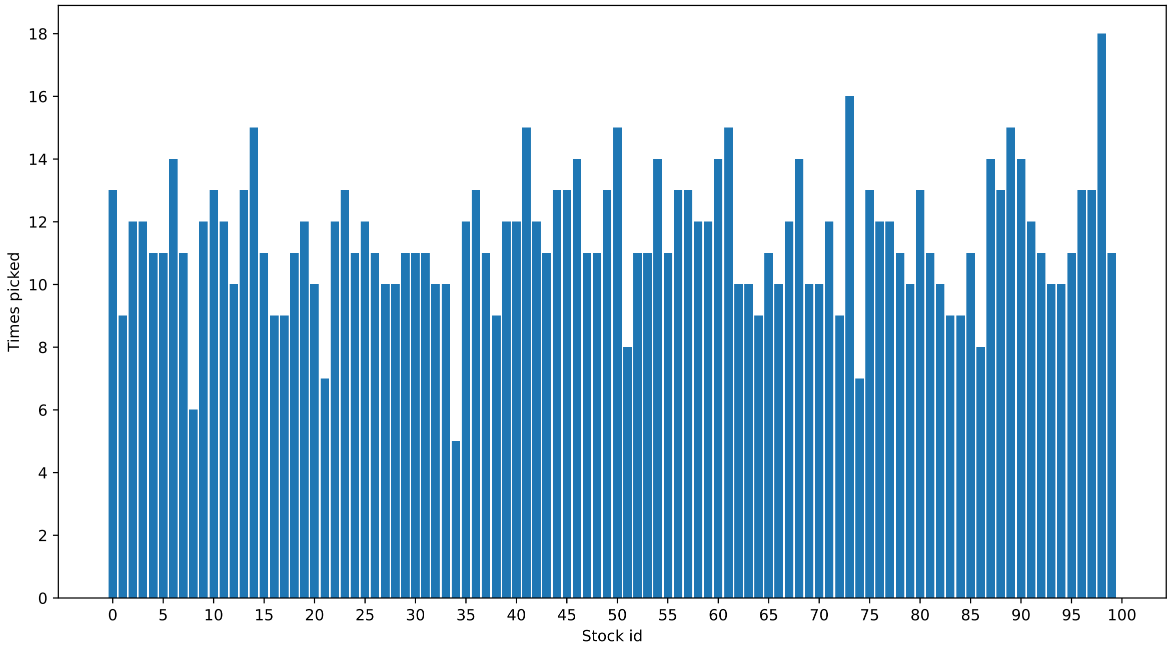

Figure 4.1: Histogram of the Randomness in the Aggregation Process

4.4 Qualitative Analysis

In this Section, we will discuss the security properties that the system gives to farmers and buyers. It will discuss not just how can these properties affect farmers and buyers but also how can the properties be achieved by either properties given by blockchain technologies and implementation details.

Regarding confidentiality, we can say that only authorized users can see information present in the blockchain. Although farmers and buyers share their location with a blockchain address and that information is stored in the IPFS, only the actors with the right CID can access the information. For this to be enforced there is a restriction that only a certain address can interact with each smart contract function. In this case, the address is the one that deploys the smart contracts which, in turn, is also the one used by the Web Server to be able to fulfill farmers' and buyers' requests. The data integrity property is given to us by blockchain technologies. For any piece of information to be modified there is the need to make a new transaction that is recorded in the blockchain. With that, there is also the capacity to track changes and see when they were made. Regarding the IPFS while there is some node that has the information with the most recent CID data is maintained. To ensure that this happens, the farmers DAO can deploy their own node. Regarding availability, any information that is on the blockchain or in the IPFS will always be available as long as there still is at least one node, or validator, capable of performing transactions, in the case of the blockchain, or one IPFS node that is still storing the information and is possible to communicate with it. Two components of this system are more centralized, the Web Server and the Event Handler Server, and to make sure that the system is always available, in terms of accessing information or performing transactions, there needs to be some redundancy. Regarding authorization, only farmers who are present in the DAO can interact with the system. Anyone can be a buyer as the system is, but that can be changed by the farmers DAO if they vote to do so. Regarding non-repudiation, this is a little harder to guarantee if we talk from the perspective of making requests. All requests have to pass through the Web Server that can be compromised or receive requests from someone who is impersonating another person. Although this is true there is still non-repudiation in the system. Every transaction is recorded in the blockchain and farmers cannot negate that they have received the money, for example. Lastly, regarding the vulnerability to cyberattacks, we can say that every transaction inside the blockchain is recorded and by using blockchain we can ensure that once a transaction is recorded it is almost impossible to alter it. A transaction is included in a block and the block contains the hash of the previous block. This makes it very hard to change a transaction that already is present in a block.

5

Conclusion

Contents

Small and medium-sized farmers have difficulty trying to integrate the supply chain. Current solutions require the use of an intermediary that eats part of the profits that can be diverted to the farmers. This work implements a system that automatically pairs multiple farmers to fulfill larger orders, thus integrating the farmers into the supply chain. Currently, no existing system is able to do that and be fair at the same time. The system described in this document pairs farmers randomly until the buyers' request is met. In addition to that, the system described in this document also reduces the inefficiency of the negotiation process by automatically taking care of it. In this system, payments are also automatic and efficient. The system is important for farmers because it allows them to stop being price takers and allows them to define the price they feel is fair for them. It encourages farmers to participate in the supply chain through the system because the aggregation process is made in a way that is certain that sooner or later the farmer will participate in an Order. For buyers, this system allows them to communicate with only one entity instead of multiple to fulfill their desired quantities. Using smart contracts and blockchain technologies to implement such a system provides farmers with a platform that brings trust to the whole process without the need to know each other.

5.1 Future Work and System Limitations

An important thing that was left out of the scope of this work is to find a way to verify that what farmers and buyers say to the system is true. As it is, there is the possibility that a malicious buyer makes several order requests without the intention of actually accepting one. Doing so would have the potential to spend all the funds that the system has. Because of that, there must be a way to verify that buyers want the product. This can be done by, for example, having the farmers DAO vote for a buyer to be able to also use the system. It is also important to find a way to verify that farmers also have the product that they inform the system they have. Achieving this can be done through the integration of this work system with a traceability system that requires photos to be taken, for example. This way it would not just be possible to verify that farmers are telling the truth but also trace the shipments during the shipment process. This work is not without limitations. One limitation is that, as the system is, the stocks are not picked in a truly random way. Unfortunately, true randomness inside a blockchain is something very hard to achieve, and the results of this work ended up showing that. Some solutions try to create true and verifiable randomness, such as Chainlink Verifiable Random Function (VRF), but this solution comes with the drawback that it would cost an extra 0.25 LINK for each request, increasing the price of the transaction, and is not synchronous. Chainlink VRF generates a random number, or more, by hashing several parameters together and waits for the transaction that generates it to be several blocks deep in the blockchain to make a callback to the smart contract that requested the random number(s). This would also increase the time to fulfill a new order request. A clear limitation is also the limited size of stocks that the aggregation process can take as an input. In the future one possible direction is to explore Blockchain layer two solutions that could have a higher block gas limit.

Bibliography

[1] M. K. Lim, Y. Li, C. Wang, and M.-L. Tseng, “A literature review of blockchain technology applications in supply chains: A comprehensive analysis of themes, methodologies and industries,” *Computers & Industrial Engineering*, vol. 154, p. 107133, Apr. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360835221000371

[2] A. Kamilaris, A. Fonts, and F. X. Prenafeta-Bold´, “The rise of blockchain technology in agriculture and food supply chains,” *Trends in Food Science & Technology*, vol. 91, pp. 640–652, Sep. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0924224418303686

[3] G. Berti and C. Mulligan, “Competitiveness of Small Farms and Innovative Food Supply Chains: The Role of Food Hubs in Creating Sustainable Regional and Local Food Systems,” *Sustainability*, vol. 8, no. 7, p. 616, Jul. 2016. [Online]. Available: http://www.mdpi.com/2071-1050/8/7/616

[4] A. Malak-Rawlikowska, E. Majewski, A. Was, S. O. Borgen, P. Csillag, M. Donati, R. Freeman, V. Hoàng, J.-L. Lecoeur, M. C. Mancini, A. Nguyen, M. Saïdi, B. Tocco, [U+FFFD] Török, M. Veneziani, G. Vittersø, and P. Wavresky, “Measuring the Economic, Environmental, and Social Sustainability of Short Food Supply Chains,” *Sustainability*, vol. 11, no. 15, p. 4004, Jul. 2019. [Online]. Available: https://www.mdpi.com/2071-1050/11/15/4004

[5] J. J. Sikorski, J. Haughton, and M. Kraft, “Blockchain technology in the chemical industry: Machine-to-machine electricity market,” *Applied Energy*, vol. 195, pp. 234–246, Jun. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306261917302672

[6] M. Di Pierro, “What Is the Blockchain?” *Computing in Science & Engineering*, vol. 19, no. 5, pp. 92–95, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8024092/

[7] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran,

"An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, Apr. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X19316280

[8] S. Köhler and M. Pizzol, "Technology assessment of blockchain-based technologies in the food supply chain," *Journal of Cleaner Production*, vol. 269, p. 122193, Oct. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S095965262032240X

[9] S. E. Chang and Y. Chen, "When Blockchain Meets Supply Chain: A Systematic Literature Review on Current Development and Potential Applications," *IEEE Access*, vol. 8, pp. 62 478–62 494, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9047881/

[10] P. Gonczol, P. Katsikouli, L. Herskind, and N. Dragoni, "Blockchain Implementations and Use Cases for Supply Chains-A Survey," *IEEE Access*, vol. 8, pp. 11 856–11 871, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8952728/

[11] W. Lin, X. Huang, H. Fang, V. Wang, Y. Hua, J. Wang, H. Yin, D. Yi, and L. Yau, "Blockchain Technology in Current Agricultural Systems: From Techniques to Applications," *IEEE Access*, vol. 8, pp. 143 920–143 937, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9159588/

[12] M. Pournader, Y. Shi, S. Seuring, and S. L. Koh, "Blockchain applications in supply chains, transport and logistics: a systematic review of the literature," *International Journal of Production Research*, vol. 58, no. 7, pp. 2063–2081, Apr. 2020. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/00207543.2019.1650976

[13] S. E. Chang, Y.-C. Chen, and M.-F. Lu, "Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process," *Technological Forecasting and Social Change*, vol. 144, pp. 1–11, Jul. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0040162518305547

[14] D. Kos and S. Kloppenburg, "Digital technologies, hyper-transparency and smallholder farmer inclusion in global value chains," *Current Opinion in Environmental Sustainability*, vol. 41, pp. 56–63, Dec. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877343519300557

[15] S. Stranieri, F. Riccardi, M. P. Meuwissen, and C. Soregaroli, "Exploring the impact of blockchain on the performance of agri-food supply chains," *Food Control*, vol. 119, p. 107495, Jan. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0956713520304114

[16] P. Burgess, F. Sunmola, and S. Wertheim-Heck, “Blockchain Enabled Quality Management in Short Food Supply Chains,” *Procedia Computer Science*, vol. 200, pp. 904–913, 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050922002976

[17] Y. Kayikci, N. Subramanian, M. Dora, and M. S. Bhatia, “Food supply chain in the era of Industry 4.0: blockchain technology implementation opportunities and impediments from the perspective of people, process, performance, and technology,” *Production Planning & Control*, vol. 33, no. 2-3, pp. 301–321, Feb. 2022. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/09537287.2020.1810757

[18] H. Hasan, E. AlHadhrami, A. AlDhaheri, K. Salah, and R. Jayaraman, “Smart contract-based approach for efficient shipment management,” *Computers & Industrial Engineering*, vol. 136, pp. 149–159, Oct. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360835219304140

[19] S. Wang, D. Li, Y. Zhang, and J. Chen, “Smart Contract-Based Product Traceability System in the Supply Chain Scenario,” *IEEE Access*, vol. 7, pp. 115 122–115 133, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8804170/

[20] L. Hang, I. Ullah, and D.-H. Kim, “A secure fish farm platform based on blockchain for agriculture data integrity,” *Computers and Electronics in Agriculture*, vol. 170, p. 105251, Mar. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016816991932006X

[21] L. Wang, L. Xu, Z. Zheng, S. Liu, X. Li, L. Cao, J. Li, and C. Sun, “Smart Contract-Based Agricultural Food Supply Chain Traceability,” *IEEE Access*, vol. 9, pp. 9296–9307, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9317793/

[22] P. De Giovanni, “Blockchain and smart contracts in supply chain management: A game theoretic model,” *International Journal of Production Economics*, vol. 228, p. 107855, Oct. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925527320302188

[23] F. R. Hermiatin, Y. Handayati, T. Perdana, and D. Wardhana, “Creating Food Value Chain Transformations through Regional Food Hubs: A Review Article,” *Sustainability*, vol. 14, no. 13, p. 8196, Jul. 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/13/8196

[24] F. Sgroi and G. Marino, “Environmental and digital innovation in food: The role of digital food hubs in the creation of sustainable local agri-food systems,” *Science*

of The Total Environment, vol. 810, p. 152257, Mar. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0048969721073332

[25] D. A. Cleveland, N. M. Müller, A. C. Tranovich, D. N. Mazaroli, and K. Hinson, "Local food hubs for alternative food systems: A case study from Santa Barbara County, California," *Journal of Rural Studies*, vol. 35, pp. 26–36, Jul. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0743016714000436

[26] A. Marusak, N. Sadeghiamirshahidi, C. C. Krejci, A. Mittal, S. Beckwith, J. Cantu, M. Morris, and J. Grimm, "Resilient regional food supply chains and rethinking the way forward: Key takeaways from the COVID-19 pandemic," *Agricultural Systems*, vol. 190, p. 103101, May 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0308521X21000548

[27] A. Mittal, C. Krejci, and T. Craven, "Logistics Best Practices for Regional Food Systems: A Review," *Sustainability*, vol. 10, no. 2, p. 168, Jan. 2018. [Online]. Available: http://www.mdpi.com/2071-1050/10/1/168

[28] M. Ioannis, M. George, and M. Socrates, "A Community-Based Agro-Food Hub Model for Sustainable Farming," *Sustainability*, vol. 11, no. 4, p. 1017, Feb. 2019. [Online]. Available: http://www.mdpi.com/2071-1050/11/4/1017

[29] A. Mittal and C. C. Krejci, "A hybrid simulation modeling framework for regional food hubs," *Journal of Simulation*, vol. 13, no. 1, pp. 28–43, Jan. 2019. [Online]. Available: https://www.tandfonline.com/doi/full/10.1057/s41273-017-0063-z

[30] M. Kumarathunga, R. Calheiros, and A. Ginige, "Towards Trust Enabled Commodity Market for Farmers with Blockchain Smart Contracts," in *Proceedings of the 2020 Asia Service Sciences and Software Engineering Conference*. Nagoya Japan: ACM, May 2020, pp. 75–82. [Online]. Available: https://dl.acm.org/doi/10.1145/3399871.3399891

[31] C.-H. Liao, H.-E. Lin, and S.-M. Yuan, "Blockchain-Enabled Integrated Market Platform for Contract Production," *IEEE Access*, vol. 8, pp. 211 007–211 027, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9265212/

[32] G. Leduc, S. Kubler, and J.-P. Georges, "Innovative blockchain-based farming marketplace and smart contract performance evaluation," *Journal of Cleaner Production*, vol. 306, p. 127055, Jul. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0959652621012749

[33] I. A. Omar, R. Jayaraman, M. S. Debe, K. Salah, I. Yaqoob, and M. Omar, “Automating Procurement Contracts in the Healthcare Supply Chain Using Blockchain Smart Contracts,” *IEEE Access*, vol. 9, pp. 37 397–37 409, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9363880/

[34] M. Kalinin and D. Ryan, “EIP-4399: Supplant DIFFICULTY opcode with PREVRANDAO,” *Ethereum Improvement Proposals*, no. 4399, Oct. 2021, [Online; accessed 23-October-2023]. [Online]. Available: https://eips.ethereum.org/EIPS/eip-4399

[35] U. N. E. C. for Europe, “Fresh Fruit and Vegetables - Standards,” 2022, [Online; accessed 23-October-2023]. [Online]. Available: https://unece.org/trade/wp7/FFV-Standards