Instituto Politécnico
de Viana do Castelo

FRUIT AND VEGETABLE TRACEABILITY

From Farm to Fork

Ricardo Jorge Lopes Morais

Escola Superior de Tecnologia e Gestão

Instituto Politécnico
de Viana do Castelo

Ricardo Jorge Lopes Morais

FRUIT AND VEGETABLE TRACEABILITY

Name of the Master's course

Master in Informatics Engineering

Work supervised by

Prof. Doutora Maria Estrela Ferreira Cruz

Prof. Doutor António Miguel Rosado da Cruz

November 2023

Abstract

Consumers are increasingly aware of problems related to food safety and fraud and environmental and ecological impacts due to food production. Besides the quality of the products, there is currently also a growing concern about the origin of the products and the path they take to reach the final consumers. Consumers demand transparency across the entire value chain of the products they consume, leading to the need to design and develop a traceability platform for these products. The platform must collect information at all stages of the chain, store it in a system, and share it with all actors in the value chain. This information must be available to all participants in the value chain, including farmers, health authorities, government industries, sellers, retailers, final consumers, and any other actor involved in the value chain. This project describes the planning and implementation of a platform that allows us to trace back and forth fruits and vegetables. This platform would be decentralized by using a blockchain-based approach. The proposed platform enables the visualization of fruit and vegetable value chain information.

Keywords: Traceability. Sustainability. Fruit and Vegetable Value Chain. Blockchain. Food Safety. Decentralized Application.

Resumo

Os consumidores estão cada vez mais conscientes dos problemas relacionados com a segurança e fraude alimentar e os impactos ambientais e ecológicos devidos à produção de alimentos. Além da qualidade dos produtos, atualmente, também existe uma preocupação crescente com a origem dos produtos e o caminho que percorrem até chegar ao consumidor final. Os consumidores exigem transparência em toda a cadeia de valor dos produtos que consomem, o que leva à necessidade de conceber e desenvolver uma plataforma de rastreabilidade para estes produtos. A plataforma deve recolher informações em todas as fases da cadeia, armazená-las num sistema e partilhá-las com todos os intervenientes na cadeia de valor. Esta informação deve estar disponível para todos os participantes na cadeia de valor, incluindo agricultores, autoridades de saúde, indústrias governamentais, vendedores, retalhistas, consumidores finais e qualquer outro interveniente envolvido na cadeia de valor. Este projeto descreve o planeamento e implementação de uma plataforma que permite rastrear frutas e vegetais. Esta plataforma seria descentralizada e usaria uma abordagem baseada em blockchain. A plataforma proposta permite a visualização de informações da cadeia de abastecimento de frutas e vegetais.

Palavras-chave: Rastreabilidade. Sustentabilidade. Cadeia de Valor de Frutas e Vegetais. Blockchain. Segurança Alimentar. Aplicação Descentralizada.

Acknowledgments

First, I would like to express my sincere gratitude to my thesis supervisors, Prof. Doutora Maria Estrela Ferreira Cruz and Prof. Doutor António Miguel Rosado da Cruz, for their unwavering patience, guidance, and support.

During this project, I had the pleasure of meeting various people in different areas, and I wanted to thank them for sharing their time with me during the interviews and study tests.

Now, I want to thank my parents. Without them, I couldn't have taken this journey. Also, my brother and sister for their neverending support. Next, my grandmother, for her kindness, and finally, to the rest of my family for believing in me.

Ricardo Jorge Lopes Morais

Words were not given to man in order to conceal his thoughts.

José Saramago

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

ABC Activity-Based Costing

AI Artificial Intelligence

API Application Programming Interface

app application

ARIS Architecture of Integrated Information Systems

B+DB Blockchain + Database

BPMN Business Process Model and Notation

BSON Binary JSON

CAGR Compound Annual Growth Rate

CFT Crash Fault-Tolerant

CRUD Create, Read, Update, and Delete

DApp Decentralized Application

DB database

DNA-based Deoxyribonucleic Acid-based

DOM Document Object Model

DSR Design Science Research

EPC Event-driven Process Chain

F2F From Farm To Fork

F&V Fruit and Vegetable

HACCP Hazard Analysis and Critical Control Points

HTTP Hypertext Transfer Protocol

ID identification

IoT Internet of Things

IPFS Interplanetary File System

JS JavaScript

JSON JavaScript Object Notation

JWT JSON Web Token

m minute

M2M Many-to-Many

MERN Stack MongoDB, Express.js, React.js, and Node.js

MVC Model-View-Controller

NFC Near Field Communication

NoSQL not only SQL

ODM Object Data Modeling

PBFT Practical Byzantine Fault Tolerance

QR Code Quick Response Code

REST Representational State Transfer

RFID Radio Frequency Identification

UI User Interface

URL Uniform Resource Locator

UUID Universal Unique Identifier

v version

Chapter 1

Introduction

According to [Verified Market Research, 2022], fresh fruits and vegetables are valued at 143,904.80 USD and estimated to reach 211,073.81 USD, with a Compound Annual Growth Rate (CAGR) of 5.14% from 2021 to 2028. The explanation for this rate is the constant growth in population, the rising focus on health and fitness, and the increasing preference for organic products, which drives the demand for rich, natural, nutrient-dense products, boosting the agricultural sector in producing fruits and vegetables.

Consumers and other actors in the food value chain are now more concerned than ever about food safety, food fraud, the environmental and ecological impact of food production, and animal welfare. This concern is even more significant when working with fresh products such as fruits and vegetables because of the delicate agro-processing steps and the various needs depending on the product. Failure in any of those steps can cause contamination in the respective products and, consequently, lead to disease, possible health risks, or even death. Thus, there is a need for transparency in the fruits and vegetables' treatment from the moment of their harvest to their final sale points.

It's important to highlight the significance of this theme worldwide as several authorities have enacted legislation on food safety. The European Union, in 2002, decreed the first legislation on food safety – Regulation (EC) Nº. 178/2002 [European Union, 2002]. This document, in short, requires that all food/ration companies in and importing into Europe have an effective traceability system. This system must keep records for those companies to identify the origin and destination of the products (raw materials or processed products) they receive and produce, and in the event of contamination, immediately collect the supplied products and warn consumers and the competent authorities. Similar traceability systems requirements can be found in the United States of America (Regulation 21 CFR 820), Japan (Guidelines for the Introduction of Food Traceability Systems), and others.

Although some of these laws are revised and renewed to improve the food safety process, like the Regulation (EU) Nº. 2019/1381 [European Union, 2019], which amends the existing one, reinforcing transparency and other aspects, and the ISO 22005:2007 [ISO, 2016] that provides the standards for creating a system to trace food/ration.

There are still situations such as the Listeriosis outbreak in South Africa in 2017–2018 derived from soiled processed meat, infecting 1060 and killing 216 [Whitworth, 2018]; the E.coli O104:H4 outbreak of 2011 in Germany that derived from contaminated fenugreek, infecting more than 3950 and killing 53 [European Food Safety Authority, 2012]; the 2011 Listeriosis outbreak in the United States from contaminated cantaloupe [Neuman, 2011] and many other outbreaks in the last decade alone.

We believe that a solution can be achieved with the traceability of these products in a From Farm To Fork (F2F) process, i.e., in a process that can identify the origin of products from their harvest until their sale. This process should, therefore, store various types of information in each step of the food value chain. This information should be available to consumers and any actor in the food value chain to ensure the quality, origin, and control of their lots, and it should enable the forward or backward traceability of lots, even if they have originated from several lots. Similarly, this process should aid in preventing food fraud since it should not be possible to change a product's origin and harvest date, helping the consumer to differentiate the freshest fruits and vegetables and, if desired, products whose origins cause fewer environmental impacts. Thus, the objective is to design and implement a platform that permits the traceability of fresh and transformed fruits and vegetables and the storage quality and sustainability indicators in an F2F process.

This project proposes a blockchain-based (i.e., Blockchain + Database (B+DB)) approach for tracking quality and sustainability throughout the Fruit and Vegetable (F&V) value chain. Blockchain is a technology that enables the secure and transparent recording of transactions and tracking of tangible or intangible assets via a shared ledger [IBM, 2023e]. We're using this technology because it allows authenticity, integrity, and immutability in the stored records, reducing food fraud and improving food safety. Its decentralized nature and immutability allow for greater transparency with consumers, bringing more trust between them and organizations.

1.1 Methodology

We're utilizing the Design Science Research (DSR) methodology for this study. DSR is a research method that aims to generate prescriptive knowledge about the design of artifacts to solve specific problems of organizations, giving them the possibility to, directly or indirectly, increase their profits [Cruz and da Cruz, 2020a]. We're using DSR for the reasons identified in [Baskerville et al., 2015], which says that DSR solves problems never solved before, uniquely and innovatively, or solves previously solved problems more efficiently and effectively than the existing ones.

According to [Cruz and da Cruz, 2020a, Hevner et al., 2004], DSR is a process composed of the following main research activities:

- **Problem identification and motivation** - for final consumers, it's necessary to have transparency in how the products in the F&V value chain reached them, i.e., to know all the steps since their harvest. For companies, it allows for greater trust with consumers and better practices in food safety.
- **Definition of the objectives for the solution** - the intention is to develop a traceability platform for the F&V value chain so that any actor in this chain can, at any stage, see lot information by inserting the lot identification (ID) or scanning the Quick Response Code (QR Code).
- **Design and implementation** - a solution using a B+DB methodology and the development of a web and mobile application (app) will be designed and implemented.
- **Demonstration** - in a more advanced state, we will prove that the artifacts can solve the mentioned problems, putting them to work in an F&V traceability simulation system.
- **Evaluation** - the developed platform will undergo different tests (e.g., performance, usability).

- **Communication** - after passing all approval tests, the project's results will be presented and published at a conference.

1.2 Dissertation Structure

The rest of this dissertation is structured as follows: the next chapter presents the basic concepts mentioned throughout our work, the tools to achieve our objectives, and the programming languages used. Chapter 3 presents a background review of the F&V traceability platforms and techniques. Chapter 4 outlines the steps involved in the F2F process and shows the generic fruit and vegetable business process. Chapter 5 covers our proposed solution for the traceability platform, presenting actors, needed data, architecture, and others. Chapter 6 details the implementation of our traceability platform, highlighting the development of the traceability algorithm, security, blockchain, database, and others. Chapter 7 presents some conclusions and draws some lines for future work.

Chapter 2

Concepts, Tools, and Languages

Chapter 2 presents some of the basic concepts that served as the starting point for this project. Also, the tools and programming languages used and how we used them.

2.1 Concepts

This section demonstrates some of the basic concepts mentioned throughout this work.

2.1.1 Blockchain

Blockchain is a decentralized technology that enables the secure and transparent recording of transactions and the tracking of tangible and intangible assets via a shared, immutable ledger [IBM, 2023e].

It operates on a decentralized peer-to-peer network where all participants (nodes) store a copy of the blockchain, ensuring no central authority or a single point of failure exists [Zahed Benisi et al., 2020].

Blockchain groups transactions into blocks and links them to a chain of previous blocks, creating an immutable and transparent record of all transactions. Each block is connected to the previous one through cryptographic hashes, making it almost impossible to tamper with. These blocks usually contain four fields: the main data, which typically includes transaction records; the hash of the previous block; a timestamp; and other information, such as the block's signature and data defined by the user [Lin and Liao, 2017].

The transparency, immutability, and other characteristics of blockchain make it ideal for use in various industries, including supply chain management, cybersecurity, finance, healthcare, banking, and others [Dutta et al., 2020].

Smart Contract

Smart contracts are self-executing digital contracts (computer programs) encoded on a blockchain network. These programs automatically execute the terms of the specific contract when met with the predetermined conditions. In blockchain, they are usually used to automate consensus between all nodes in the network, eliminating the need for intermediaries, thus decreasing transaction costs and increasing efficiency and trust in business processes [Lin and Liao, 2017, IBM, 2023b].

Consensus Mechanism

A consensus mechanism is a technique used by nodes within a blockchain network to agree on the transaction's validity, order, and liveness. These mechanisms ensure that all nodes in the network reach a consensus on the state of the blockchain. There are various consensus mechanisms (e.g., Proof of Work, Proof of Stake, Delegated Proof of Stake, and Proof of Elapsed Time), each with positives and negatives, depending on the desired solution [Lin and Liao, 2017, Wang et al., 2019].

2.1.2 Sustainability

In 1987, [World Commission on Environment and Development, 1987, p. 16] defined sustainability as "meeting the needs of the present without compromising the ability of future generations to meet their own needs."

The definition highlights the need to move towards sustainability but to achieve this, there needs to be stability between the three pillars of sustainability - economic, environmental, and social. So, to achieve sustainability, an industry needs to be profitable throughout, advance social welfare, and not harm the ecosystems [Mollenkamp, 2023].

In the food industry, sustainability refers to measures that aim to reduce economic, environmental, and social impact, increase inclusive, green, and eco-social growth, encourage social responsibility, and provide long-term food security. It acknowledges the correlation between food production, processing, distribution, and consumption with economic, environmental, and social factors [Food and Agriculture Organization of the United Nations, 2018].

2.1.3 Traceability

Porto Editora describes traceability as the "possibility of identifying the origin of a product and reconstructing its path from production to distribution" [Porto Editora, 2010]. Its purpose is to ensure transparency, accountability, compliance with current regulations, quality control throughout the value chain, and others [Safdie, 2023].

The traceability of a product boils down to the ability to uniquely identify it at any point in the value chain, so it usually ends up with identification tools such as barcodes, QR Codes, Radio Frequency Identification (RFID) tags, Universal Unique Identifiers (UUIDs), and more. These technologies are attached to the lots or the individual products and carry information that allows the tracking and traceability of the specific product. With these systems, it's possible to create a product history, permitting various organizations to rapidly and accurately identify any product for issues relating to safety, quality, authenticity, and others [Food Standards Agency, 2002].

In the food industry, traceability is the process of documenting and tracking the movement of food products and their information across the value chain to guarantee food safety, quality control, transparency with consumers, and regulatory compliance. Also, it helps to improve value chain management, allow product recall, walk towards sustainable development, and others [Food Standards Agency, 2002].

2.1.4 Value Chain

A value chain is a concept that describes all the activities that companies and workers engage in to create and deliver a product or service to its end consumers. In other words, it's a set of unified activities, contained to a singular or many organizations, that add value at each stage in the production process [Gereffi and Fernandez-Stark, 2016].

The value chain framework helps to understand how organizations are assembled by analyzing their internal operations, focusing on the succession of value-adding steps from creation to end consumers. In other words, it's a framework for organizations to analyze and escalate their internal operations, allowing them to maintain or improve competitive advantages, improve profits, decrease inefficient operations, etc. [Tardi, 2023, Gereffi and Fernandez-Stark, 2016].

The food value chain, it is as mentioned but applied to food products. It's a bunch of operations since the food leaves the farm until it arrives at the final consumer, such as producing, processing, distributing, consuming, waste management, and retail [Deloitte, 2013, Gereffi and Fernandez-Stark, 2016].

Efficient food value chain management is essential to increase traceability and waste management strategies, reach consumers' demands, optimize the chain, and others. With those, it's possible to empower food safety and quality, sustainability in this industry, etc. [Deloitte, 2013].

2.2 Tools

This section presents some crucial tools, such as frameworks and libraries, used to develop our traceability platform and explains how we used them.

2.2.1 API

Application Programming Interfaces (APIs) are sets of rules and protocols that permit different software applications to communicate and interact with each other. With this, developers can access the functionality of other software components without the need to understand any details about how they work [IBM, 2023d].

REST API

First, Representational State Transfer (REST) is an architectural style that provides standards for communication and interaction between software applications on the web [Codecademy Team, 2023]. Now, REST APIs, also known as RESTful APIs, are APIs that adhere to the principles of REST [IBM, 2023c].

We're using REST APIs to realize the communication between the frontend and backend.

2.2.2 Axios

Axios is a promised-based JavaScript (JS) library used to make Hypertext Transfer Protocol (HTTP) requests from Node.js (server-side) and browsers (client-side). It supports HTTP methods like POST, GET, PUT, DELETE, etc., and beyond the promised-based requests, it offers interceptors, error handling, timeouts, and others [Axios, 2023].

We're using Axios on the frontend to communicate with the backend and third-party APIs to make Create, Read, Update, and Delete (CRUD) operations.

2.2.3 bcrypt

Bcrypt is a widely used cryptographic hashing method for password hashing and storage [Grigutytė, 2023]. It applies a salted and iterated approach to hashing, improving the security and making it less susceptible to attacks, such as a rainbow table, brute force, dictionary-based, and others [Grigutytė, 2023, Sriramya and Karthika, 2015].

We're using bcrypt to encrypt all of the passwords on the platform.

2.2.4 Bootstrap

Bootstrap is a popular, open-source frontend toolkit for developing fast, responsive, mobile-first websites [Bootstrap, 2023, Alexandrea, 2023]. It is great for developers because it provides pre-designed templates, components, tools (e.g., grid system and navigation elements), and plugins [Bootstrap, 2023].

React Bootstrap

React Bootstrap is a React extension that provides React components to Bootstrap.

We're using Bootstrap, specifically React Bootstrap, to help us design our frontend pages.

2.2.5 BPMN

Business Process Model and Notation (BPMN) is a standardized graphical notation to model and represent business processes plainly and understandably. It's popular with organizations because it allows them to easily portray their business processes, enabling stakeholders to participate in the process design, analysis, and improvement in general [Object Management Group, 2023].

We're using BPMN to model and represent the business processes of some fresh and transformed fruits and vegetables and our generic business process for the F&V value chain.

2.2.6 Express.js

Express.js, known as just Express, is a fast, minimalistic, unopinionated web framework for Node.js, providing a simple way of creating APIs and web apps [Express, 2023]. Developers use it to simplify the handling of routing, middleware, HTTP requests and responses, and others, with the possibility to integrate it with other frameworks and libraries [Pramanick, 2023].

We're using Express.js to develop the backend APIs.

2.2.7 Hyperledger Fabric

Hyperledger Fabric is a modular and open-source enterprise blockchain framework created by the Hyperledger project of the Linux Foundation. It's intended to provide a secure and scalable platform for building various Decentralized Applications (DApps) and blockchain-based solutions for different industries, including healthcare, finance, and supply chain management [Hyperledger Foundation, 2018].

We're using Hyperledger Fabric to develop the blockchain.

2.2.8 JWT

JSON Web Token (JWT) is an open standard for securely transmitting and verifying data between parties as a JavaScript Object Notation (JSON) object [JWT, 2023]. It's widely used for authentication and authorization in client applications and single sign-on systems [IBM, 2023a]. JWT consists of three parts: the Header, which contains information about the token type and the signing algorithm; the Payload, which holds the statement about the user and custom data; the Signature, which carries a signature of the encoded Header and Payload, the algorithm specified in the Header and a secret [JWT, 2023].

We're using JWT to create access and refresh tokens to handle authentication and authorization, providing security in the traceability platform.

2.2.9 Leaflet

Leaflet is a lightweight, open-source JavaScript library for creating interactive and customizable maps for web and mobile apps. It's compatible with most major web and mobile browsers. This tool offers high customizability features such as map controls, tile providers, visuals, interactions, and more, making it a popular choice with developers [Leaflet, 2023].

React Leaflet

React Leaflet is a React extension that provides React components to Leaflet maps.

We're using Leaflet, specifically React Leaflet, to create the traceability map for the web and mobile apps.

2.2.10 MERN Stack

MongoDB, Express.js, React.js, and Node.js (MERN Stack) is a stack of JavaScript technologies offering a full-stack solution to build web apps only using JS, i.e., developers can work on the frontend and backend using only one programming language. In this stack, React is for the frontend, Node.js and Express.js are for the backend, and MongoDB is for the storage [MongoDB, 2023c].

We're using MERN Stack to develop a 3-tier architecture for the traceability platform.

2.2.11 MongoDB

MongoDB is a widely used not only SQL (NoSQL) database (DB) that stores data in a flexible and JSON-like format known as Binary JSON (BSON), designed to overcome the constraints of relational and other non-relational DB approaches [MongoDB, 2023d, MongoDB, 2023b]. Its format and the fact that the documents are polymorphic enable easy schema evolution and add to the database's scalability and flexibility [MongoDB, 2023b]. This DB is used for modern web and data-intensive apps such as big data processing because it provides high performance, high availability, horizontal scalability, and others [MongoDB, 2023d].

We're using MongoDB to create the off-chain database.

2.2.12 Mongoose

Mongoose is a MongoDB Object Data Modeling (ODM) library for Node.js [Hall, 2022]. It's popular with Node.js developers because it permits the definition of data schemas, the creation of models, the execution of various CRUD operations, and others while providing features such as data validation, middleware support, and query modeling [Hall, 2022, Rimu, 2023].

We're using Mongoose to facilitate the interaction between Node.js and MongoDB, allowing us to work in an organized, structured, and familiar manner.

2.2.13 Node.js

Node.js is a free, open-source, cross-platform JavaScript runtime environment built on Google Chrome's V8 JavaScript engine, enabling developers to create server-side apps [Node.js, 2023]. Its success comes from the various packages (using npm), libraries, modules, and compatibility with JS frameworks, simplifying the development of APIs, web servers, real-time applications, network-based systems, and others [Node.js, 2023, Pramanick, 2023].

We're using Node.js to develop the traceability platform's backend.

2.2.14 React

React is a popular JavaScript library that allows developers to create fast, interactive User Interfaces (UIs). It uses a component-based model, i.e., deconstructs the UI into smaller reusable pieces (known as components). React is also declarative and uses a virtual Document Object Model (DOM) that allows the update of only the necessary components instead of the whole page, offering a better performance [React, 2023, Alexandrea and Muhammad, 2023].

We're using React to develop the web app's frontend.

2.2.15 React Native

React Native is a cross-platform, open-source JavaScript library that enables developers to create UI for native applications. It allows one to write code once and deploy it to different mobile platforms (e.g., Android and iOS). React Native benefits from React, combining its components with native ones to create responsive and interactive UIs [React Native, 2023, Meta Open Source, 2021].

We're using React Native to develop the mobile app's frontend.

2.2.16 Token

Tokens are artifacts that carry information to help with authentication and authorization in an application [Arias and Bellen, 2021].

Access Token

An Access Token is an artifact used in token-based authorization and authentication that client-side applications can use when wanting to make secure calls to an API. This token informs the server-side that the sender can or can't perform specific tasks or access certain resources. These tokens usually have a short validity [Arias and Bellen, 2021, Auth0, 2023a].

We're using access tokens for the authentication and authorization.

Refresh Token

A Refresh Token is a credential artifact that refreshes the access token without user action, providing a smoother and more secure user experience [Auth0, 2023b].

We're using refresh tokens to refresh the access tokens when these expire.

2.2.17 UUID

A UUID is a 128-bit value (i.e., 32-alphanumeric value) for uniquely identifying anything computer-based across different systems [Gasparik, 2023]. These are extremely useful in distributed systems prone to high conflicts, which need the uniqueness of entities such as DB records, separate components, keys, and others [Gasparik, 2023]. The creation of UUID differs depending on the version (v). In version 4, random and pseudo-random numbers are used [da Paixão, 2018].

We're using UUID, specifically v4, to create IDs for the records in the blockchain.

2.3 Languages

This section presents the programming languages used to develop our project.

2.3.1 JavaScript

JavaScript is a high-level programming language widely known for web development. However, with the help of tools such as Node.js, it can develop server-side and other non-browser applications. JS supports object-oriented and functional programming, offers various libraries and frameworks, and more [MDN Web Docs, 2023].

We're using JS to develop everything on the platform: the frontend, the backend, the smart contract, and others.

Chapter 3

Background Review

There are several traceability platforms for different industries [da Cruz and Cruz, 2020], such as textiles [Agrawal et al., 2021], pharmaceuticals [Akhtar and Rizvi, 2021], cosmetics [Othman et al., 2020], and especially in the food industry. Some platforms track specific products, such as [Fernandes et al., 2022] for tracing olive oil or [Cruz and da Cruz, 2020b] designed to trace fish. Other platforms are more generic, as in [Baralla et al., 2019], which traces local food products from the island of Sardinia. Table 3.1 summarizes the F&V traceability platforms, highlighting the used technologies, measurements, and other information.

3.1 Fruit and Vegetable Traceability using Blockchain Technology

Blockchain is a technology that seems to satisfy all traceability needs [da Cruz and Cruz, 2020], so multiple traceability platforms have been using it.

The work developed by [Yang et al., 2021] presents a traceability system for fruits and vegetables based on blockchain, aiming to solve current traceability problems. The solution uses a "Blockchain + Database" method and a query platform, where the blockchain makes it possible to store data immutably, and public data is stored in the database to reduce the overload on the blockchain chain and perform more efficient queries. Hyperledger Fabric is used to develop the blockchain, the smart contracts with the Go language, and the query platform with .Net/C#. The conclusion is that some of the current problems in traceability are alleviated, but to meet business needs, it is necessary to invest in multi-chain.

In [Keerthivasan et al., 2022], the authors present a blockchain-based traceability system for fruits and vegetables to analyze the impact of this technology on this value chain. The conclusion is that blockchain is a promising technology in this field, but many obstacles may hinder its progress. Some barriers include expert opinion, guidance, strategies, and management structures.

Feng Tian [Tian, 2016] analyzes two types of agri-food products: the first being fresh products (fruits and vegetables) and the second being meats (pork, chicken, and beef). A combination of blockchain and RFID is used to achieve product safety and quality in a "From Farm To Fork" process at all stages. Blockchain is used in a decentralized way so that all interested parties have access to any transaction and product information. This platform also provides monitoring and tracking throughout the F2F process. In [Tian, 2017], Tian presents a reinvention of his previous solution, keeping the platform decentralized, using blockchain, Internet of Things (IoT) tools, and Hazard Analysis and Critical Control Points (HACCP).

The authors of the work seen in [Xu et al., 2021] propose a traceability system for urban fruits in a "From Farm To Fork" process based on blockchain for IoT to mitigate the risk of poor quality and fraud. In addition, the blockchain consensus mechanism and smart contract model are developed.

The work developed by [Salah et al., 2019] presents a solution for soybean traceability, using an Ethereum blockchain with smart contracts and the Interplanetary File System (IPFS) to store some of the data, avoiding overloading the blockchain chain.

3.2 Fruit and Vegetable Traceability using Other Technologies

The authors of the work in [Bevilacqua et al., 2009] reinvented the business process for the vegetable supply chain and created a computational platform for managing the traceability of the respective products. A framework based on the Event-driven Process Chains (EPCs) methodology is presented, using Activity-Based Costing (ABC) to define and analyze the current state of the supply chain. To provide information to the final consumer, Architecture of Integrated Information Systems (ARIS) is used to develop a Web interface.

Massimo Conti [Conti, 2020] proposes using Android smartphones with Near Field Communication (NFC) technology throughout the supply chain, i.e., in a "From Farm To Fork" process to achieve traceability of fruits and vegetables. The architecture consists of a product identification system that transmits various data to a cloud database through different smartphone apps throughout the food chain process. Any actor in the chain (e.g., final consumers, farmers, government institutions) has access to the information stored in the cloud.

In [Xie et al., 2022], a system for the traceability of apples in a "From Farm To Fork" process was developed and tested in an orchard in Qixia, Shandong Province, China, for about a year. It's an integrated machine-to-machine system that auto-collects information from the different operations. This solution includes a hardware system based on IoT, a smart cloud farming platform, and a mobile app. End consumers can track products using QR Codes.

One of the most recent ways to trace fruits and vegetables is through Deoxyribonucleic Acid-based (DNA-based) traceability, where chemical/biochemical, biomolecular, and isotopic techniques are used to confirm the origin of products. We can see literature on these techniques applied to avocados [Muñoz-Redondo et al., 2022], chili peppers [Trovato et al., 2022], citrus fruits [Coelho et al., 2021, Pei et al., 2022], jujube [Song et al., 2021], lemon and lime [Jungen et al., 2021], peaches [Li et al., 2021, Tamasi et al., 2021], nectarines [Tamasi et al., 2021], red grapes [Radulescu et al., 2021], tomatoes [Panebianco et al., 2022], among others. The authors in [Ben Ayed et al., 2022] show that a combination of DNA-based traceability techniques and other 4.0 traceability technologies (e.g., Blockchain, IoT, Artificial Intelligence (AI)) can offer a reliable way to guarantee the quality of products, in particular olive oil. Another work that addresses DNA-based traceability and traceability 4.0 tools can be seen in [Hassoun et al., 2023], where several current techniques and tools for traceability of fruits and vegetables are analyzed.

Table 3.1: Summary of Relevant Information from the Traceability Platforms in the Background Review

Traced Product	Farm to Fork	Storage	Other Technologies	Measures Quality	Measures Sustainability	References
All Fruits and Vegetables	No	Blockchain + Database	N/D	No	No	[Yang et al., 2021]
All Fruits and Vegetables	No	Blockchain	N/D	No	No	[Keerthivasan et al., 2022]
All Fruits and Vegetables (+Meats)	Yes	Blockchain	RFID	No	No	[Tian, 2016]
All Fruits and Vegetables (+Meats)	Yes	Blockchain	RFID, HACCP	No	No	[Tian, 2017]
Chinese Urban Fruits	Yes	Blockchain	IoT	Yes	No	[Xu et al., 2021]
Soybean	No	Blockchain	IPFS	No	No	[Salah et al., 2019]
All Vegetables	No	N/D	EPC, ABC, ARIS	No	No	[Bevilacqua et al., 2009]
All Fruits and Vegetables	Yes	Cloud	NFC	No	No	[Conti, 2020]
Apple	Yes	Cloud	IoT, QR Code	Yes	No	[Xie et al., 2022]

3.3 Conclusions

To conclude our background review, we can affirm that blockchain has been a popular choice for traceability systems because of the possibility of immutability, transparency, and decentralization. Despite these positive features, this tool has many disadvantages and obstacles discussed in the studied solutions, such as scalability issues and management structures.

One of the points where we will stand out is the measurement of sustainability indicators since none of the solutions studied measure them. Regarding sustainability indicators, only two solutions measure them using IoT devices.

Most solutions opted for traceability in an F2F process, which allowed data collection at each step. Later, this information is available to any interested actor.

Some solutions that do not use blockchain were studied, and despite being able to track products and measure quality, we can say, based on the other solutions studied, that a tool that offers immutability is necessary, as data can be tampered with, enabling food fraud.

Finally, DNA-based traceability is a consistent way to guarantee origin, but it currently cannot guarantee quality and other factors.

Chapter 4

Modeling the Fruit and Vegetables Value Chain

Chapter 4 presents the various steps analyzed to start the value chain modeling and arrive at our generic business process. Since we're working in a From Farm To Fork process, it's imperative to know all the steps from harvesting to the final sale point to extract a generic business process. To accomplish this, we studied different F&V business processes and interviewed local farmers and market workers to discover the different stages the products go through, from harvesting to their final destinations.

4.1 Fruit and Vegetable Business Processes

As mentioned, we'll study different F&V business processes with and without transformations. More attention is paid to products grown in Portugal (e.g., Pera Rocha do Oeste, Framboesa do Algarve, Pêssego da Cova da Beira) and in Europe (e.g., Olive Oil, Tomato, Mushroom). Next, we'll present examples of business processes for fresh and transformed fruits and vegetables. We conclude with the results from the interviews with local farmers and market workers. The notation used to represent the business processes is BPMN.

4.1.1 Fresh Fruits and Vegetables

For the example of fresh fruits and vegetables, we're presenting the business process of a fruit grown in Portugal - Pera Rocha do Oeste (Fig. 4.1).

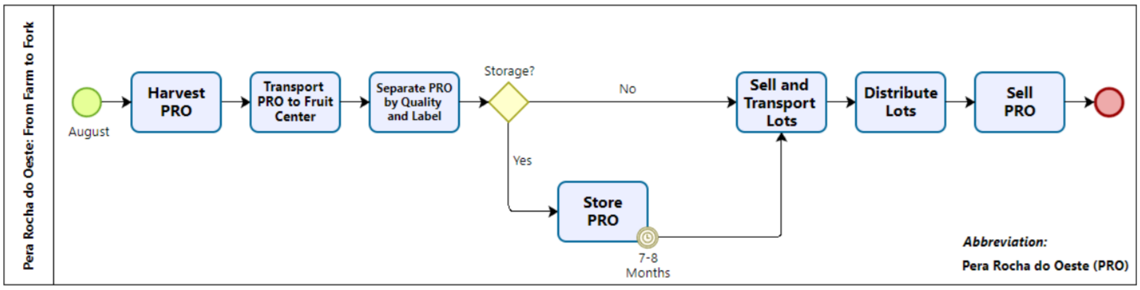

Figure 4.1: Business Process of Pera Rocha do Oeste (obtained through the study of [Pera Rocha, 2023]).

4.1.2 Transformed Fruits and Vegetables

For the example of transformed fruits and vegetables, we're presenting the business process of a transformed fruit grown and produced in Europe - Olive Oil (Fig. 4.2).

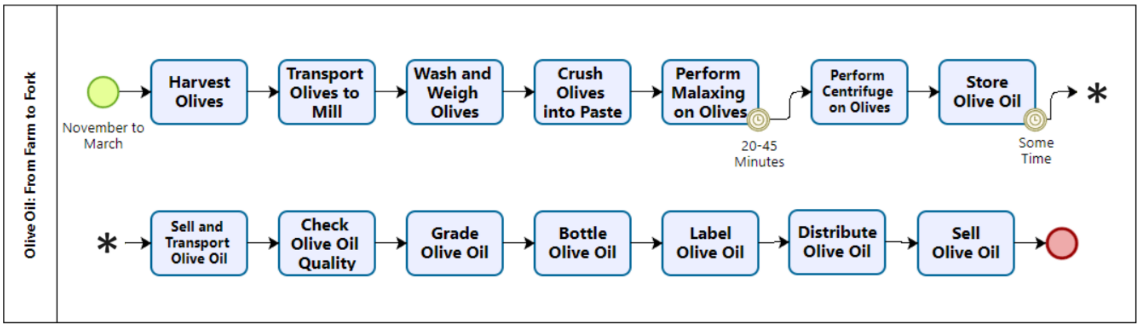

Figure 4.2: Business Process of Olive Oil (obtained through the study of [Eat Happy Project, 2015]).

4.1.3 Local Farmers and Market Workers

Interviews were conducted with local farmers and market workers in Braga and Amares, Portugal, to understand their roles in the value chain.

Throughout the conversation with the farmers, we got great detail about the first step in this value chain - Harvest. They explained that there is an almost immediate quality check to see which products can continue the process. Also, some farmers explained the various post-harvest activities, giving us our second step in this value chain - Post-Harvest. After this, the farmers didn't have any further detailed explanations.

With the conversation with the market workers, we learned that they're inserted in the final step in this value chain but nothing more because most markets got their products directly from the fields (i.e., the Harvest step).

4.2 Steps

After studying multiple F&V business processes and speaking with local farmers and market workers, we concluded that from harvest to the final consumer, there are four steps and two essential activities.

4.2.1 Harvest

Harvesting is the first activity in our value chain. After harvesting the product, it will go through a classification to see which ones are available for sale. In this stage, it is necessary to store the activity and the measurements that come with it.

4.2.2 Post-Harvest

After selecting the products available for sale, they will undergo various treatments (e.g., fumigation, bleaching, acid immersion, and biopesticides treatments) to enhance the color and appearance of the product and extend its useful life. After each treatment, there is a new quality reassessment. When no more treatments are needed, the products can be stored for a certain period. When the products are ready, they will be packaged, creating lots. Finally, these lots are sold and transported to the respective organizations. This stage can be divided into "Treatment" and "Packaging", being necessary to store the treatments carried out, the measurements that come with them, and the lots created.

4.2.3 Processing

When lots arrive at an organization, they will carry out the quality assessment. Lots that pass this evaluation and do not need storage will be processed immediately. First, they will undergo primary processing, which includes techniques such as sorting, washing, drying, cutting, and new packaging, among others. Then, if no storage is needed, they will go through secondary processing, which includes activities such as drying and osmotic dehydration to preserve and add value to the products. As mentioned, it is possible to add value by processing the products to obtain derivatives or by-products like juices, gelatins, jellies, sweets, syrups, sauces, canned food, alcoholic beverages, vinegar, oils, etc. After each processing, there is a new quality reassessment, and when no further processing is needed, they are sold and transported to the respective organizations. This stage ends when the organization is a distributor. In this step, it is necessary to store the different processes carried out, the measurements that come with them, and the lots created.

4.2.4 Distribution

Finally, the lots will be distributed to different sites such as supermarkets, end consumers, retail stores, food services, etc. In this stage, it is necessary to store the distribution activity and the measurements that come with it.

4.2.5 Transport & Storage

These two activities are not stages since they appear several times in the ones mentioned above. However, they must be highlighted because they are vital to guarantee the quality of the lots.

4.3 Generic Business Process for the Fruit and Vegetable Value Chain

To conclude this chapter, we used all the information about the obtained steps to create a generic business model for the F&V value chain, from the harvest to the final sale (Fig. 4.3).

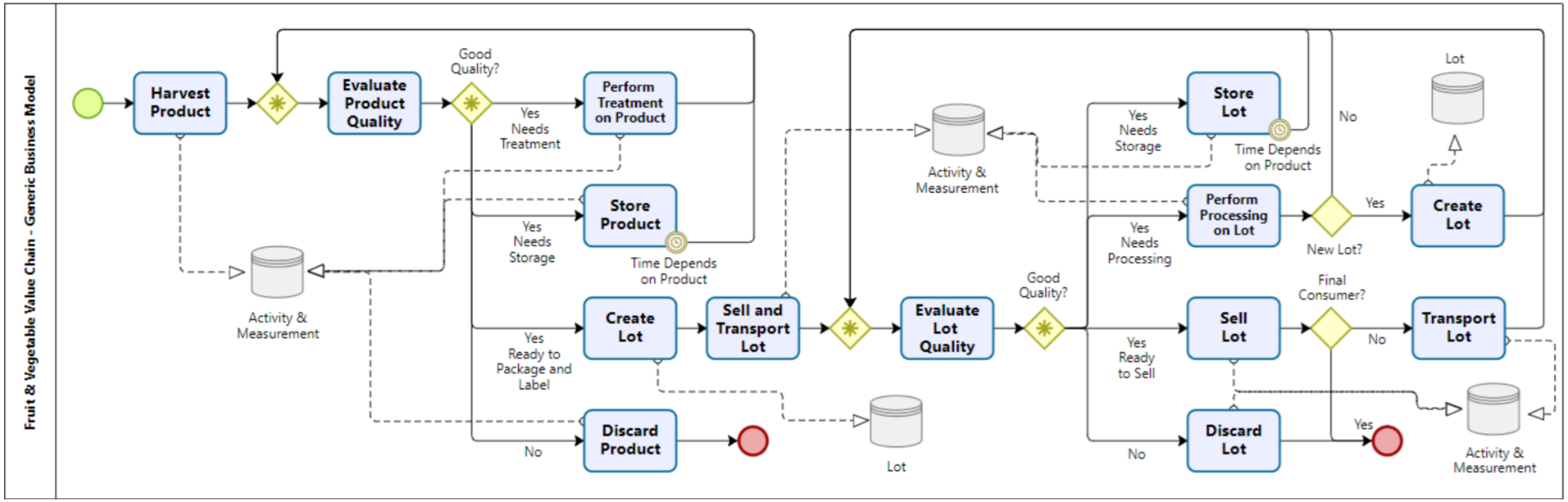

Figure 4.3: Generic Business Process for the Fruit and Vegetable Value Chain.

Chapter 5

Proposed Solution

Chapter 5 details different aspects of our proposed solution for creating an F&V traceability platform, starting with a use-case model to identify actors and their activities on the platform. The following section presents the domain model, detailing the data that needs to be stored and the methods for tracing quality and sustainability indicators. Finally, we show the proposed architecture for this solution.

5.1 Identification of Actors and their Activities

After studying the results obtained in Chapter 4, we can state that there are three actors in this value chain. Fig. 5.1 illustrates the actors and their activities on the platform using a use-case model.

1. **Final Consumer** - represents anyone interested in the traceability of lots and information about them. This actor does not need to log in, being able to consult information about products and lots, as well as trace them back to their origin.

2. **Operator (Normal or Administrator)** - this actor is more generic and represents any employee of an organization that performs functions in this value chain. Both operators have to log in, and the Normal Operator can do everything the final consumer does, plus register different activities (e.g., harvesting, transport, treatment). The Administrator Operator can do everything the final consumer does, plus tasks such as updating public information (i.e., information about organizations, measure-

ments, and products) and adding new products, activity types, etc.

3. **Platform Administrator** - represents an entity that manages organizations in the value chain.

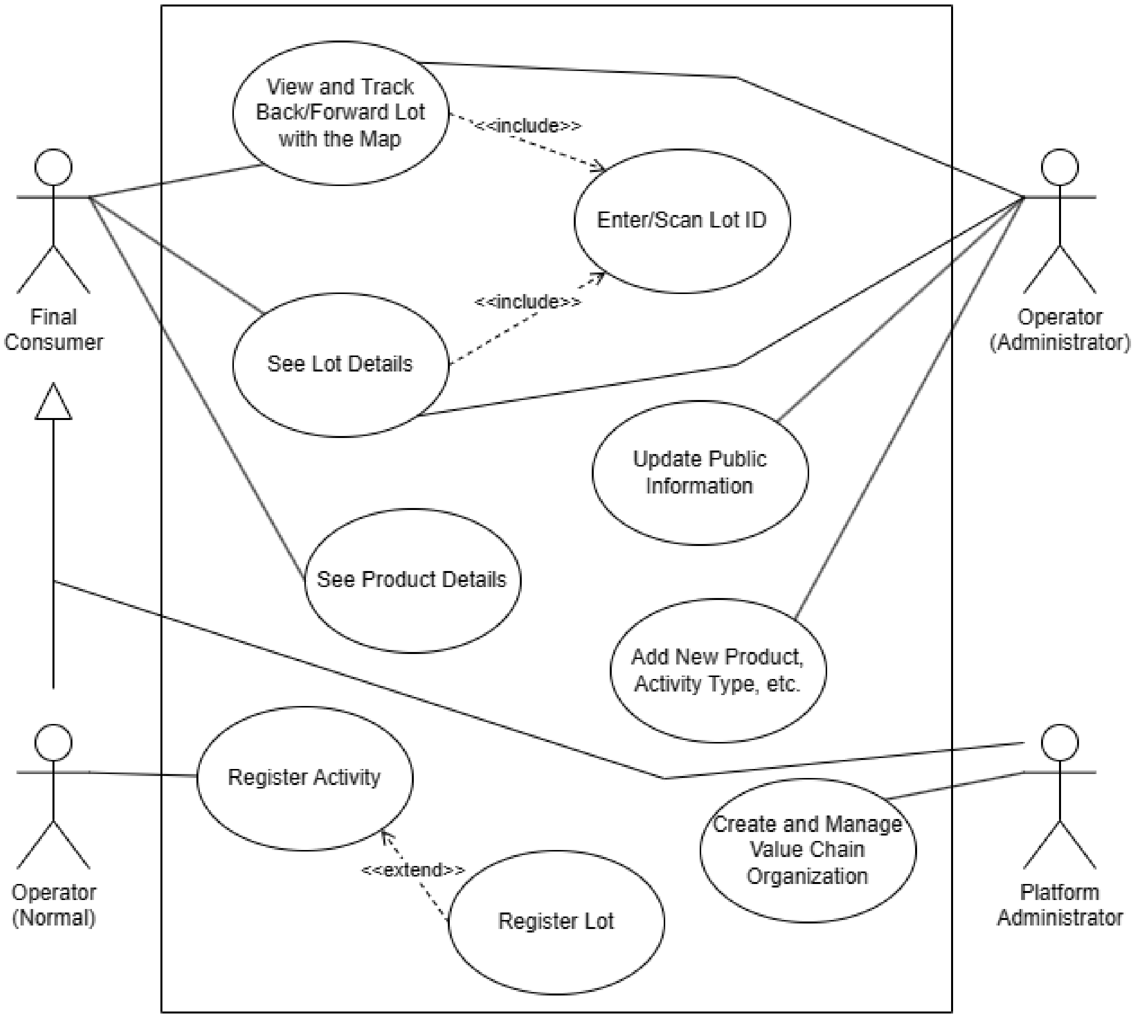

Figure 5.1: Use-Case Model for the Fruit and Vegetable Traceability Platform.

5.2 Data to Store and Tracking Quality and Sustainability

With the stages studied and the actors and their activities on the platform identified, it is necessary to know what data will be stored for the platform to work. As we are going to use a B+DB methodology, it is necessary to differentiate the data that is stored in the blockchain (on-chain) and the database (off-chain). In the on-chain storage, we will only keep the data that will make our traceability immutable (i.e., lots and activities carried out on them). In the off-chain storage, we will store everything else (e.g., organization information, product information). We will also detail the tables with which we measure the quality and sustainability of the products. A visual representation of all the data that needs to be stored can be seen in Fig. 5.2, represented with a domain model. All the different used tables are:

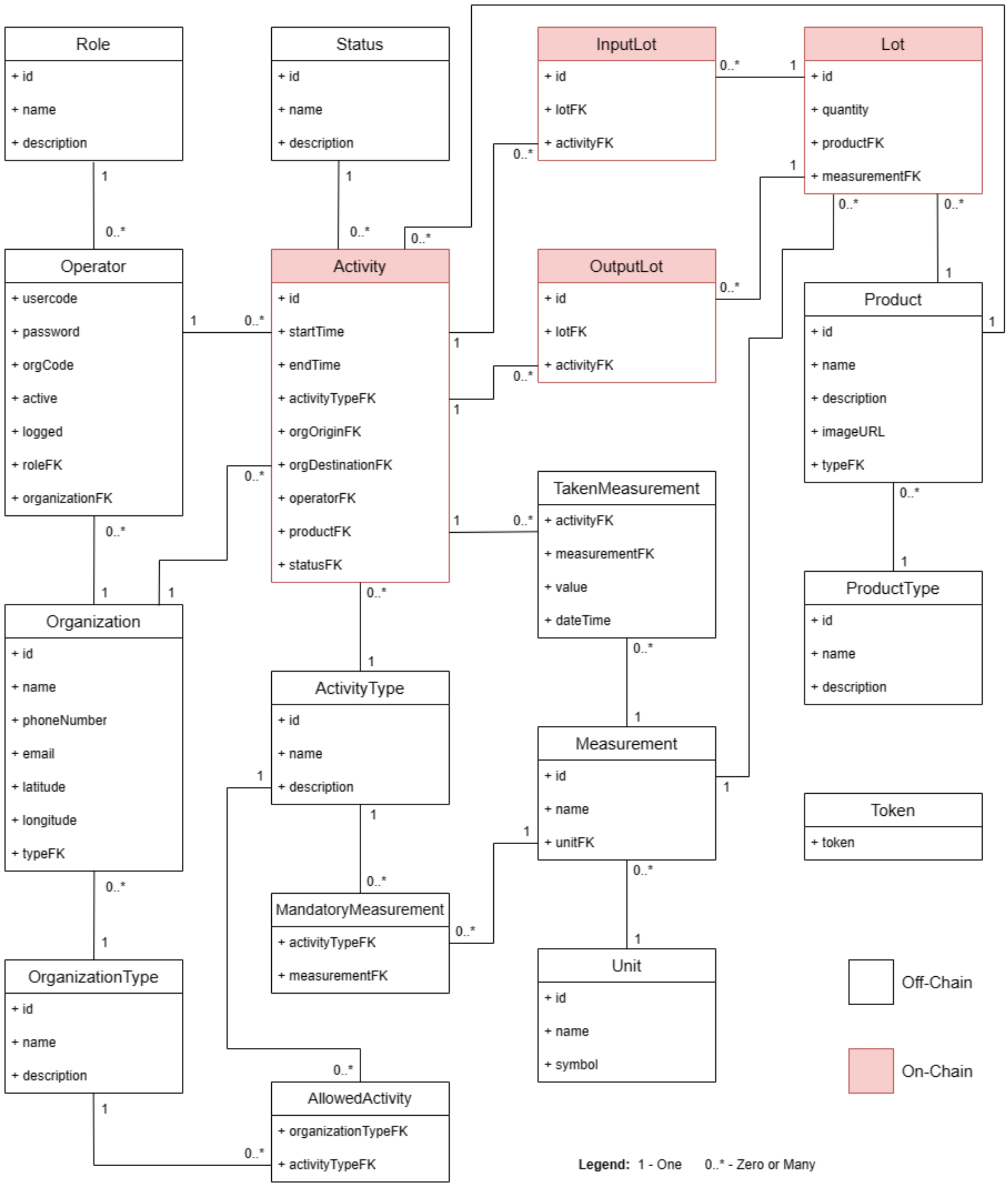

Figure 5.2: Domain Model for the Fruit and Vegetable Traceability Platform.

- **Organization, OrganizationType, AllowedActivity, Operator & Role** - Organization represents an entity linked to activities carried out on our platform, defined by their name, phone number, email, and location (coordinates). OrganizationType represents what type of organization it is (e.g., transport, control, and treatment), defining what activities it will carry out on the platform. AllowedActivity results from a Many-to-Many (M2M) relationship between the OrganizationType and ActivityType tables, allowing the definition of the types of activities that organizations can carry out on the platform. Operator will represent an employee of an organization, being responsible for taking action on their behalf, defined by the code and password that allows him to log in to the platform, by a private code (orgCode) only recognizable by the organization where he works, by his role, which will determine whether it is a Normal/Administrator Operator or a Platform Administrator and by another two fields (active and logged) which will allow us to know whether the operator is logged in and the other controls whether a user is active or not, i.e., whether they will be able to log in and carry out activities on the platform. The role represents the different roles in the platform.

- **Product, ProductType & Lot** - Product represents any product in our system (i.e., any fruit or vegetable, whether or not transformed), defined by its name, description, and type. Lot represents a product in quantity and with any event performed on it. ProductType represents the different product types in the platform (e.g., Fresh, Transformed).

- **ActivityType, Measurement, TakenMeasurement, Unit, MandatoryMeasurement** - ActivityType stores a type of activity (e.g., Harvest, Transport, and Storage). Measurement saves the name of important and specific conditions for each activity (e.g., temperature, humidity, and fuel), TakenMeasurement stores an actual value to those fields (e.g., 45), and Unit saves the name and symbol of a measurement (e.g., degrees Celsius and the corresponding symbol "ºC"). Mandatory measurement results from the many-to-many relationship between the ActivityType and Measurement tables, allowing us to know the obligatory measurements by activity type.

- **Status** - Status represents the current stage of an activity (e.g., Started, Ongoing, and Finished).

- **InputLot & OutputLot** - InputLot is the lots going through an activity, and OutputLot is the lots that originated from that activity. Both tables result from a M2M connection between the Activity and Lot entities.

- **Activity** - Activity represents the joining of all the entities described until now, defined by the operator in charge of the activity, the organization where the activity starts (orgOriginFK), the organization where the activity will end (orgDestinationFK), the start and end timestamps, the type, the product, and the status.

- **Token** - Token is going to store all the active refresh tokens.

5.2.1 Tracking Quality and Sustainability

For a well-maintained value chain, the collection and storage of quality and sustainability indicators plus traceability are essential. These indicators enable the creation of real-time risk prediction systems [Donaghy et al., 2021] and applications in smart and precision agriculture [Misra et al., 2022].

Several types of sustainability and quality indicators can be collected, such as water, to increase use efficiency [Ciruela-Lorenzo et al., 2020] and to monitor quality [Kamilaris et al., 2018]; waste, to identify sources of food waste in the value chain [Kamble et al., 2020] and to develop proactive practices to support resource recovery from waste [Sgarbossa and Russo, 2017]; soil, to maintain soil fertility [Hou et al., 2020] and to obtain the information necessary to identify soil conditions [Kolipaka, 2020]; infections, to detect types of infections [Jiang et al., 2021].

After that, we can confirm that collecting and saving quality and sustainability indicators can help increase food safety and mitigate food fraud and the environmental and ecological impacts of food production. So, next, we will explain the tables needed to store these indicators and give an example of some of the indicators we will collect and save. Table 5.1 illustrates an example of some of these indicators per activity.

Tables Used

Because the quality and sustainability indicators aren't necessary to maintain immutable traceability, we will use the database (off-chain). We will use the three measurement tables and the unit table to keep them stored.

Any activity has mandatory measures to store, defined by the "MandatoryMeasurement" table. In addition to the obligatory measurements, organizations may want to save others. The "TakenMeasurement" table keeps all the measurements, in more detail, their value and timestamp. In the "Measurement" table, we store the names of all the measurements that organizations wish to save, and in the "Unit" table, we have the unit's name and the symbol that represents it. Fig. 5.3 illustrates a small example of the process.

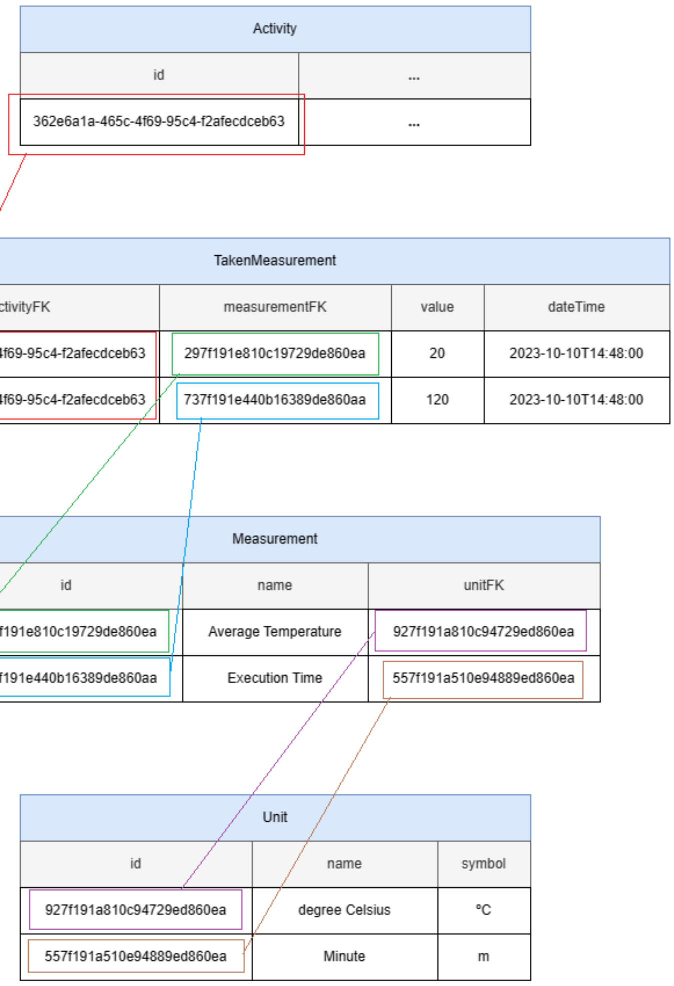

Figure 5.3: Example of the Storage Process of Quality and Sustainability Indicators.

Table 5.1: Example of some Quality and Sustainability Indicators per Activity

Activity	Indicator					
	Quality			Sustainability		
	Average Temperature	Execution Time	Soil Moisture	Energy	Product Waste	Water
Harvest	√	√	√	√	√	
Wash	√	√		√	√	√
Packaging	√	√		√	√	
Transport	√	√			√	
Storage	√	√		√	√	

5.3 Architecture

With all the previous information, we developed an architecture for the F&V traceability platform. It is a 3-tier architecture, achieved through the use of MERN Stack. As the name suggests, it consists of three distinct layers that we will detail in this section. A simple representation of the architecture is illustrated in Fig. 5.4.

5.3.1 Storage

As mentioned, the development of the storage layer will use a B+DB methodology. Briefly, the methodology will help with the scalability of the blockchain, i.e., it will relieve an overload in the blockchain's chain by only saving the data necessary to make the traceability immutable, while public data (e.g., information about products, organizations, and measurements) is going to the database, having a field in the two storage technologies that allow its connection.

To create our blockchain, we'll use Hyperledger Fabric v2.4.9. The creation of the blockchain's smart contract will be with JS. To develop the database, we'll use MongoDB, a NoSQL DB.

5.3.2 Backend

Next, we have the back end. This layer will communicate with the others and has two major components - Services and APIs.

Starting with the services developed with Node.js, allowing information management (i.e., create, retrieve, and update data). This component will communicate with both storage systems to send and retrieve data.

The second one is the APIs, developed using Express, allowing communication with the frontend apps.

5.3.3 Frontend

Finally, we have the frontend, where we'll develop the mobile and web apps with React Native and React, respectively, using Axios to communicate with the APIs. This way, any actor in the supply chain (e.g., final consumers, farmers, government institutions, companies) can view a product/lot.

Users will use the apps to track lots to their origin through an interactive map developed with Leaflet. Also, on this map, users will see all the activities carried out by the lots, and when clicking, see details about them. It's also possible to do forward traceability, i.e., verify where the lots have gone. This feature is handy for organizations in the case of contaminated lots, only needing to scan the QR Code or enter the lot ID manually.

The mobile app, developed for the final consumers, will permit the traceability of lots and the visualization of lot and product details. The web app has the same functions as the mobile one, plus it is where the operators and platform administrators will log in to carry out their duties.

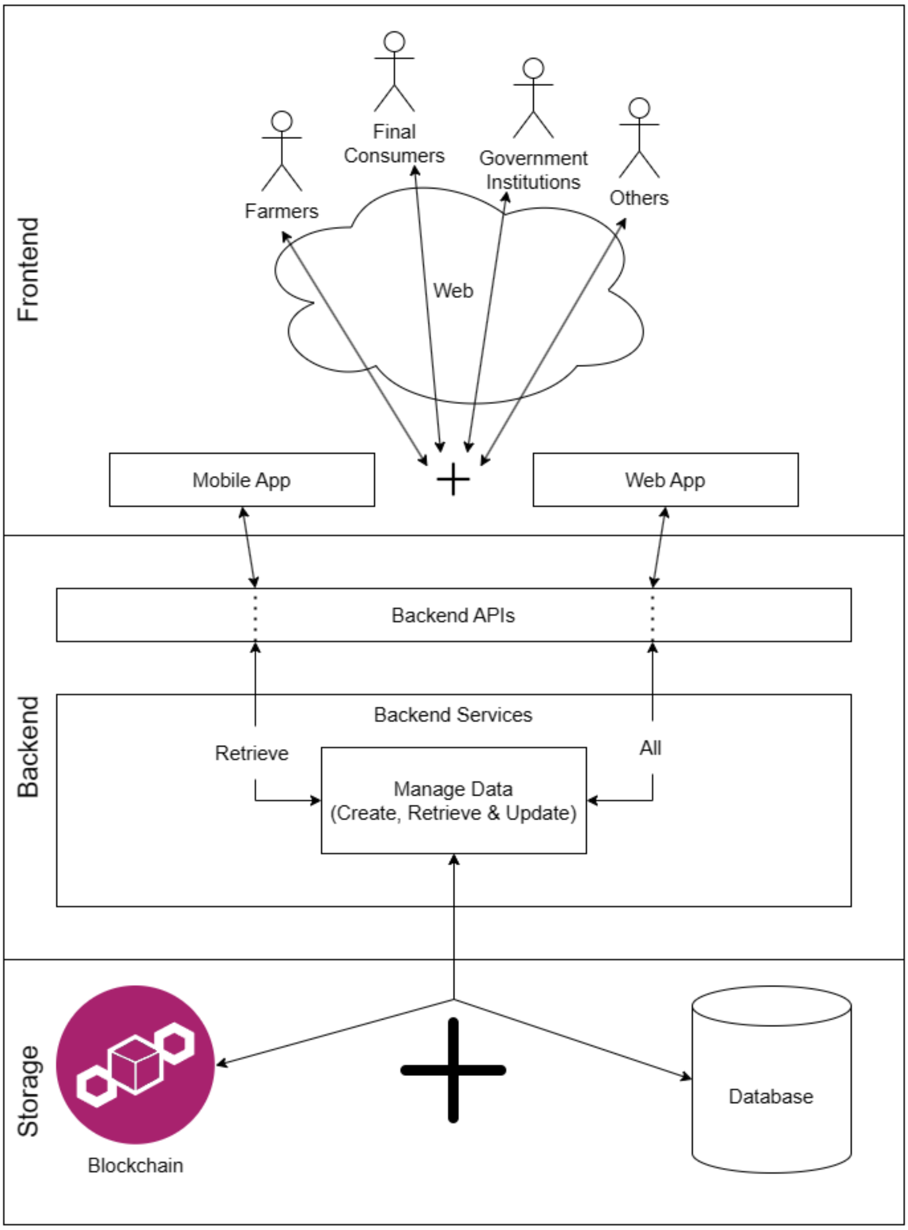

Figure 5.4: Architecture for the Fruit and Vegetable Traceability Platform.

Chapter 6

Platform Implementation

Chapter 6 explains the "Whys" and "Hows" relating to the implementation of the traceability platform, highlighting various steps in its development, starting with picking a name and logo for our platform. Next, we detail the blockchain, explaining the reason for the chosen consensus mechanism and presenting the smart contract (chain code). Continuing with the storage, we clarify the choice for our database and present its development. After that, we demonstrate some forms of automatic data collection in the platform and some security mechanisms implemented. Then, we detail how our traceability algorithm works and present the mobile and web apps. Lastly, we show the results of the tests performed on the platform.

6.1 Name and Logo

The first step was to name our platform and create a logo. The chosen name was "FreshTracker". Fig. 6.1 presents the platform's logo.

Figure 6.1: FreshTracker - Traceability Platform for Fruits and Vegetables Logo.

6.2 Hyperledger Fabric

As mentioned, we wanted a decentralized blockchain-based solution. After studying some options, we chose Hyperledger Fabric v2.4.9 because of its balance between scalability and security, the possibility of creating smart contracts (called "chaincode" in Fabric) authored in different programming languages (e.g., Java, JavaScript, and Go), and its optimizable for various industries, including supply chain management [Hyperledger, 2023a].

Next, we'll discuss our choices for the consensus mechanism and present our smart contract.

6.2.1 Consensus Mechanism

Selecting the appropriate consensus mechanism is crucial when building a blockchain-based application. Hyperledger Fabric v2.4.9 offers various mechanisms, such as Raft and Solo, each with strengths and weaknesses [Hyperledger, 2023b].

To verify if the consensus mechanism meets our application-specific requirements, such as scalability, security, simplicity, and reliability, we have to understand the characteristics of each.

Therefore, we will introduce the three provided consensus mechanisms of Hyperledger Fabric and explore their positives and negatives:

- **Kafka**, is a Crash Fault-Tolerant (CFT), decentralized consensus mechanism that uses the distributed streaming platform "Apache Kafka" as a message broker [Spydra, 2023, Lücking et al., 2020, Tripathi, 2023]. This mechanism works by replicating transactions across multiple computers (nodes) and guaranteeing devotion to the same order by all participant nodes [Spydra, 2023].

- **Raft**, is a CFT, leader-based consensus mechanism [Lücking et al., 2020, Hyperledger, 2023b]. It uses the Raft protocol, which elects a leader per channel [Hyperledger, 2023b]. The leader node receives all transactions and replicates them to all follower nodes [Hyperledger, 2023b]. This relationship ensures a distributed and fault-tolerant system [Lücking et al., 2020].

- **Solo**, operates as a single-node consensus mechanism where a single orderer node sends new blocks to all peer nodes [Lücking et al., 2020, Tripathi, 2023]. It's primarily used for testing and development because of limitations [Hyperledger, 2023b, Tripathi, 2023]. Solo is the fastest of all built-in mechanisms in Hyperledger Fabric [Lücking et al., 2020].

As mentioned, choosing a consensus mechanism depends on the solution we are looking for. We intended a DApp, balancing safety, scalability, and simplicity. After studying the three offered consensus mechanisms, we'll explain our reasons for accepting or discarding them.

Starting with Solo, we didn't want to use this one because it did not fulfill our intentions. It is a centralized solution, and its security has some limitations - it's not CFT. Also, in v2.x it's deprecated [Hyperledger, 2023b].

Passing to Kafka, even if it is a CFT, decentralized solution and Hyperledger Fabric recommended it in earlier stages, it has a complex configuration and architecture. Also, in v2.x it's deprecated [Hyperledger, 2023b].

Raft is the last consensus mechanism and our choice. This mechanism is the recommendation from Hyperledger Fabric [Hyperledger, 2023b], provides a decentralized solution, is CFT, and balances the three "S's" - scalability, security, and simplicity.

6.2.2 Smart Contract (Chaincode)

Hyperledger Fabric offers the possibility to create smart contracts in various programming languages. So, to develop the platform all in the same language, we chose JavaScript. Our blockchain needs to allow the creation of activities and lots, the update of activities, and the retrieval of all this data.

Create Data

To create an activity or a lot, we use the "writeData" function (Listing 6.1). This method receives as parameters the context, the key (serves as ID for the blockchain) from the transaction, and a JSON with all the information. The key is sequential and starts with "LOT" if it's a lot (e.g., LOT20) or "ACT" if it's an activity (e.g., ACT76).

```
const { Contract } = require('fabric-contract-api')

class FreshTracker extends Contract {
    async initLedger(ctx) { }

    async writeData(ctx, key, value) {
        await ctx.stub.putState(key, value);

        return value;
    }
```

Listing 6.1: Smart Contract - Function to Create Data

Retrieve All Data

To retrieve all the information about lots or activities, we use the ”readData” function (Listing 6.2). This method receives two parameters: the context and a field used to choose activities or lots. If the value is A, it will retrieve all activities, if not all lots. After the condition is verified to collect all records, an iterator and loops are used.

```
    async readData(ctx, entity) {
        const table = (entity === 'A') ? 'ACT' : 'LOT';

        const startKey = `${table}0`;
        const endKey = `${table}999`;

        const iterator = await ctx.stub.getStateByRange(startKey, endKey);
        const allResults = [];

        while (true) {
            const res = await iterator.next();

            if (res.value && res.value.value.toString()) {
                const Key = res.value.key;
                let Record;

                try {
```

```
                    Record = JSON.parse(res.value.value.toString('utf8'
                        ));
                } catch (err) {
                    Record = res.value.value.toString('utf8');
                }
                allResults.push({ Key, Record });
            }
            if (res.done) {
                await iterator.close();

                return JSON.stringify(allResults);
            }
        }
    }
```

Listing 6.2: Smart Contract - Function to Retrieve All Data

Retrieve a Single Record

To retrieve a single record, we use the "readSingle" function (Listing 6.3). This method receives as parameters the context and the key. The latter obtains the specific record.

```
    async readSingle(ctx, key) {
        let response = await ctx.stub.getState(key);
        return response.toString();
    }
```

Listing 6.3: Smart Contract - Function to Retrieve a Single Record

Update Data

To update activities, we use the "updateData" function (Listing 6.4). This method receives as parameters the context, the key, a field to control which fields to update, and the data. The updatable fields of the activity are its status, the end timestamp, and the output lots for when the activity finishes.

```
    async updateData(ctx, key, field, data) {
        const asBytes = await ctx.stub.getState(key);
```

```

        if (!asBytes || asBytes.length === 0) {
            throw new Error(`${key} does not exist`);
        }
        const activity = JSON.parse(asBytes.toString());

        if (field === 'endedAt') { activity.endedAt = data; }
        else if (field === 'outLots') { activity.outLots = data; }
        else if (field === 'status') { activity.status = data; }
        else { console.log('Error'); }

        await ctx.stub.putState(key, Buffer.from(JSON.stringify(
            activity)));
    }
}

module.exports = FreshTracker
```

Listing 6.4: Smart Contract - Function to Update Data

6.3 MongoDB

One of the features that most influenced the choice of the database was the scalability and ease of constant future adaptations. With this in mind, we went to NoSQL DBs. These databases have positives and negatives but share their roots: handling high amounts of data at high speeds, easy updates to fields and schemas, storing unstructured, semi-structured, or structured data, and others [MongoDB, 2023a]. So, with these characteristics, we found what we were looking for. After looking at different options, we chose to use MongoDB. This DB provides scalability and flexibility, enabling easy schema evolution [MongoDB, 2023a]. To work with MongoDB, we opted to use Mongoose. This tool is an ODM for Node.js and provides a schema-based solution to model application data [Mongoose, 2023]. We chose to work with Mongoose because it enables the creation of familiar data structures, and it's for Node.js, so we don't need to work directly with the database.

Next, we'll show how we structured the backend using MongoDB, Mongoose, and

Node.js.

6.3.1 Structure

To develop the backend part of the blockchain using MongoDB, Mongoose, and Node.js, a Model-View-Controller (MVC) chimera was used, where we have the "Model" that defines the schemas of all entities (tables) in our database and the "Controller" that implements the API's endpoints and routes them. Fig. 6.2 illustrates the "Model" and an example, and Fig. 6.3 shows the "Controller" and an example.

Model & Controller

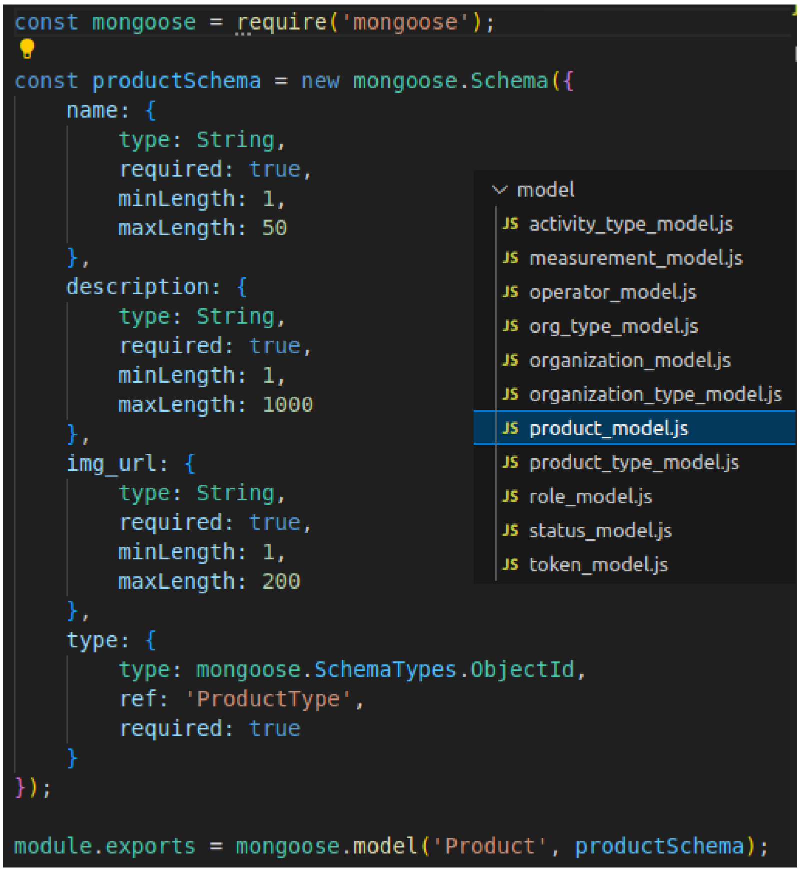

Figure 6.2: Backends Model Files and Example.

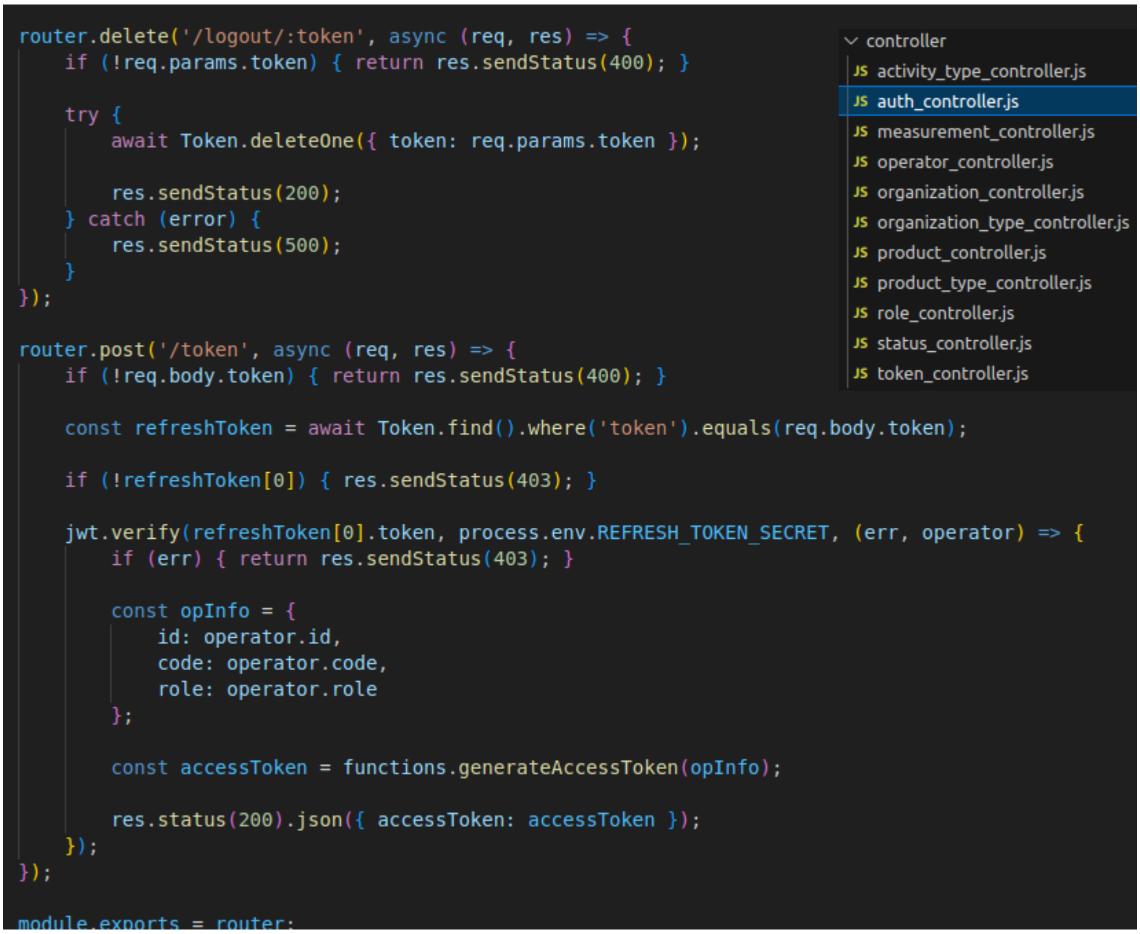

Figure 6.3: Backends Controller Files and Example.

6.4 Automatic Data Collection

Automation has reached every industry. In [Sure Controls, 2013], industrial automation definition is the use of control systems (i.e., computers and robots) and information technologies to handle different processes. With the help of technologies such as the IoT and AI, it's possible to speed up the information flow of supply chains, enabling real-time monitoring, helping with decision-making, and others [Haleem et al., 2021, Lou et al., 2011].

In the following sections, we'll demonstrate how we implemented some automation in our traceability platform.

6.4.1 QR Code

Our first form of automation is using QR Codes. Scanning this tool is a quick way to obtain or update information. For example, when an operator starts or finishes an activity, scanning the QR Codes on the lots is a quick way to register them in the platform. Typing the ID should be a last-resort situation (e.g., damaged QR Code). The information contained in these codes is the key to the lot or activity (also visible beneath the QR Code) for the blockchain and the UUID given to every lot and activity. Fig. 6.4 shows an example of a QR Code.

LOT256

Figure 6.4: Fruit and Vegetable Traceability Platform Lot QR Code Example.

6.4.2 Measurements

In the platform, for the automatic collection of quality and sustainability data, we simulate IoT devices using APIs since we do not physically own any devices. Fig. 6.5 illustrates the potential of IoT for monitoring, collecting, and saving measurements.

Next, we'll show the API used to get measurements and how the request works.

API

Open-Meteo [Open-Meteo, 2023] is a weather forecast API. It provides different metrics such as average, maximum, and minimum temperature, humidity, and soil and wind information, collecting these metrics automatically after an activity ends. Listing 6.5 shows the request to the API.

```
const getMeasurements = async (json) => {
    const openMeteo = `https://api.open-meteo.com/v1/forecast?`;
    const string = `latitude=${json.lat}&longitude=${json.lon}&${json.
        measurements}`;
```

```

    try {
        const res = await axios.get(`${openMeteo}${string}`);

        setMeasurements(res.data);
    }catch (error) {
        console.log(error);
    }
}
```

Listing 6.5: Measurements Automation API

The function "getMeasurements" will receive a JSON as a parameter. This JSON has three fields: latitude, longitude, and a string with all the wanted measurements. In the function, we have two contents, one with the base Uniform Resource Locator (URL) of Open-Meteo and the other one with all the parameters for the URL, completed with the data from the JSON. After, we join both parts of the URL and use Axios to make the request, finishing by storing the response in a variable.

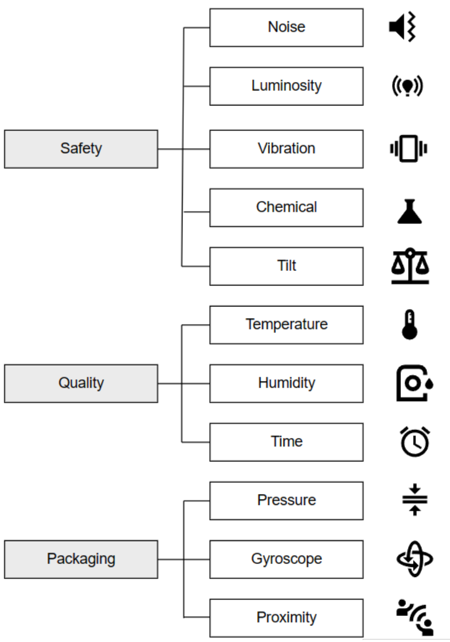

Figure 6.5: Example of Internet of Things Potential (adapted from [Hassoun et al., 2023]).

6.5 Security

In this section, we're detailing some of the security measures implemented in the application. We'll be starting by showing the tokens.

6.5.1 Tokens

As seen in Section 2.2.16, tokens allow applications to carry out the authentication and authorization process. We'll use this section to detail how the tokens are created and deleted and how we use them.

Creation and Deletion

After a successful login, two functions trigger to create an access token and a refresh token (Listing 6.6). Both of these functions use JWT.

```
require('dotenv').config({ path: __dirname + '/../../.env' });

module.exports = {
    generateAccessToken: function (operator) {
        const acessToken = jwt.sign(operator, process.env.ACCESS_TOKEN_SECRET, { expiresIn: '15m' });

        return acessToken;
    },
    generateRefreshToken: function (operator) {
        const refreshToken = jwt.sign(operator, process.env.REFRESH_TOKEN_SECRET);

        return refreshToken;
    }
};
```

Listing 6.6: Functions to Create Access and Refresh Tokens

When creating the access token, JWT uses data from the user who logged in, an environment variable (Listing 6.7) that contains the secret key previously defined by us, and the duration for which the token is valid. In our case, we have decided to set this duration to 15 minute (m).

```
ACCESS_TOKEN_SECRET=18
    c8927ae7254a975129be8c9665d708660332443ab709a9d693f2a81
882139d4461fa88ec0d9af156e11935d55887879b39dc8b477a3834da98d87b0ba79ce6
REFRESH_TOKEN_SECRET=
    e2ff290a9daec1faf9af727e57234f0844ed0bdfa5c1934b51181024
d57c29b8dad0c3401082731651a03f71a44d802b6ce8bcd02144053a81c69fb59c1713c6
```

Listing 6.7: Access and Refresh Tokens Secret

Similarly, when creating a refresh token, the process is the same as creating an access token, except for the expiration time. The refresh token is stored in our MongoDB database to enable the refreshing of the access token, i.e., requesting a new access token when the previous one expires.

Every time someone logs out, their refresh token gets deleted. Consequently, the access token becomes invalid.

Using an Access Token

Whenever a user needs to perform an action that requires authentication or authorization, such as saving an activity to the blockchain, they'll need to include the access token in the requests header (Listing 6.8).

```
const axiosJWT = axios.create({
    headers: {
        authorization: Bearer ${loginDetails.accessToken}
    }
});
```

Listing 6.8: Using an Access Token

Using a Refresh Token

In our platform, when the access token expires (after 15m), we use a refresh token to obtain a new one. To avoid having users refresh their access token manually, we automate this process using Axios.interceptors (Listing 6.9). This tool intercepts requests made using Axios. If an access token is invalid, a function triggers to get a new one. If the access token is valid, the request is allowed to proceed without interruption.

```
axiosJWT.interceptors.request.use(
    async (config) => {
        let currentDate = new Date();
        const decodedToken = jwt_decode(loginDetails.accessToken);

        if (decodedToken.exp * 1000 < currentDate.getTime()) {
            const data = await refreshToken();

            config.headers['authorization'] = Bearer
            ${data.accessToken};
        }

        return config;
    }, (error) => {
        return Promise.reject(error);
    }
);
```

Listing 6.9: Using A Refresh Token

6.5.2 Middleware

Our middleware (Listing 6.10) works together with our tokens. Its function is to verify whether an access token exists in the header, and if it does, it checks whether it is the correct one for the intended operation. The middleware is only needed when authentication and authorization verification are necessary. Additionally, it will send out different error messages based on the result.

```
require('dotenv').config({ path: __dirname + '/../../.env' });
const jwt = require('jsonwebtoken');

module.exports = {
    authenticateToken: function (req, res, next) {
```

```
        const authHeader = req.headers.authorization;

        if (authHeader) {
            const token = authHeader.split(' ')[1];

            jwt.verify(token, process.env.ACCESS_TOKEN_SECRET, (err, operator) => {
                if (err) { return res.status(403).json({ msg: 'Wrong Permission to Conclude Operation', action: 0 }); }
                req.operator = operator;

                next();
            });
        }
        else {
            res.status(401).json({ msg: 'No Permission to Conclude Operation', action: 0 });
        }
    }
};
```

Listing 6.10: Middleware

6.5.3 3-tier Verification

The platform has verifications at every layer, such as token verification, length validation, and missing field checks. Fig. 6.6 (red rectangles) shows null verification at the front, back, and storage layers.

6.6 Traceability Algorithm

We developed and tested different algorithms to see which was the fastest and most practical. One which we did several tests with was Trees - a JavaScript Data Structure. This algorithm ended up being more complex than using basic JS arrays.

Our traceability process has three parts. The first consists of receiving the data in bulk, the second is the traceability algorithm, and the third is mapping the array on the map.

Next, we'll go through the process, but only detailing our traceability algorithm and mentioning the basics of the other steps.

Figure 6.6: Example of Null Verification in the Three Layers (Red Rectangles).

6.6.1 Receiving Data in Bulk

When a user on the front end scans a lot ID to start the traceability, the API will retrieve all Activity records. As shown before (Section 5.2), the Activity table stores output and input lots. After receiving this bulk, we'll pass on to the next step.

6.6.2 Traceability Algorithm

The second step, still in the backend, is the traceability algorithm. After receiving the bulk data and the lot ID, the algorithm starts by finding the last recorded activity of the lot with the "getActivityId" function.

Now, with the activity ID, we can use the collection of input lots to keep going back using the recursive function - routeLot. To trace forward is the same process but using the output lots instead. Every time it goes back, it will save the activity in the mentioned array, and it only stops when the activity doesn't have input lots (i.e., represents a Harvest). Listing 6.11 illustrates the most crucial parts of the traceability algorithm.

```
function getActivityId(activityData, lotId) {
    for (let i = 0; i < activityData.length; i++) {
        if (activityData[i].outLot.length != 1) {
            for (let j = 0; j < activityData[i].outLot.length; j++) {
                if (activityData[i].outLot[j] === lotId) {
                    return activityData[i].id;
                }
            }
        }
        if (activityData[i].outLot[i] === lotId) {
            return activityData[i].id;
        }
    }
}

const key = getActivityId(activityData, lotId);

let trackArray = [{
    activityId: key,
    before: null
}];

function routeLot(activityData, activityId) {
    let found = activityData.filter(function (item) { return item.id === activityId; });

    if (found[0].pre[0]) {
        if (found[0].pre.length != 1) {
            for (let i = 0; i < found[0].pre.length; i++) {
                trackArray.push({
                    activityId: found[0].pre[i],
```

```
                    state: found
                });


                routeLot(activityData, found[0].pre[i]);

            }
        }
        else {
            trackArray.push({
                activityId: found[0].pre[0],
                state: found
            });

            let newID = found[0].pre[0];

            routeLot(activityData, newID);
        }
    }
}
```

Listing 6.11: Traceability Algorithm

6.6.3 Mapping the Array on the Map

The third and final step in this process happens in the front end. After receiving the tracked array, we need to use the ".map()" method and have a condition to deal with non-linear paths.

6.7 Mobile Application

The mobile app is designed for personal use, primarily allowing consumers to scan product QR Codes, view product details, and track lots while on the go.

Next, we're presenting some pages and sharing details about them.

6.7.1 Scanning Page

On this page, users can scan product QR Codes to see details or to track the product to its origin. These page uses React Native with some React libraries. Fig. 6.7 illustrates the development of this page and its final version.

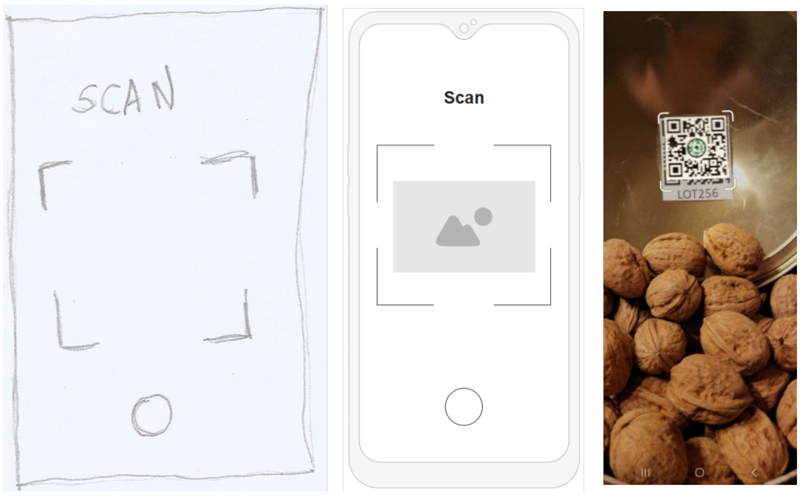

Figure 6.7: Mobile Application - Scanning Page

6.7.2 Map Page

On this page, users visualize the traceability of the chosen lot. Users also can click on any activity to see details. This page uses React Native and Leaflet to do the interactive map. Fig. 6.8 illustrates the development of this page and its final version.

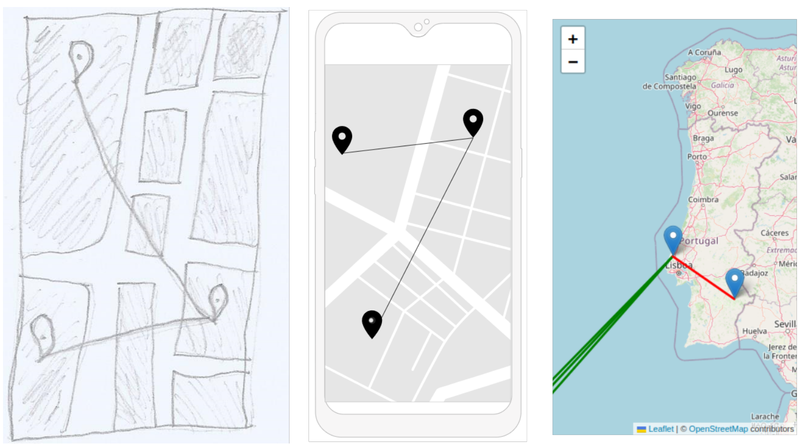

Figure 6.8: Mobile Application - Traceability Page

6.8 Web Application

The web app offers everything that the mobile one does, plus it's where the operators and platform administrator login to carry out their activities. All actors in the F&V value chain can use this app. Next, we will show some of the developed pages per user type.

6.8.1 All Users

Home Page

The home page (Fig. 6.9) is the first page any unlogged user encounters. From here, they can go to the pages that allow them to log in, trace lots, and view details about lots and products. The development of this page was with React and React Bootstrap.

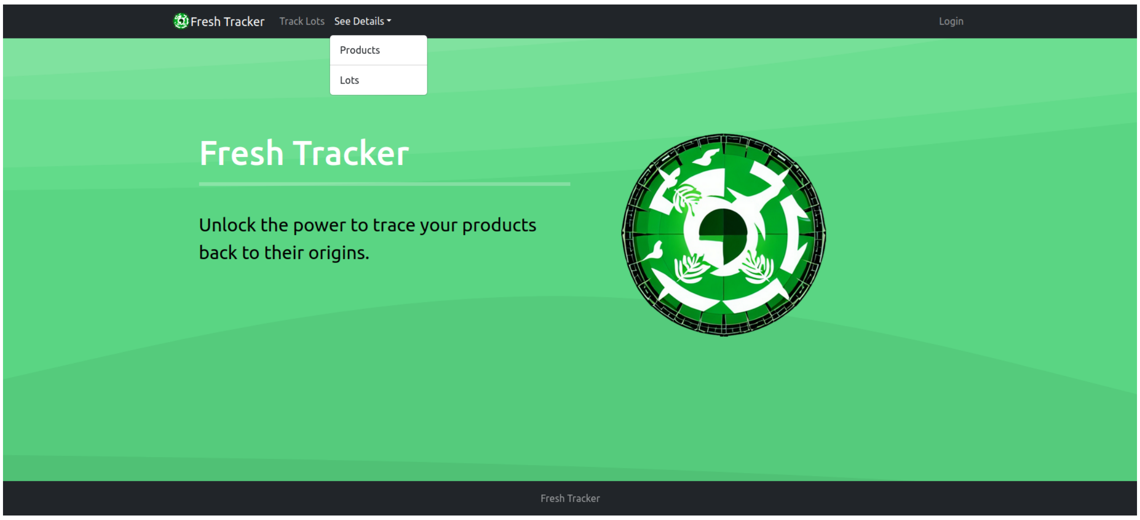

Figure 6.9: Web Application - Home Page

Product Details Page

On the product details page (Fig. 6.10), users can see details about any existing product in the traceability platform. They only need to choose from the list of available products or start typing the wanted product that the field filters by user input in real-time. After seeing the desired products, the user can only go to the home page by clicking the back arrow or the logo. The design of this page was with React and React Bootstrap. Axios is used to make the requests to the backend.

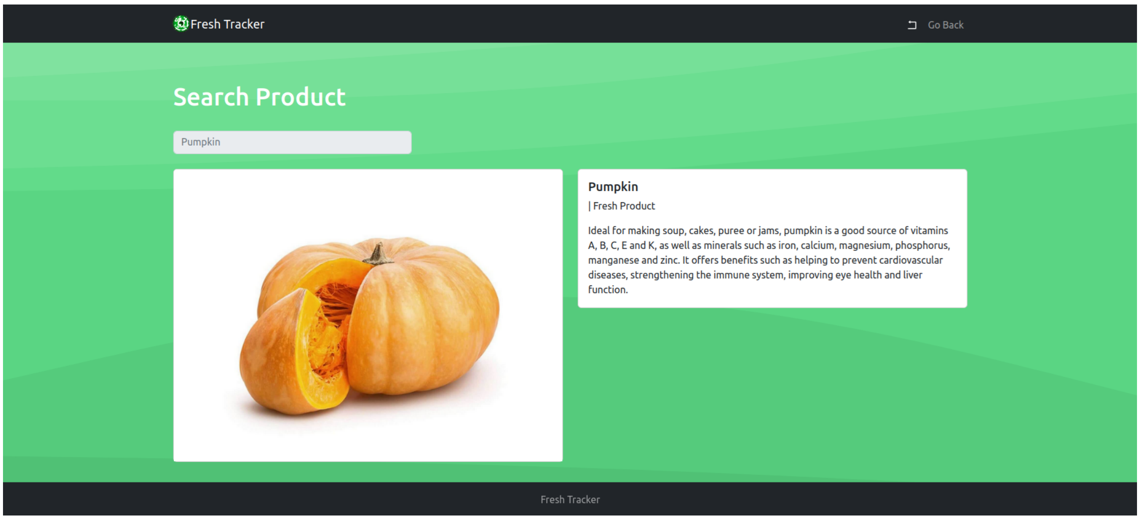

Figure 6.10: Web Application - Product Details Page

Activity Details Page

On the activity details page (Fig. 6.11 and Fig. 6.12), users can see all the details of a specific activity, such as the location, type of activity, timestamps, product, input and output lots, and the quality and sustainability measures collected and saved from that activity. Some information is a link to see it in detail. The design of this page was with React and React Bootstrap. Axios is used to make the requests to the backend, and Leaflet to create the map.

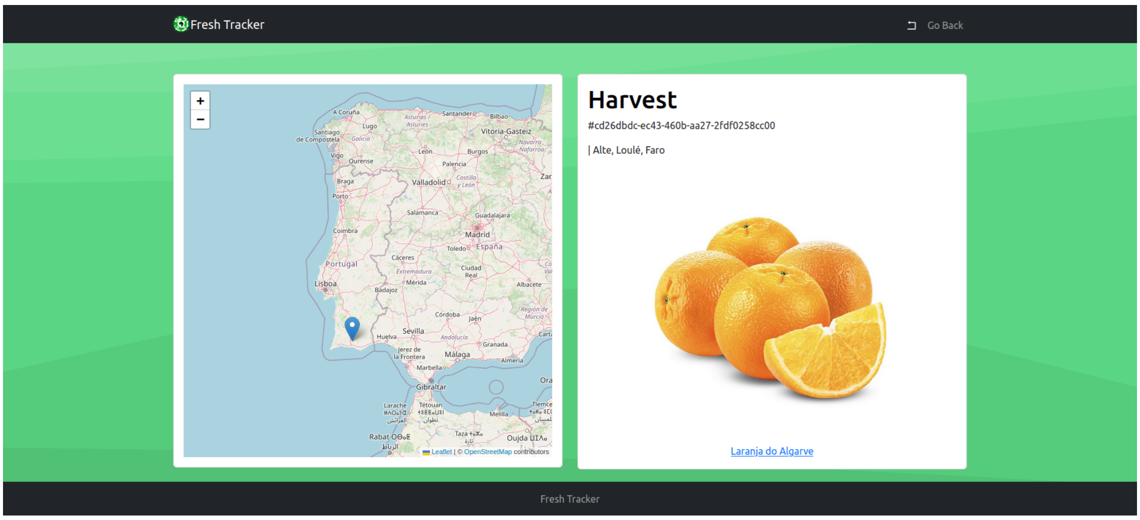

Figure 6.11: Web Application - Activity Details Page 1

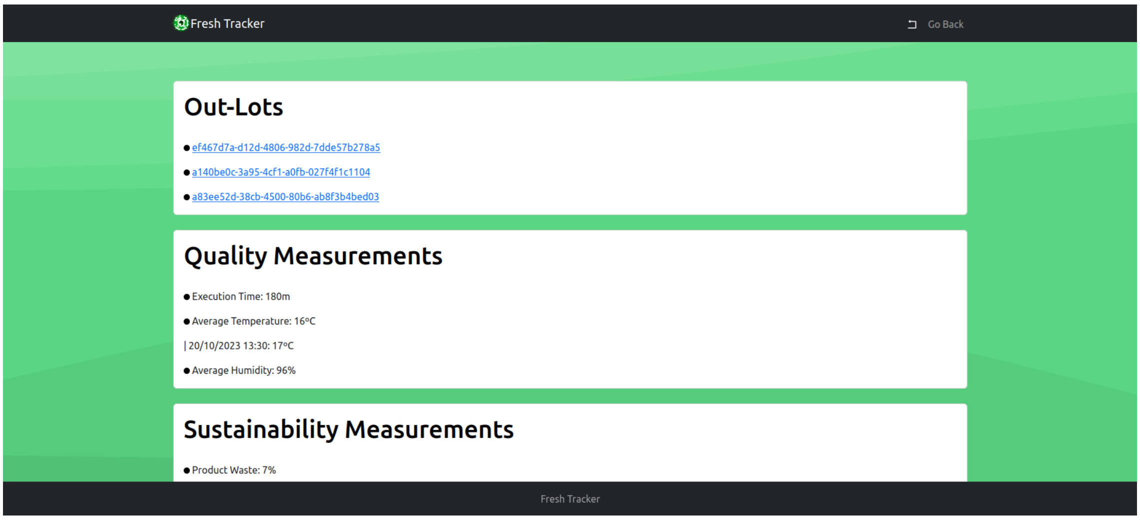

Figure 6.12: Web Application - Activity Details Page 2

Traceability Page

On the traceability page (Fig. 6.13), we can see an example of backward traceability. Users can scan the QR Code or input the ID manually to start tracking the respective lot. The design of this page was with React and React Bootstrap. Axios is used to make the requests to the backend, and Leaflet to create the map.

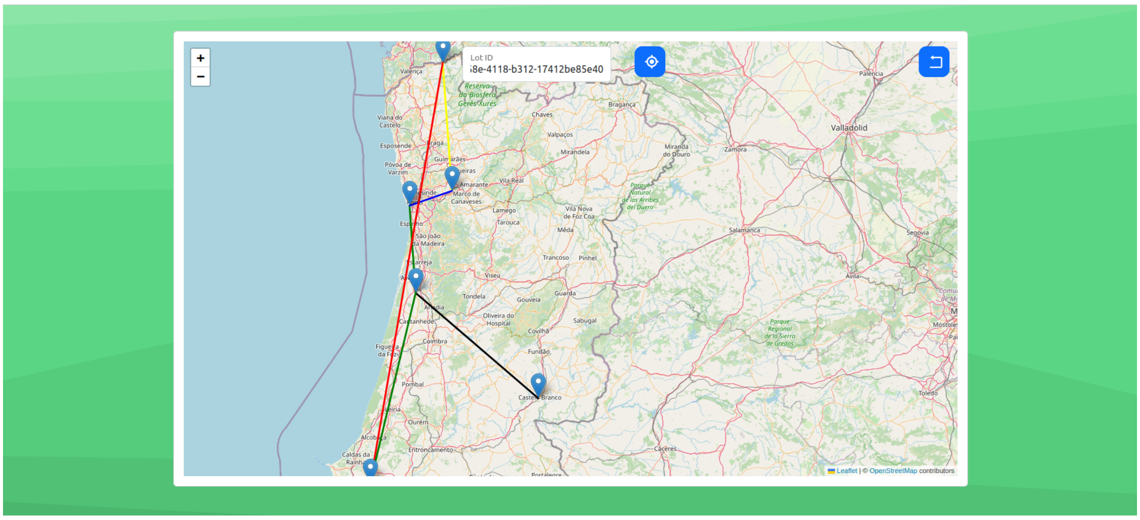

Figure 6.13: Web Application - Traceability Page

6.8.2 Administrator Operator

Manage Operators Page

On the manage operators page, the administrator operator can see every operator that works in his organization. He can see the number of active activities they have (clicking shows details) if they are logged in and an option to deny their access and functions on the traceability platform. The design of this page was with React and React Bootstrap. Axios is used to make the requests to the backend.

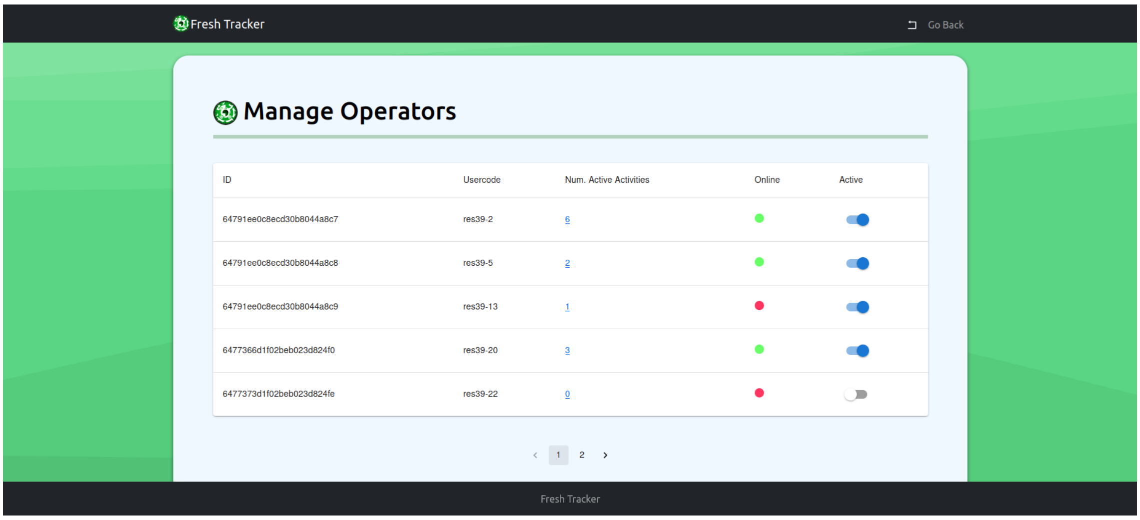

Figure 6.14: Web Application - Manage Operators Page

6.8.3 Normal Operator

Register Activity

The Register Activity Page (Fig. 6.15) is very dynamic, i.e., most fields can be auto-completed depending on the activity chosen and by scanning the QR Code. The operator starts by selecting the type of activity by clicking the respective field and choosing from the options or by typing the name in the respective field, which will filter the activities. When scanning the QR Code, the "Product" field is auto-completed, and manual selection is only necessary if there are no input lots (e.g., Harvest). The final fields to start an activity are the origin and destination organizations. The latter can be auto-filled with a switch button if the origin and destination are equal, and if they differ, then it has to be chosen manually. The design of this page was with React, React Bootstrap, and other React libraries.

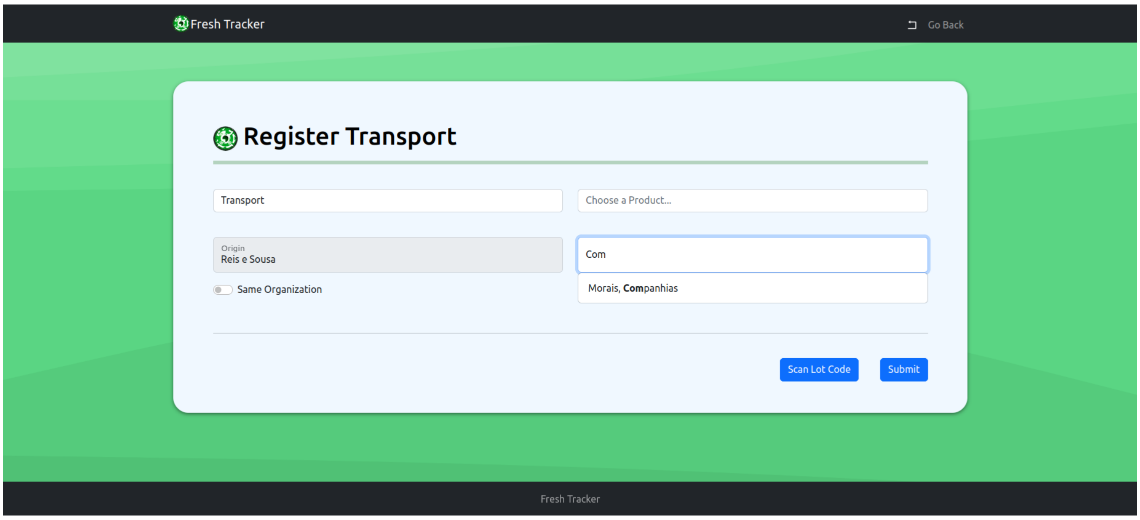

Figure 6.15: Web Application - Register Activity Page

6.8.4 Platform Administrator

All pages for operator administrators are either forms or statistics. We are not highlighting any specific page as we have already shown similar ones.

6.9 Tests

This section presents some of the tests realized to our traceability platform. The first was to see if our platform is compatible with all the major used web browsers. The second was to see the performance of our storage system, more specifically, comparing both methods. The third and final was a usability test.

6.9.1 Compatability

Different users can use various web browsers, so we wanted to do a compatibility test to see if our traceability platform was compatible with, at least, the most used web browser. The test subjects were Google Chrome, Microsoft Edge, Mozilla Firefox, Opera, and Safari. After using all web browsers, we can affirm that the traceability platform worked with all of them. Working with technologies that are compatible with most web browsers helped achieve this.

6.9.2 Usability

To perform usability tests, we invited six people to trial our traceability platform. The group was divided (into pairs of 2) based on their expertise with computer systems:

- **High** is someone who worked or works with computer systems.
- **Medium** is someone who doesn't work with computer systems but has general experience with them.
- **Low** is someone with a low interaction with computer systems.

After that, the testees, who never saw the platform, were asked to perform various crucial activities. The results include speed in concluding tasks, difficulty finding the next step, suggestions, and more. These results were collected, analyzed, and utilized to better the platform.

6.9.3 Performance

With the performance tests, we wanted to evaluate the create and read speed from MongoDB and Hyperledger Fabric to see the differences. Fig. 6.16 shows the results from registering and reading 1, 10, and 100 records. Based on the results, it is plain that the blockchain is significantly slower, with one record taking almost one second to be registered.

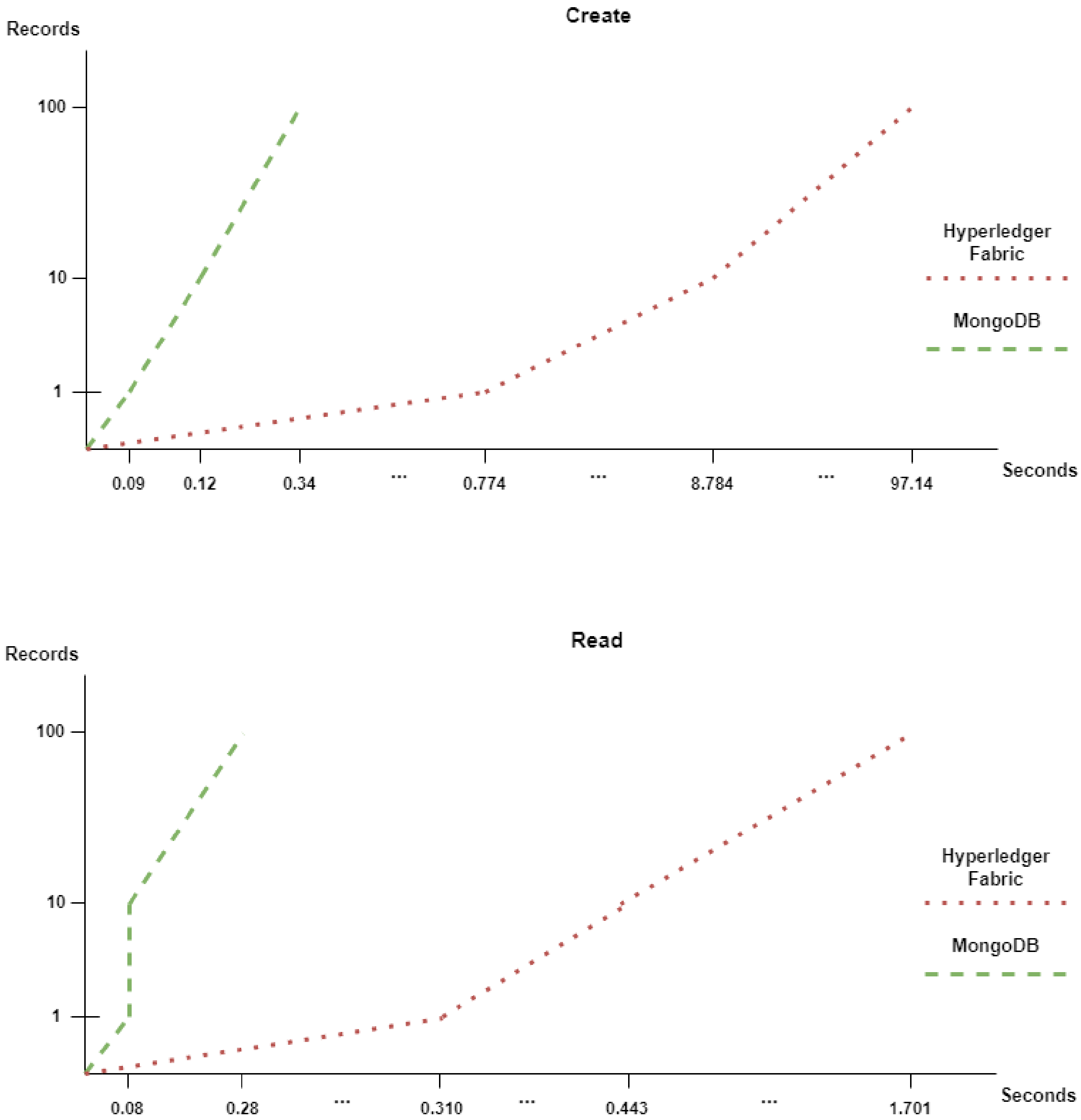

Figure 6.16: Performance Tests

6.10 Discussion

In this section, we will discuss the results obtained in this project, starting with the collecting and storing of quality and sustainability indicators. Because we didn't have the actual IoT hardware, we had to simulate these tools using third-party APIs. This method resulted in an exceedingly solid and efficient method to replicate what we pretended to do. In every step in the F&V value chain, we collected quality and sustainability indicators automatically and manually. These indicators were stored in the off-chain database (MongoDB) and were available for all actors to see on the activity pages of the mobile and web apps. With this, we can affirm that our traceability platform provides a way to decrease food fraud, increase food quality and transparency, and help walk towards sustainable development in the Fruit and Vegetable value chain.

Blockchain is the central technology of this project. We built around this tool because it offers immutability and high transparency. Knowing blockchain scalability issues, we opted for hybrid storage between blockchain and a regular database, hoping it would alleviate the stress on the blockchain chain.

For the regular database (off-chain), we ended up choosing a NoSQL database (MongoDB), which offered high scalability (to support the blockchain) and ease of constant future adaptations. These characteristics are essential when working in a value chain where new demands and regulations demand more information to be collected and stored. In this DB, we chose to keep all the information that is not crucial for the immutability of the traceability.

For the blockchain (on-chain), we ended up choosing Hyperledger Fabric, which offered a balance between scalability and security and optimization for supply chain management. In this database, we chose to keep all the information that is necessary to make the traceability immutable.

The hybrid storage system became extremely useful because of the results obtained from the performance test (Fig. 6.16). This test proves the enormous speed differences between the on-chain and off-chain databases. Ending up saving only required data for immutability in the blockchain is crucial for the long-term use of this technology.

With blockchain, we achieved decentralization and immutable records. This way, no one uniquely controls the information, giving our platform the ability to provide to consumers and every other actor in the F&V value chain transparency and authenticity.

Chapter 7

Conclusions

This research aimed to design and implement a traceability platform for the fruit and vegetable value chain and to collect and store quality and sustainability measurements. Based on the results of this project, we can confirm that we achieved all outlined objectives and more. A blockchain-based traceability system for the fruit and vegetable value chain, using a "From Farm to Fork" process, was implemented.

Blockchain was the central technology because of its decentralized and immutable nature, which provides a transparent and secure platform for recording and tracking information throughout the entire fruit and vegetable value chain. With these characteristics, blockchain supplies a method to improve traceability and transparency, allowing actors to confirm the quality, origin, and sustainability exercises applied in the F&V value chain through access to tamper-proof, real-time quality, and sustainability indicators data.

To mitigate blockchain's known scalability problem, we used a "Blockchain + Database" storage system.

Although blockchain seems the perfect solution, it has its share of hindrances. Compared to the regular database that we used, it's very slow at creating new records and starts to get slower at retrieving records as it scales. We opted to retrieve the records in bulk and perform the filtration in the backend instead of querying the blockchain because of speed issues. Implementing blockchain in the fruit and vegetable value chain requires investment in infrastructure and technological capabilities because of scalability and other issues.

In conclusion, we believe this project can help to start the standardization of transparency and immutability in the fruit and vegetable value chain and other food-related chains. With our traceability platform, we provide a way to enhance food quality, reduce food fraud and waste, comply with the regulations and demands of consumers, and work towards sustainable development in the food industry.

The project presented in this dissertation originated a publication [Morais et al., 2023] in the 2023 18th Iberian Conference on Information Systems and Technologies (CISTI).

7.1 Future Work

As we worked in a small simulation environment, some aspects of the traceability platform weren't polished.

In terms of design, the UI in both apps deserves a review from a designer to comply with modern standards. The map could have icons depending on the activity type.

In more technical aspects, the neutral and only available language in the traceability platform is English, and in the future, we would like to implement others. There is room to argue that the quality and sustainability indicators should be stored in blockchain. But, more detailed analysis and testing need implementation because of the blockchain issues, specifically scalability. The utilization of this traceability platform in a realistic scenario, utilizing genuine IoT tools, would be advantageous in comprehending more possible problems.

References

[ISO, 2016] (2016). Iso/iec/ieee international standard for systems and software engineering – life cycle management – part 4: Systems engineering planning. *ISO/IEC/IEEE 24748-4 First edition 2016-05-15*, pages 1–73.

[Agrawal et al., 2021] Agrawal, T. K., Kumar, V., Pal, R., Wang, L., and Chen, Y. (2021). Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry. *Computers & Industrial Engineering*, 154:107130.

[Akhtar and Rizvi, 2021] Akhtar, M. M. and Rizvi, D. R. (2021). *Traceability and Detection of Counterfeit Medicines in Pharmaceutical Supply Chain Using Blockchain-Based Architectures*, pages 1–31. Springer International Publishing, Cham.

[Alexandrea, 2023] Alexandrea, J. (2023). What is bootstrap? *Hostinger*. `https://www.hostinger.com/tutorials/what-is-bootstrap/`.

[Alexandrea and Muhammad, 2023] Alexandrea, J. and Muhammad, A. (2023). What is react: Understanding the features and how to deploy for modern web development. *Hostinger*. `https://www.hostinger.com/tutorials/what-is-react`.

[Arias and Bellen, 2021] Arias, D. and Bellen, S. (2021). What are refresh tokens and how to use them securely. *Auth0*. `https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/`.

[Auth0, 2023a] Auth0 (2023a). Access tokens. `https://auth0.com/docs/secure/tokens/access-tokens`.

[Auth0, 2023b] Auth0 (2023b). Refresh tokens. `https://auth0.com/docs/secure/tokens/refresh-tokens`.

[Axios, 2023] Axios (2023). Getting started. `https://axios-http.com/docs/intro`.

[Baralla et al., 2019] Baralla, G., Pinna, A., and Corrias, G. (2019). Ensure traceability in european food supply chain by using a blockchain system. In *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pages 40–47.

[Baskerville et al., 2015] Baskerville, R. L., Kaul, M., and Storey, V. C. (2015). Genres of inquiry in design-science research: Justification and evaluation of knowledge production. *MIS Quarterly*, 39(3):541–564.

[Ben Ayed et al., 2022] Ben Ayed, R., Hanana, M., Ercisli, S., Karunakaran, R., Rebai, A., and Moreau, F. (2022). Integration of innovative technologies in the agri-food sector: The fundamentals and practical case of dna-based traceability of olives from fruit to oil. *Plants*, 11(9).

[Bevilacqua et al., 2009] Bevilacqua, M., Ciarapica, F., and Giacchetta, G. (2009). Business process reengineering of a supply chain and a traceability system: A case study. *Journal of Food Engineering*, 93(1):13–22.

[Bootstrap, 2023] Bootstrap (2023). `https://getbootstrap.com/`.

[Ciruela-Lorenzo et al., 2020] Ciruela-Lorenzo, A. M., Del-Aguila-Obra, A. R., Padilla-Meléndez, A., and Plaza-Angulo, J. J. (2020). Digitalization of agri-cooperatives in the smart agriculture context. proposal of a digital diagnosis tool. *Sustainability*, 12(4).

[Codecademy Team, 2023] Codecademy Team (2023). What is rest? *Codecademy*. `https://www.codecademy.com/article/what-is-rest`.

[Coelho et al., 2021] Coelho, E. M., da Silva Haas, I. C., de Azevedo, L. C., Bastos, D. C., Fedrigo, I. M. T., dos Santos Lima, M., and de Mello Castanho Amboni, R. D. (2021). Multivariate chemometric analysis for the evaluation of 22 citrus fruits growing in brazil's semi-arid region. *Journal of Food Composition and Analysis*, 101:103964.

[Conti, 2020] Conti, M. (2020). Food traceability in fruit and vegetables supply chain. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.

[Cruz and da Cruz, 2020a] Cruz, E. F. and da Cruz, A. M. R. (2020a). Design science research for is/it projects: Focus on digital transformation. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.

[Cruz and da Cruz, 2020b] Cruz, E. F. and da Cruz, A. M. R. (2020b). Using blockchain to implement traceability on fishery value chain. In *Proceedings of the 15th International Conference on Software Technologies (ICSOFT 2020)*, pages 501–508.

[da Cruz and Cruz, 2020] da Cruz, A. M. R. and Cruz, E. F. (2020). Blockchain-based traceability platforms as a tool for sustainability. In *ICEIS (2)*, pages 330–337.

[da Paixão, 2018] da Paixão, J. R. (2018). O que é uuid? porque usá-lo? *Medium*. https://medium.com/trainingcenter/o-que-%C3%A9-uuid-porque-us%C3%A1-lo-ad7a66644a2b.

[Deloitte, 2013] Deloitte (2013). The food value chain a challenge for the next century. https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Consumer-Business/dttl_cb_Food%20Value%20Chain_Global%20POV.pdf.

[Donaghy et al., 2021] Donaghy, J. A., Danyluk, M. D., Ross, T., Krishna, B., and Farber, J. (2021). Big data impacting dynamic food safety risk management in the food chain. *Frontiers in Microbiology*, 12.

[Dutta et al., 2020] Dutta, P., Choi, T.-M., Somani, S., and Butala, R. (2020). Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transportation Research Part E: Logistics and Transportation Review*, 142:102067.

[Eat Happy Project, 2015] Eat Happy Project (2015). Ancient oil: from farm to fork. *YouTube*. [Video] https://www.youtube.com/watch?v=PHwjNDC2IYc&list=PLbPWPsvL8htmEsYS9-6m4vX8M30NfNUXJ&index=9.

[European Food Safety Authority, 2012] European Food Safety Authority (2012). E.coli: Rapid response in a crisis. https://web.archive.org/web/20181120064044/http://www.efsa.europa.eu/en/press/news/120711.

[European Union, 2002] European Union (2002). Regulation (ec) no 178/2002 of the european parliament and of the council of 28 january 2002 laying down the general principles and requirements of food law, establishing the european food safety authority and laying down procedures in matters of food safety. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex%3A32002R0178.

[European Union, 2019] European Union (2019). Regulation (eu) 2019/1381 of the european parliament and of the council of 20 june 2019 on the transparency and sustainability of the eu risk assessment in the food chain and amending regulations (ec) no 178/2002, (ec) no 1829/2003, (ec) no 1831/2003, (ec) no 2065/2003, (ec) no 1935/2004, (ec) no 1331/2008, (ec) no 1107/2009, (eu) 2015/2283 and directive 2001/18/ec (text with eea relevance.). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019R1381.

[Express, 2023] Express (2023). https://expressjs.com/.

[Fernandes et al., 2022] Fernandes, M. A., Cruz, E. F., and Rosado Da Cruz, A. M. (2022). Smart contract and web dapp for traceability in the olive oil production chain. In *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.

[Food and Agriculture Organization of the United Nations, 2018] Food and Agriculture Organization of the United Nations (2018). Sustainable food systems: Concept and framework. https://www.fao.org/3/ca2079en/CA2079EN.pdf.

[Food Standards Agency, 2002] Food Standards Agency (2002). Traceability in the food chain. https://www.adiveter.com/ftp_public/articulo361.pdf.

[Gasparik, 2023] Gasparik, M. (2023). Understanding uuid: Purpose and benefits of a universal unique identifier. *Medium*. https://medium.com/@gaspm/understanding-uuid-purpose-and-benefits-of-a-universal-unique-identifier-59110154d897.

[Gereffi and Fernandez-Stark, 2016] Gereffi, G. and Fernandez-Stark, K. (2016). Global value chain analysis: A primer, 2nd edition. https://dukespace.lib.duke.edu/dspace/bitstream/handle/10161/12488/2016-07-28_GVC%20Primer%202016_2nd%20edition.pdf.

[Grigutytė, 2023] Grigutytė, M. (2023). What is bcrypt and how does it work? *NordVPN*. https://nordvpn.com/blog/what-is-bcrypt/.

[Haleem et al., 2021] Haleem, A., Javaid, M., Singh, R. P., Rab, S., and Suman, R. (2021). Hyperautomation for the enhancement of automation in industries. *Sensors International*, 2:100124.

[Hall, 2022] Hall, J. (2022). Getting started with mongodb & mongoose. *MongoDB*. https://www.mongodb.com/developer/languages/javascript/getting-started-with-mongodb-and-mongoose/.

[Hassoun et al., 2023] Hassoun, A., Kamiloglu, S., Garcia-Garcia, G., Parra-López, C., Trollman, H., Jagtap, S., Aadil, R. M., and Esatbeyoglu, T. (2023). Implementation of relevant fourth industrial revolution innovations across the supply chain of fruits and vegetables: A short update on traceability 4.0. *Food Chemistry*, 409:135303.

[Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

[Hou et al., 2020] Hou, D., Bolan, N. S., Tsang, D. C., Kirkham, M. B., and O'Connor, D. (2020). Sustainable soil use and management: An interdisciplinary and systematic approach. *Science of The Total Environment*, 729:138961.

[Hyperledger, 2023a] Hyperledger (2023a). Introduction. https://hyperledger-fabric.readthedocs.io/en/release-2.5/whatis.html.

[Hyperledger, 2023b] Hyperledger (2023b). The ordering service. https://hyperledger-fabric.readthedocs.io/en/latest/orderer/ordering_service.html.

[Hyperledger Foundation, 2018] Hyperledger Foundation (2018). Hyperledger fabric explainer video. *YouTube*. [Video] https://www.youtube.com/watch?v=1ORrdusUzeg.

[IBM, 2023a] IBM (2023a). Json web token (jwt). https://www.ibm.com/docs/en/cics-ts/6.1?topic=cics-json-web-token-jwt.

[IBM, 2023b] IBM (2023b). What are smart contracts on blockchain? https://www.ibm.com/topics/smart-contracts.

[IBM, 2023c] IBM (2023c). What is a rest api? https://www.ibm.com/topics/rest-apis.

[IBM, 2023d] IBM (2023d). What is an api? https://www.ibm.com/topics/api.

[IBM, 2023e] IBM (2023e). What is blockchain technology? https://www.ibm.com/topics/blockchain.

[Jiang et al., 2021] Jiang, H., Li, X., and Safara, F. (2021). Withdrawn: Iot-based agriculture: Deep learning in detecting apple fruit diseases. *Microprocessors and Microsystems*, page 104321.

[Jungen et al., 2021] Jungen, M., Lotz, P., Patz, C.-D., Steingass, C. B., and Schweiggert, R. (2021). Coumarins, psoralens, and quantitative 1h-nmr spectroscopy for authentication of lemon (citrus limon [l.] burm.f.) and persian lime (citrus × latifolia [yu.tanaka] tanaka) juices. *Food Chemistry*, 359:129804.

[JWT, 2023] JWT (2023). Introduction to json web tokens. https://jwt.io/introduction.

[Kamble et al., 2020] Kamble, S. S., Gunasekaran, A., and Gawankar, S. A. (2020). Achieving sustainable performance in a data-driven agriculture supply chain: A review for research and applications. *International Journal of Production Economics*, 219:179–194.

[Kamilaris et al., 2018] Kamilaris, A., Antón, A., Bonmatí, A., and Prenafeta Boldú, F. (2018). Assessing and mitigating the impact of livestock agriculture on the environment through geospatial and big data analysis. *International Journal of Sustainable Agricultural Management and Informatics*, 4:98.

[Keerthivasan et al., 2022] Keerthivasan, E., Raja Vignesh, K., Shinthan, D., Naveen Kumaar, S., Subha, R., and Suchithra, M. (2022). A trusted blockchain based traceability system for fruits and vegetable agricultural products. In *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 2041–2046.

[Kolipaka, 2020] Kolipaka, V. R. R. (2020). Predictive analytics using cross media features in precision farming. *International Journal of Speech Technology*, 23(1):57–69.

[Leaflet, 2023] Leaflet (2023). `https://leafletjs.com/`.

[Li et al., 2021] Li, A., Zhao, J., Xi, J., Yang, X., Jin, X., Chen, Q., and Pan, L. (2021). Geographical authentication of peach in china based on stable isotope combined with multielement analysis of peach juice. *Food Control*, 127:108126.

[Lin and Liao, 2017] Lin, I.-C. and Liao, T.-C. (2017). A survey of blockchain security issues and challenges. *International Journal of Network Security*, 19(5):653–659.

[Lou et al., 2011] Lou, P., Liu, Q., Zhou, Z., and Wang, H. (2011). Agile supply chain management over the internet of things. In *2011 International Conference on Management and Service Science*, pages 1–4.

[Lücking et al., 2020] Lücking, M., Kannengiesser, N., Kilgus, M., Riedel, T., Beigl, M., Sunyaev, A., and Stork, W. (2020). The merits of a decentralized pollution-monitoring system based on distributed ledger technology. *IEEE Access*, 8:1–17.

[MDN Web Docs, 2023] MDN Web Docs (2023). Javascript. `https://developer.mozilla.org/en-US/docs/Web/JavaScript`.

[Meta Open Source, 2021] Meta Open Source (2021). Eli5: React native. *YouTube*. [Video] `https://www.youtube.com/watch?v=wUDeLT6WXnQ`.

[Misra et al., 2022] Misra, N. N., Dixit, Y., Al-Mallahi, A., Bhullar, M. S., Upadhyay, R., and Martynenko, A. (2022). Iot, big data, and artificial intelligence in agriculture and food industry. *IEEE Internet of Things Journal*, 9(9):6305–6324.

[Mollenkamp, 2023] Mollenkamp, D. T. (2023). What is sustainability? how sustainabilities work, benefits, and example. `https://www.investopedia.com/terms/s/sustainability.asp`. Investopedia.

[MongoDB, 2023a] MongoDB (2023a). Advantages of nosql databases. `https://www.mongodb.com/nosql-explained/advantages`.

[MongoDB, 2023b] MongoDB (2023b). Json and bson. https://www.mongodb.com/json-and-bson.

[MongoDB, 2023c] MongoDB (2023c). Mern stack explained. https://www.mongodb.com/mern-stack.

[MongoDB, 2023d] MongoDB (2023d). Mongodb features. https://www.mongodb.com/features.

[Mongoose, 2023] Mongoose (2023). https://mongoosejs.com/.

[Morais et al., 2023] Morais, R., Da Cruz, A. M. R., and Cruz, E. F. (2023). Towards a blockchain-based traceability platform for the fruit and vegetables value chain. In *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.

[Muñoz-Redondo et al., 2022] Muñoz-Redondo, J., Bertoldi, D., Tonon, A., Ziller, L., Camin, F., and Moreno-Rojas, J. (2022). Multi-element and stable isotopes characterization of commercial avocado fruit (persea americana mill) with origin authentication purposes. *Food Control*, 137:108975.

[Neuman, 2011] Neuman, W. (2011). Deaths from cantaloupe listeria rise. *The New York Times.* https://www.nytimes.com/2011/09/28/business/deaths-from-cantaloupe-listeria-rises.html.

[Node.js, 2023] Node.js (2023). Introduction to node.js. https://nodejs.org/en/learn/getting-started/introduction-to-nodejs.

[Object Management Group, 2023] Object Management Group (2023). https://www.bpmn.org/.

[Open-Meteo, 2023] Open-Meteo (2023). https://open-meteo.com/.

[Othman et al., 2020] Othman, A., Norton, L., Finny, A. S., and Andreescu, S. (2020). Easy-to-use and inexpensive sensors for assessing the quality and traceability of cosmetic antioxidants. *Talanta*, 208:120473.

[Panebianco et al., 2022] Panebianco, S., Mazzoleni, P., Barone, G., Musumarra, A., Pellegriti, M., Pulvirenti, A., Scordino, A., and Cirvilleri, G. (2022). Feasibility study of tomato fruit characterization by fast xrf analysis for quality assessment and food traceability. *Food Chemistry*, 383:132364.

[Pei et al., 2022] Pei, Y., He, C., Liu, H., Shen, G., and Feng, J. (2022). Compositional analysis of four kinds of citrus fruits with an nmr-based method for understanding nutritional value and rational utilization: From pericarp to juice. *Molecules*, 27(8).

[Pera Rocha, 2023] Pera Rocha (2023). Do campo ao prato. `http://perarocha.pt/pera-rocha/#campo`.

[Porto Editora, 2010] Porto Editora (2010). Rastreabilidade. In *Dicionário da Lingua Portuguesa* (1st ed., p. 1341).

[Pramanick, 2023] Pramanick, S. (2023). 10 best backend web development frameworks for 2023. *Kelton*. `https://www.kellton.com/kellton-tech-blog/top-10-best-backend-web-development-frameworks`.

[Radulescu et al., 2021] Radulescu, C., Olteanu, R. L., Nicolescu, C. M., Bumbac, M., Buruleanu, L. C., and Holban, G. C. (2021). Vibrational spectroscopy combined with chemometrics as tool for discriminating organic vs. conventional culture systems for red grape extracts. *Foods*, 10(8).

[React, 2023] React (2023). `https://legacy.reactjs.org/`.

[React Native, 2023] React Native (2023). `https://reactnative.dev/`.

[Rimu, 2023] Rimu, S. (2023). Comparing prisma and mongoose. *LinkedIn*. `https://www.linkedin.com/pulse/comparing-prisma-mongoose-sumaiya-rimu`.

[Safdie, 2023] Safdie, S. (2023). Why is traceability essential to supply chain management? *Greenly*. `https://greenly.earth/en-gb/blog/company-guide/why-is-traceability-essential-to-supply-chain-management`.

[Salah et al., 2019] Salah, K., Nizamuddin, N., Jayaraman, R., and Omar, M. (2019). Blockchain-based soybean traceability in agricultural supply chain. *IEEE Access*, 7:73295–73305.

[Sgarbossa and Russo, 2017] Sgarbossa, F. and Russo, I. (2017). A proactive model in sustainable food supply chain: Insight from a case study. *International Journal of Production Economics*, 183:596–606. Closed Loop Supply Chain (CLSC): Economics, Modelling, Management and Control.

[Song et al., 2021] Song, L., Cao, B., Zhang, Y., Meinhardt, L. W., and Zhang, D. (2021). Mining single nucleotide polymorphism (snp) markers for accurate genotype identification and diversity analysis of chinese jujube (ziziphus jujuba mill.) germplasm. *Agronomy*, 11(11).

[Spydra, 2023] Spydra (2023). Exploring consensus mechanisms in hyperledger fabric: Raft, solo, and kafka. *LinkedIn*. `https://www.linkedin.com/pulse/exploring-consensus-mechanisms-hyperledger-fabric-raft-solo-kafka`.

[Sriramya and Karthika, 2015] Sriramya, P. and Karthika, R. (2015). Providing password security by salted password hashing using bcrypt algorithm. *ARPN Journal of Engineering and Applied Sciences*, 10:5551–5556.

[Sure Controls, 2013] Sure Controls (2013). What is industrial automation? `https://www.surecontrols.com/what-is-industrial-automation/`.

[Tamasi et al., 2021] Tamasi, G., Bonechi, C., Leone, G., Andreassi, M., Consumi, M., Sangiorgio, P., Verardi, A., Rossi, C., and Magnani, A. (2021). Varietal and geographical origin characterization of peaches and nectarines by combining analytical techniques and statistical approach. *Molecules*, 26(14).

[Tardi, 2023] Tardi, C. (2023). Value chain: Definition, model, analysis, and example. `https://www.investopedia.com/terms/v/valuechain.asp`. Investopedia.

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

[Tian, 2017] Tian, F. (2017). A supply chain traceability system for food safety based on haccp, blockchain & internet of things. In *2017 International Conference on Service Systems and Service Management*, pages 1–6.

[Tripathi, 2023] Tripathi, S. (2023). Hyperledger fabric consensus mechanisms: Exploring the options. *Spydra*. https://www.spydra.app/blog/hyperledger-fabric-consensus-mechanisms-exploring-the-options.

[Trovato et al., 2022] Trovato, E., Vento, F., Creti, D., Dugo, P., and Mondello, L. (2022). Elucidation of analytical–compositional fingerprinting of three different species of chili pepper by using headspace solid-phase microextraction coupled with gas chromatography–mass spectrometry analysis, and sensory profile evaluation. *Molecules*, 27(7).

[Verified Market Research, 2022] Verified Market Research (2022). Fresh fruits and vegetables market size and forecast. https://www.verifiedmarketresearch.com/product/fresh-fruits-and-vegetables-market/#:\%E2\%88\%BC:text=Fresh\%20Fruits\%20and\%20Vegetables\%20Market\%20Size\%20And\%20Forecast,5.14\%25\%20from\%202021\%20to\%202028.

[Wang et al., 2019] Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., and Kim, D. I. (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, 7:22328–22370.

[Whitworth, 2018] Whitworth, J. (2018). South africa declares end to largest ever listeria outbreak. *Food Safety News*. https://www.foodsafetynews.com/2018/09/south-africa-declares-end-to-largest-ever-listeria-outbreak/.

[World Commission on Environment and Development, 1987] World Commission on Environment and Development (1987). Report of the world commission on environment and development: Our common future. https://sustainabledevelopment.un.org/content/documents/5987our-common-future.pdf.

[Xie et al., 2022] Xie, J., Wan, C., Tolón Becerra, A., and Li, M. (2022). Streamlining traceability data generation in apple production using integral management with machine-to-machine connections. *Agronomy*, 12(4).

[Xu et al., 2021] Xu, C., Chen, K., Zuo, M., Liu, H., and Wu, Y. (2021). Urban fruit quality traceability model based on smart contract for internet of things. *Wireless Communications and Mobile Computing*, 2021:1–10.

[Yang et al., 2021] Yang, X., Li, M., Yu, H., Wang, M., Xu, D., and Sun, C. (2021). A trusted blockchain-based traceability system for fruit and vegetable agricultural products. *IEEE Access*, 9:36282–36293.

[Zahed Benisi et al., 2020] Zahed Benisi, N., Aminian, M., and Javadi, B. (2020). Blockchain-based decentralized storage networks: A survey. *Journal of Network and Computer Applications*, 162:102656.