Journal of Agriculture and Food Research 5 (2021) 100169

Contents lists available at ScienceDirect

Journal of Agriculture and Food Research

journal homepage: www.sciencedirect.com/journal/journal-of-agriculture-and-food-research

Traceability system for quality monitoring in the fishery and aquaculture value chain

José Oliveira [a], José Evaristo Lima [a], Dimitri da Silva [b], Volodymyr Kuprych [b], Pedro Miguel Faria [a], Cláudio Teixeira [b], Estrela Ferreira Cruz [a,c], António Miguel Rosado da Cruz [a,c,*]

[a] *Polytechnic Institute of Viana do Castelo, Viana do Castelo, Portugal*
[b] *University of Aveiro, Aveiro, Portugal*
[c] *Algoritmi Research Center, University of Minho, Guimarães, Portugal*

ARTICLE INFO

Keywords:
Food traceability
Software platform
Fish
Fishery
Aquaculture
Value chain

ABSTRACT

Nowadays it is important to know the origins of most products, not only for public health reasons, with regard to food products, but also because of the increasing people's awareness of environmental and social aspects, posing new challenges in the quality of products and their environmental impact. Food traceability information must be collected throughout the supply chain, and be registered on a platform external to all operators in that chain. This information must be available, not only to value chain operators and public health authorities, but also to end customers. One of the best ways to provide customers with this traceability information, in the places where they need to make a decision about which products to buy, is to add kiosks at the points of sale for those products. This article describes the design and architecture of a traceability platform for fish and fishery products, within the scope of the ValorMar R&D project, as well as the applications that supply the platform with traceability information, and a kiosk application for points of sale and a mobile app for final consumers, which will allow them to consult the traceability of fish and fishery products' lots.

1. Introduction

According to FAO's 2020 report "The State of World Fisheries and Aquaculture" [16], the production of fishery products peaked at about 179 million metric tons in 2018, including aquaculture. Of the overall total, 156 million metric tons were used for human consumption and 22 million metric tons were mainly used to produce fishmeal and fish oil. Between 1961 and 2018, food fish consumption grew from 9.0 kg to 20.5 kg, per capita, at an average rate of about 1.5% per year, with Aquaculture representing today 46% of the total fishery products consumption, or 52% if non-food uses are excluded. Capture fishery production has been rather static since the 1980s, putting aquaculture responsible for coping with the progress of fish consumption [16].

Food safety has become an important public-health issue, with authorities, such as the European Union, being proposing directives requiring the registration of the origin of certain products, improving product traceability and faster recalls when necessary. With regard to food safety, the European Commission established, about ten years ago, food legislation that introduced the concept of traceability, specifying that any food product must be able to be traced from its origin to the final consumer. This concept is already implemented in some sectors (e.g. meat), but in terms of fisheries and aquaculture there is still a way to go.

Fish products (capture or aquaculture) are subject to contamination and other problems such as poor conservation conditions or excessive storage time at the source or along the business value chain. The traceability of the business value chain of fishery and aquaculture allows to identify problematic product lots, and streamlines lots' recalls.

Food safety has also become an important marketing issue, as consumers want to be informed about what they are eating or buying. This growing consumer interest in the origin of the products they consume, and how products are treated (e.g. in respect to storage, transportation) along the value chain, has put some pressure on the need to trace the products in the fisheries and aquaculture value chain.

Growing demands for quality and awareness of customers implies knowing the geographical origin of the products and that the location of

* Corresponding author. Polytechnic Institute of Viana do Castelo, Viana do Castelo, Portugal.
E-mail address: miguel.cruz@estg.ipvc.pt (A.M. Rosado da Cruz).

https://doi.org/10.1016/j.jafr.2021.100169
Received 31 March 2021; Received in revised form 30 May 2021; Accepted 30 May 2021
Available online 3 June 2021

the products and their conditions of transport and storage are continuously measured and monitored. The study presented in Ref. [25], identifies and describes the quantitatively estimated benefits of traceability and concludes that traceability in fish supply chains allows to:

1. Reduce costs involved in product recalls when necessary and the recall may be performed in a faster way;
2. Reduce risks and costs associated with foodborne disease outbreaks and all the associated impact, like medical care costs, miss work, etc.;
3. Improve the company reputation and help maintaining market and consumers' confidence in the product.

This article reports the development of a fishery and aquaculture products' traceability platform, in the scope of ValorMar R&D project, from the identification of the business processes and stakeholders involved in the fish and fisheries value chain, to the specification and design of a proposed platform for supporting the traceability of fish and fishery products from the sea or aquaculture farms to the plate. The ValorMar Project involved thirteen R&D entities and eighteen companies, including one of the major retail distribution networks in Portugal and the state owned firm responsible for most of the first sales of fish at auction, especially wild fish, but also fish from aquaculture.

This proposed "from sea to fork" traceability solution includes a backend platform, or middleware, and a set of frontend applications targeted at different types of stakeholders along the value chain. The presented solution is an answer to the EU's General Food Law, which has made traceability compulsory for all food and feed businesses, and is the first Portuguese answer to provide fish traceability information to the final consumer. With this platform, final consumers become equipped with information that enables them to make informed choices when buying fish and fishery products.

This project's research process is structured according to the Design Science Research (DSR) methodology, which is aimed at research problems with unstable requirements and complex interactions, either between the diversity of stakeholders involved, from the entire fisheries value chain, and between the problem sub-components and their solution [18].

DSR requires the creation of innovative artifacts in order to solve a specific problem on a specific domain. It is based on an iterative and sequential approach involving six main activities: problem identification and motivation, objectives definition, design and development, demonstration, evaluation and communication [10,19]. Although, with these activities, the sequential order should be respected, the process is iterative, having some phases (steps) that can be repeated as many times as required, so that the best and accurate result possible is achieved. So, even when on the evaluation or communication phase, near the end of the iterative process, it can go back a few iterations, to the definition of objectives or to design and development, if required. But then, going forward, all the phases will have to be repeated sequentially.

The remainder of the article is organized as follows. Section 2 reviews the relevant literature on traceability in value chains giving special attention to fish value chains. Section 3 identifies and models the inter-value-chain participants' business process. Section 4 presents the platform architecture and the design of the traceability middleware platform. Section 5 explains how data is collected to feed traceability information into the platform. Section 6 addresses how the traceability information is made available to the final consumer and other intermediate value chain users. The results are analyzed and discussed in section 7. Finally, section 8 concludes the article and proposes possible avenues for future research.

2. Related work - food traceability platforms

Agricultural traceability systems have an important role to play in improving our food safety. Since the European Union published directives regarding the traceability in food sector, several platforms have been proposed to implement and support the traceability in feed and food value chains. There are platforms to track different kinds of agricultural food products, such as fruit [28], vegetables [4], cheese [32], wine [5], coffee [26], protected designation of origin products [1], etc.

Narrod et al. [28] created a "public–private partnership" to implement traceability of fruits and vegetables supply chain. The paper compares the ways in which small producers of grape, in India, and green beans, in Kenya, have coped with increased demands for food safety from their main export markets [28].

Bevilacqua et al. [4] reengineered the business processes of vegetables supply chain (fresh vegetables, canned vegetables, mixed vegetables, cooked and pre-cooked vegetables) and created a software system for managing the traceability of those products. The authors present a framework based on EPCs (event-driven process chains) and use ARIS (Architecture of Integrated Information Systems) to create a Web interface to provide information to the final consumer [4].

Regattieri et al. [32] propose a platform to support the traceability of the Parmigiano Reggiano cheese, from the bovine farm to the final consumer. The developed system collects data in all identified steps of the food chain and stores it in a central database assuring a origin of the milk and the cheese [32].

Biswas et al. [5] propose a blockchain-based platform to trace the wine value chain, from its production to the purchase history. The authors designed and implemented a blockchain with one entity for each wine chain operator. Three of the entities were designed as the miners and all entities in the chain generate their own blocks containing transactions. These transactions are verified by the miners before being added in the chain [5].

Alves et al. in Ref. [1] are proposing a blockchain-based platform to implement traceability in PDO (Protected Designation of Origin)/PGI (Protected Geographical Indication)/TSG (Traditional Specialty Guaranteed) products. The platform has two main goals: the first one, is to avoid forgeries; the second one, is to provide information to the consumer about when, by who and where the product (and raw materials) was produced or manufactured.

In [26] the authors implemented a blockchain-based Web application, more specifically Hyperledger Fabric, to track the coffee produced in Colombia. The aim of the platform is to improve the transparency and fairness in the value chain and thus improving the lives of workers involved in the chain.

Some platforms are more generic and are created with the main objective of tracking any agricultural product, as is the case of [7,15,24,36].

Folinas et al. [15] propose a web application for modeling agricultural processes and data. The application assures that all business partners have access and share all information. For that, they define standards and specify the exchanged information.

In Cheng et al. [7] the authors perform an analysis of existing agricultural traceability systems and propose a common agricultural traceability method, describing the traceability information of different products with different processes of circulation by quoting the concepts: traceable resource unit, information granule, and supply chain, which can be applied to all agricultural traceability.

Feng Tian proposes a decentralized system based on IoT and blockchain technology to support traceability in food supply chain [36].

In [24] the authors propose a food traceability system based on blockchain and Internet of Things (IoT) technologies. This system uses IoT devices to replace manual recording and verification as much as possible, reducing this way the human intervention in the system effectively.

Baralla et al. [3] propose a platform to guarantee the origin and provenance of food items in a Smart Tourism Region context. The created blockchain-based platform allows to manage and track food items in an agri-food supply chain, assuring transparency and trust to the final consumer.

Several authors have proposed platforms to trace fish and fishery products from capture (or production) to the final consumer as is the case of [11,17,27,29]. Nicolae et al. propose a structural design for a tool to support the traceability in Romanian fisheries supply chain with the aim of monitoring the safety and quality of fish and fishery products [29].

Gardner et al. [17] performed a study in Madagascar about the value chain of octopus and mangrove mud crab. The project focused on reducing post-harvest mortality in the mud crab fishery.

Liliana Moga designed a Cloud computing based system for monitoring the supply chain of fish and fishery products. The author identifying the factors that influence the acceptance of traceability by the fish and fishery products business sector taking in consideration the general principles of traceability, the European Union (EU) and the national legal framework, as well as the particularities of the fishery industry [27].

In [11], two of this article's authors propose a blockchain-based platform's prototype to trace fish lots, back and forth, throughout the entire fisheries value chain. The Blockchain technology allows registering all chain activities in a distributed, transparent, secure and trustful manner [11]. To implement the traceability platform, the authors defined a smart contract to be used on the Ethereum blockchain. This blockchain-based prototype is still very far from a production stage, and needs further developments, namely towards resolving a grave performance issue.

3. Modelling the fish and fisheries value chain

The traceability of fish or fishery products, like other food products, is very important, both to inform the consumer about the path of the fish through the value chain and to be able to recall fish lots in case of threats to the public health [8].

Fisheries supply chains start at the sea, lakes or rivers, where fish species are captured, or at aquaculture farms. Fresh fish is then typically sold at fishery auction traders, and then continues its way through the value chain (e.g. logistics, industry) until the final customer. Sometimes, while still at sea, the fish is transformed, in industrial fishing vessels. At the vessel, the fish may be frozen, salted or transformed in other fish products. These fish products also make their way through the value chain until the final customer. The ValorMar R&D project has been developing a traceability platform of fish products, originating both from fisheries and aquaculture [9,33]. The traceability information must be inserted, in the developed platform, by the involved stakeholders, that is by the supply chain operators.

In previous work [9], the authors analyzed several value chains in the fish and fisheries area, such as canned sardines, fresh wild fish, salted fish, frozen fish, and others, and represented each value chain as an inter-organizational business process model, by using the Business Process Model and Notation (BPMN) language. In the same paper, all different modeled value chains are abstracted, and similar steps in the different value chains are identified, leading to the proposal of a generic integrated business process model. This model is able to represent, at a high abstraction level, all participants and all activities involved in any fish or fishery products value chain. The proposed business process model, represented in Fig. 1, identifies the main types of companies, or value chain operators, involved, represented in the model as external participants, such as: ship and ship owner, aquaculture company, industry, retailer and logistic company.

During the value chain process life cycle, the fish or fish product lot can be sold, transported, stored and transformed multiple times. Fresh fish, in principle, does not undergo any transformation, but frozen, salted and canned fish can undergo various types of transformations. For example, cod fish can be salted and dried for storage. Then the fish can be soaked and frozen to be stored again, before being defrosted to be cooked. Once cooked it can be refrozen. Sometimes these transformations are made by different companies, so there may be activities of transportation, sale and storage between each transformation. After each of these activities, there is typically a product quality assessment and, often, there is a lot renumbering, as may be seen in Fig. 1 (activity *Register and quality assessment*).

The information on each activity must be properly stored. This information must be provided by the business partner responsible for carrying out the respective business activity. The business partner, represented in Fig. 1 as an external participant, must be properly identified and its information must be properly stored.

In [8], the domain model for the Fish and Fisheries traceability platform has been extracted, based on the information represented in the BPMN business process model represented in Fig. 1, following the rules presented in Ref. [12]. The resulting domain model, represented in Fig. 2, has served as the conceptual basis for the ValorMar Traceability platform development. Here, we can find the main conceptual entities for the fish and fisheries value chain traceability problem. The central *Event* represents the value chain activities. It has several specializations,

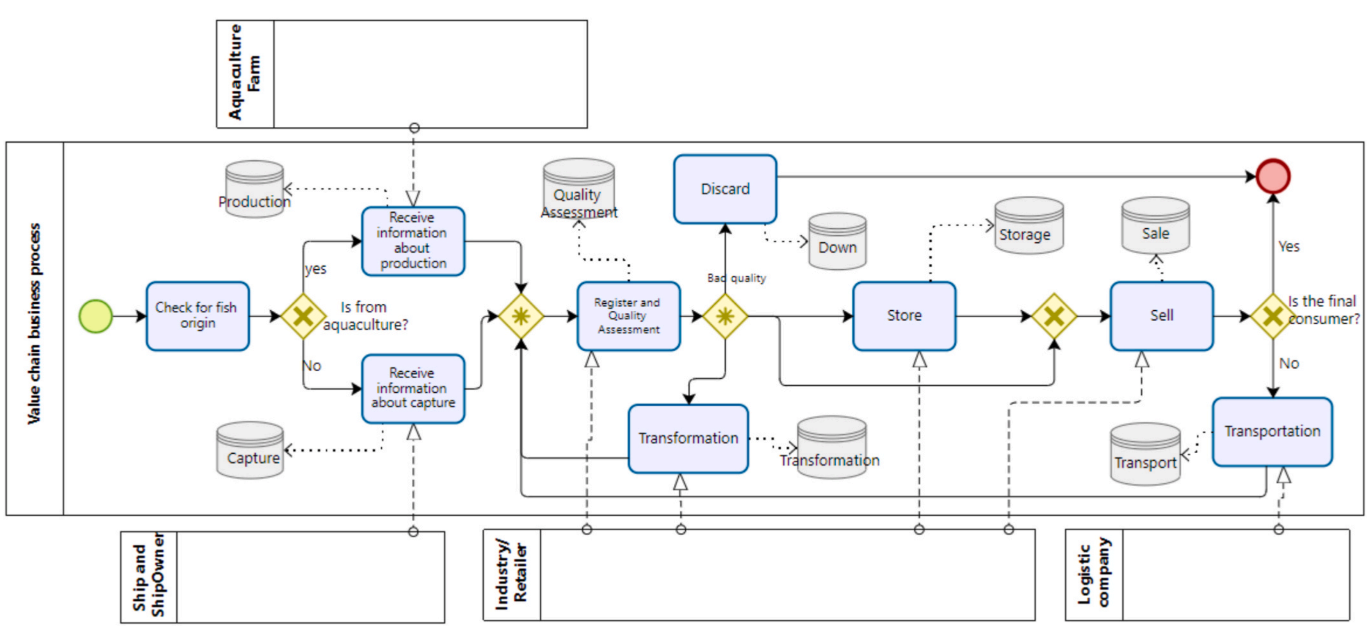

Fig. 1. Generic Integrated fishery value chain business Process Model (adapted from Ref. [9]).

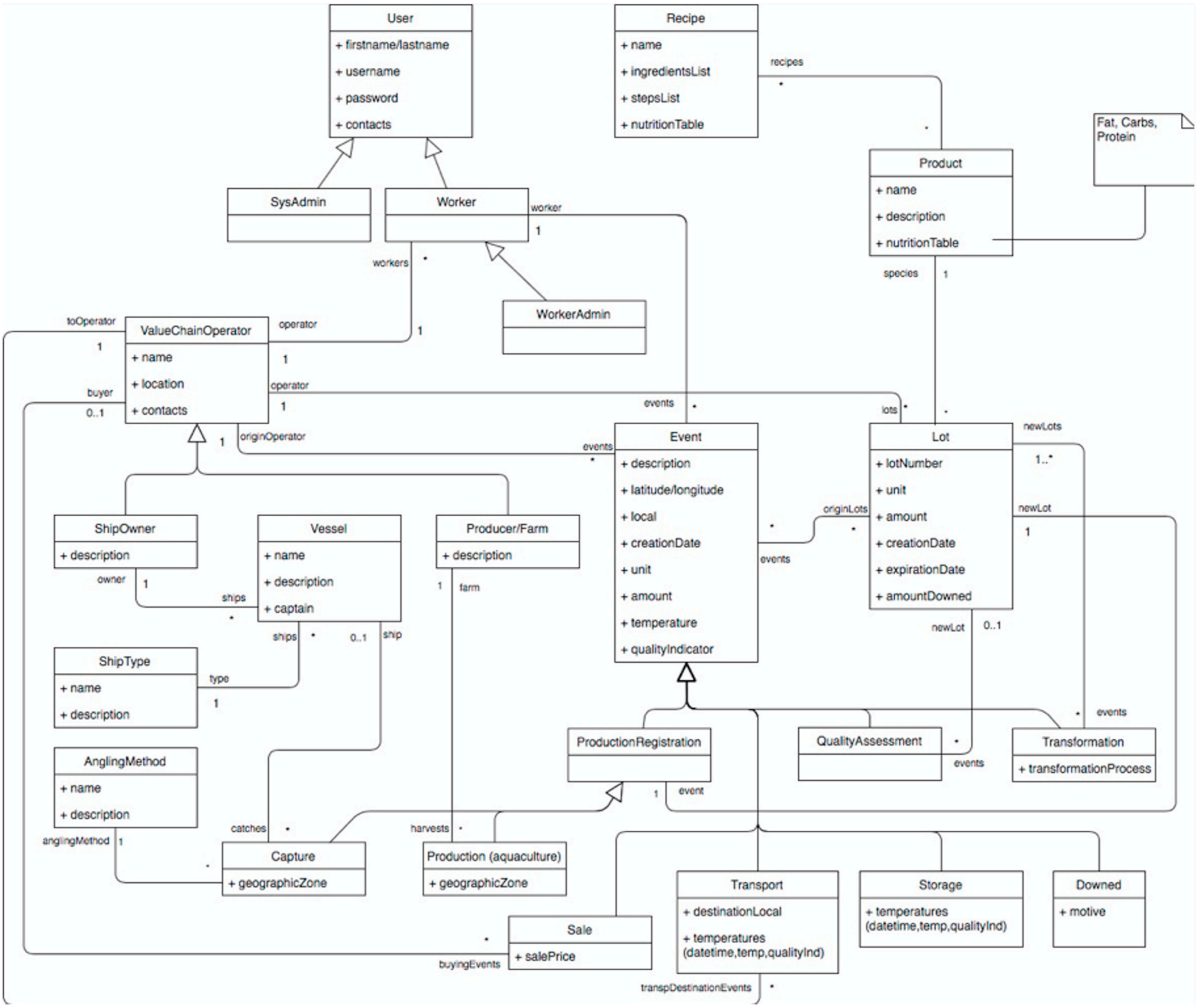

Fig. 2. Domain model (taken from Ref. [9]).

that are conceptually connected to the business process activities, such as *Capture*, *ProductRegistration*, *Transformation* or *Sale*, among others. Each *Event* (value chain activity) acts on one or more *Product Lots*, and may yield other *Lots* as output. *Events* are performed by *ValueChainOperators*, and these own, or are responsible by, *Lots*.

4. Traceability platform architecture and design

The overall architecture of the ValorMar project is depicted in Fig. 3. The proposed traceability platform is based on a backend middleware platform that processes and stores the traceability information in a Database (DB). This middleware offers a backend Application Programming Interface (API) for interacting with applications that feed the platform with information, and applications that consult the traceability information in the platform and present it to final users. The former are applications from the value chain operators, such as adapted Enterprise Resource Planning (ERP) or Manufacturing Resource Planning (MRP) applications. These applications send operation requests to the API through a Kafka Message Queue broker. This allows all requests to be executed at the pace that the platform is able to process them. The latter applications are kiosks, used in retail stores, and mobile apps for final consumers and authorities.

4.1. Middleware

The proposed middleware platform uses a Microservice Oriented Architecture [21], as depicted in Fig. 4. This choice of architecture enables the storage and retrieval of traceability information in a collaborative approach, with all the operators communicating with the middleware. The middleware platform required the adoption of widely used protocols, to ease the process of integration of the operators' applications, for feeding the platform, and to provide high flexibility for the development of applications, for different scenarios of consuming the information provided.

The information stored in the middleware is focused on two major sets of data [13]: traceability information and informative data. The informative data serves as a complement to the traceability information and is used to enrich the end stories about the products presented to the customer, such as taxonomy or nutritional information.

The middleware platform allows interactions from users belonging to three main entities: Operators, Middleware Admins and End-Product Consumers. Operators' users can define the structure of the

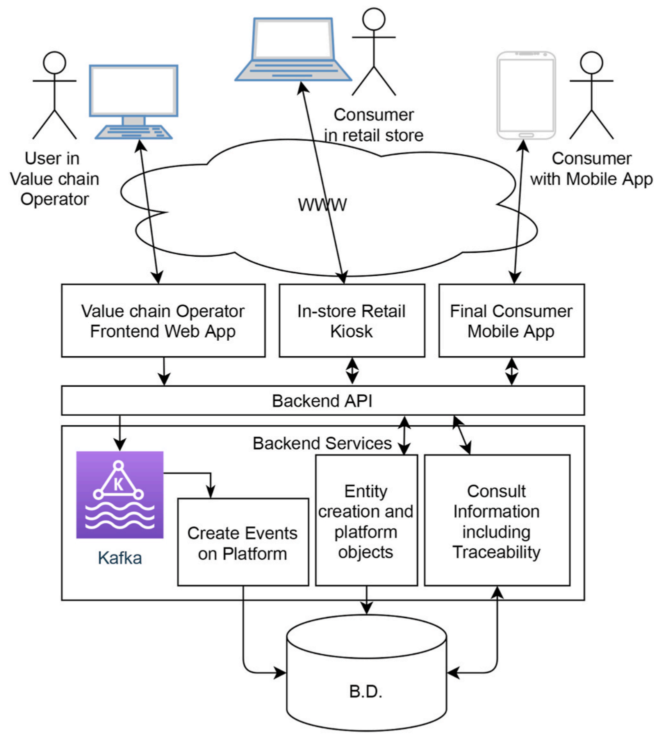

Fig. 3. ValorMar platform overall architecture.

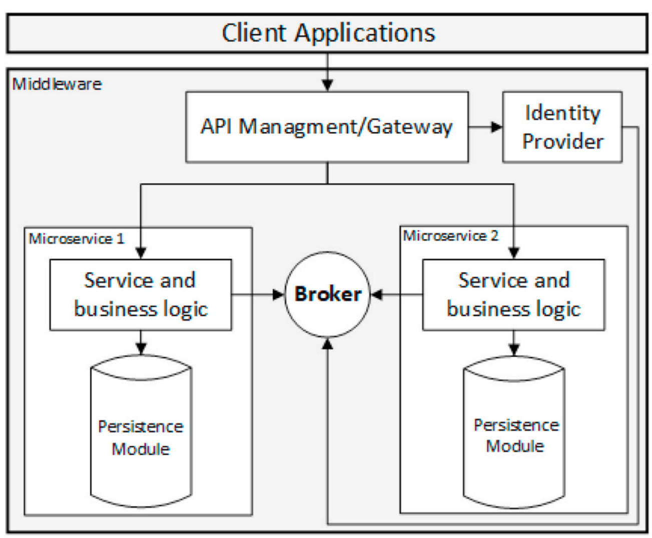

Fig. 4. Micro-services' structure of the Middleware Platform (Backend services) (adapted from Ref. [13]).

traceability information they want to register in the platform, and may register that information. Middleware Admins manage the users' and operators' permissions for the entire system, and the product consumers will be the ones consuming the collected data. In later development iterations new user profiles may be added for Data Science purposes.

The extensibility and interoperability requirements were the main reasons for the adoption of a microservices architecture for the middleware. These are advantages of this type of architecture over monolithic applications, along with scalability. As previously stated, the goal was to merge information with different structures, purposes and volume, and the use of microservices allows us to tailor each feature as a separate service that manages each data domain. The REST APIs, that are usually implemented in this type of architecture, also suit the need to allow interaction with the middleware from different kinds of client applications, which can range from responsive web applications to autonomous agents, which collect data from the value chain, using sensors, and communicate that data to the traceability platform.

4.2. Domain segmentation

The data model used in the current implementation of the middleware platform was based on the domain model depicted in Fig. 2, with some changes made during the final stage of design. The main change was the segmentation of the middleware's functionalities with Domain Driven Design [23]. Five domains have been identified, and each should be developed as a separate sub-system. This segmentation was done focusing primarily on the volume of data and the expected frequency of operations regarding that data, as well as the functionality attached to each entity and the feasibility to decouple them from each other. The current implementation is represented in Fig. 5, where each domain is identified, together with their composing entities.

Following is the list of identified domains:

1. **Information**: Mostly static information regarding the businesses, processes, products and participants related with the supply chain. This data can be of any type and unstructured, such as long text descriptions, videos or images. This domain was implemented using a Wiki engine and, from the perspective of the remaining components of the middleware, two types of entities are used to store the information: *WikiPages* and *WikiInfoboxes*. The pages allow us to store unstructured information, and they may be complemented with Infoboxes, for situations where the information stored needs to have a defined structure to enable an easier retrieval process.
2. **Tracking**: Data related to the events and product lots registered by the operators in the value chain. This is used to create the history of the final product lot and for logging and tracing purposes. In relation to their previous iteration's counterparts, the Event entities have dynamic structures instead of fixed types and fields. The fields related to traceability *Events* are stored in the *EventInfo* table which acts as a Key-Value store for any needed field.
3. **Governance**: Comprises the management of the runtime configurations for the traceability event data structures, related monitoring triggers and internal access control. This approach allows the operators to adapt the middleware to their particular needs while maintaining a fixed structure, providing advantages for data warehousing and data mining operations. Resulting from this feature expansion were the *EventType* and other auxiliary entities. In each event type, a list of mandatory fields can establish a fixed structure for that type of Event. All the entities regarding Operators and Users in the original model were simplified into a *User* and *Operator* entity in this domain which mostly serves to establish the relation of the information with the other domains.
4. **Users and Access control**: Contains information about users, roles and permissions, and the user data related to the security mechanisms of the API Gateway. An Identity Provider was used for the implementation of all of these features and the *UserIdentity* table represents the information which this system is responsible for, that is shared with other domains.
5. **Monitorization**: Consumes the collected events and produces notifications according to configurable rules defined for each type of event. The *Trigger* entity class is used to describe the monitoring triggers. These triggers allow operators to monitor the occurrence of value chain events of a specified type. The goal is to streamline the management and quality control of fish and fishery products' lots. A trigger can have one or several criteria. A criterion allows to define a condition on a field, for firing the trigger. An *Action*, associated to a trigger, defines the recipients that will receive the messages whenever a trigger is fired. One trigger message can go to multiple

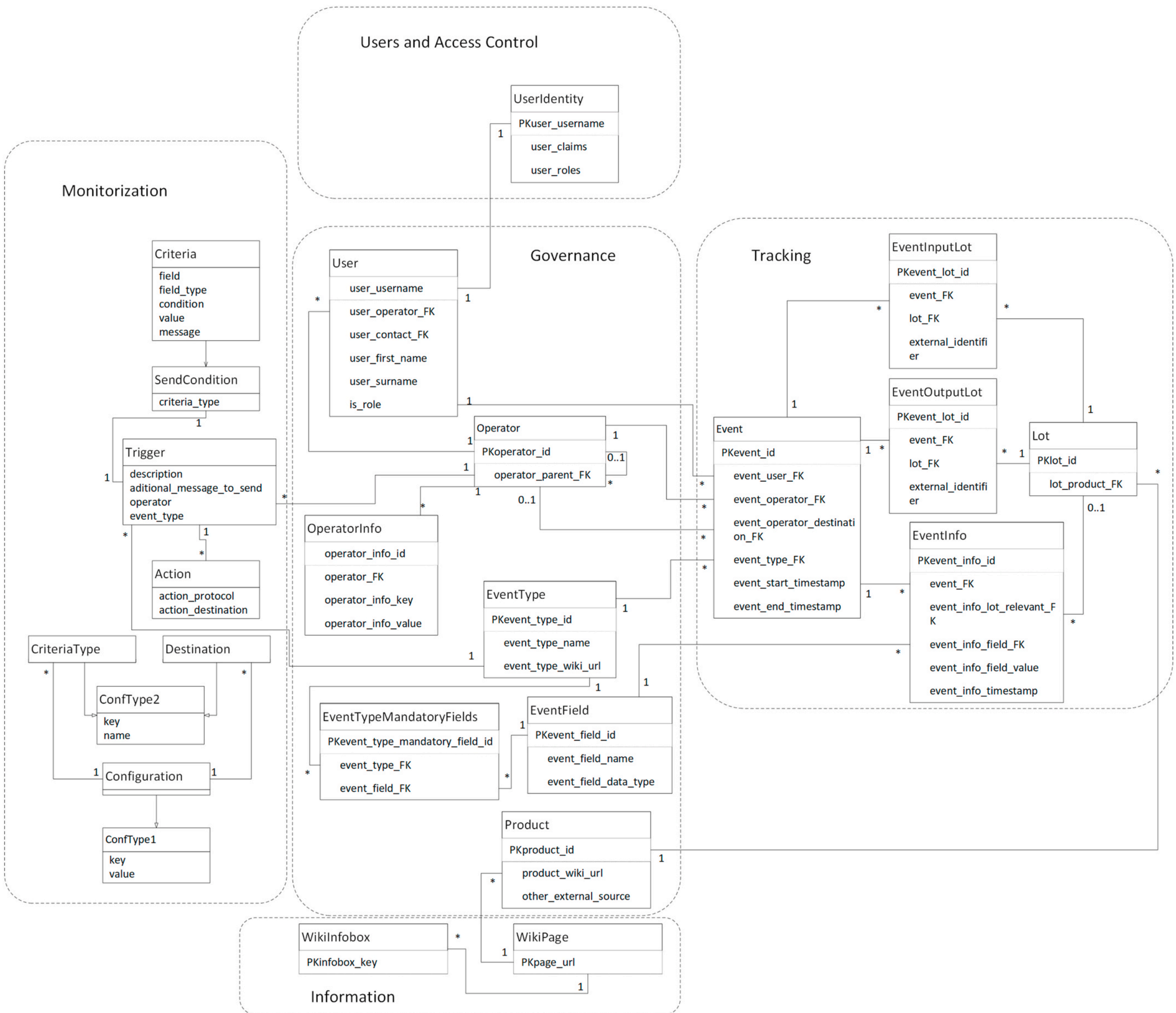

Fig. 5. Middleware database model, with entities distributed by the five identified domains.

destinations, using different actions. The *Configuration* class allows to configure the criteria and the actions.

Of these five domains, resulting from the refinement of the originally planned architecture, the nuclear resources to satisfy the requirements and objectives of this project are those contained in the Tracking domain. The diagram in Fig. 6 shows the current implementation of all the components that operate in this domain and their communication.

The focus during the implementation of the Tracking domain's microservice was to enable asynchronous insertion of traceability Events through a REST API. This requests, to feed the platform with traceability information, from the operators' integration applications, are routed through the API Gateway to a WSO2 Micro-Integrator instance that registers the requests in Kafka, to be later processed and persisted in a database. This process is done by a.Net Core application that consumes requests from Kafka and inserts them into the Microsoft SQL Server engine used with the Entity Framework Object Relational Mapping (ORM) tool. In later instances of development, other interfaces were created to push traceability information to the middleware using table files. These files can be sent attached to emails or through a REST API, and are then processed into the same format as the requests from the original API to maintain the remaining event insertion pipeline. During the insertion of the events to the database, the rules configured for that event type are enforced. These configurations are present in this database by using Kafka Connect, to replicate the information managed by the Governance domain. The reply to the HTTP requests for event insertion only acknowledges that the middleware has queued the request for processing with the proper identification. The acknowledgement for the correct persistence of the information, or the error report in the non-complying cases, is sent via HTTP to a REST API. This requires the integration applications to implement this functionality with a REST API to enable this feature.

The event retrieval and traceability querying operations are exposed through the API Gateway, which routes to a REST API, implemented with.Net Core and Entity Framework to query the database and return the result. These acknowledgement messages are produced after the processing of the request by the internal applications and pushed onto the Kafka broker. In the current implementation, these messages are only consumed by the.Net Core application that notifies the operators from a REST API but, in the future, more applications can be added to

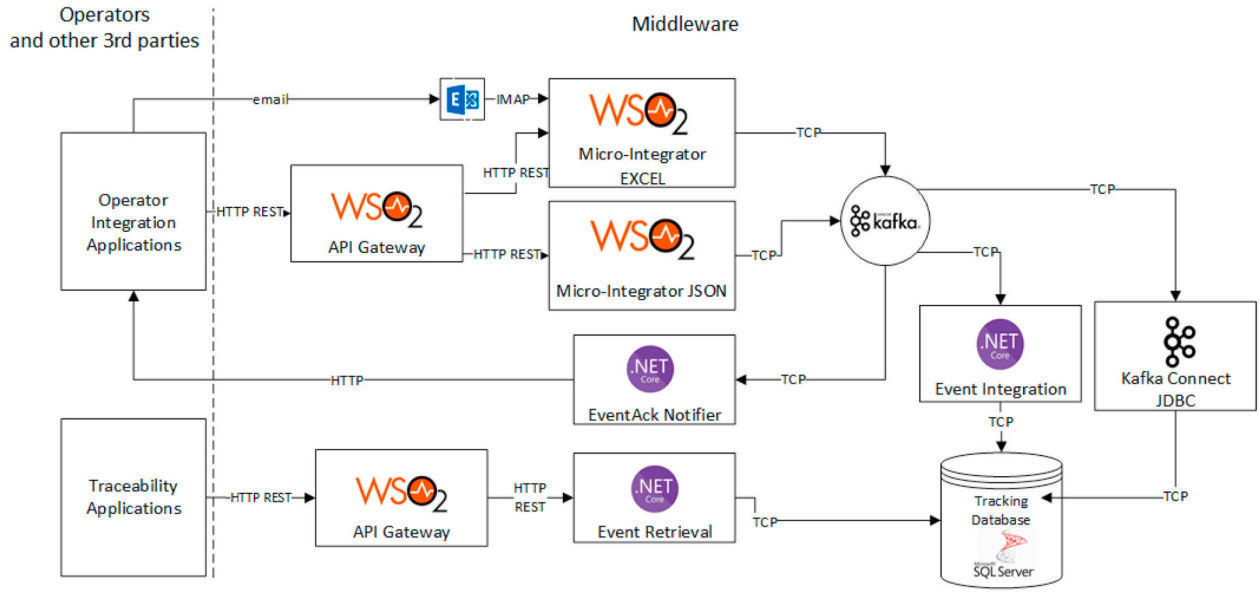

Fig. 6. Tracking domain implementation diagram.

implement more types of communication and improve the interoperability of the middleware.

4.3. Microservice management and cross cutting features

In a microservice-based architecture we can still find layers that are typically found in more Monolithic Architectures, such as the persistence layer and the business logic layer, but here they exist in the scope of each microservice. Despite this, there are always common features across all of the microservices such as the ones implemented by the API Management Layer [20] (refer to Fig. 4). This API Management layer acts as a logically centralized governance system to manage the microservices that are added as the project lifecycles progress, as well as version control of these services. It also allows the developers to expose the services in a managed way to the consumers, building an ecosystem of APIs and API consumers. This layer can also handle cross-cutting features such as API security by communicating with the User and Access Control microservice and API logging and auditing.

Internal communication between microservices is a feature commonly present in a microservices-based architecture. As seen in Fig. 4, one way to achieve this is through the use of a message broker, as suggested in Ref. [22]. The microservices communicate with each other with messages that are created by the microservice that owns the domain of the information being manipulated and received by any microservices that must be aware of the change that was done and, if needed, such data is then replicated into the listening microservice's persistence layer.

5. Feeding the platform with traceability information

Applications to communicate information onto the traceability platform are used by the value chain operators. These, use their ERP, MRP or other management applications, which have been adapted to integrate with the traceability platform. Some operators don't have an adapted management application, and, for these, a Web application has been designed in order to allow feeding the platform with traceability information. This application is the one described here, although in what respects to the interaction with the middleware, there are not many differences from the others.

As depicted in Fig. 3, the value chain activities (Events) submitted to the platform are queued through a Kafka message broker, before being fetched and processed by the platform middleware. Such activities are important from a traceability perspective, since they also contain product lots' information, and quality attributes, dates, locations and responsible operators, for each product lot. The Front-End applications' interaction process with the middleware, presented in Fig. 7, aims to illustrate the insertion and visualization of information related to the middleware API. It is a Single Page Application (SPA) composed of several components grouped by the domains that structure the API (refer to Fig. 2).

The application in (1) (refer to Fig. 7) presents the recent activities related to the value chain operator (e.g. Activities, Product Lots). In (2), the user can submit a new value chain activity operation on an identified product lot, such as the ones identified in the business process model in Fig. 1 (e.g. Fish Capture, Transformation, Transport, Storage, Sale). As a result of this activity a payload is generated (3), that aggregates all the information that will be sent to the API. Every activity on the application will generate a payload that is sent (3.1) to the middleware API (4) for validation and insertion. The requests are received, in the middleware, by its Middleware gateway (4), which is the API gateway layer, which forwards the request to the correct service. For services that require authentication the gateway uses an Identity Provider (5) to validate the users' identities. Successfully authenticated and authorized requests are then routed to a pre-processor (7). This instance takes the activity data and puts it in topics on the Kafka message queue.

The Processing component (8) consumes the requests stored in those Kafka topics and stores the data in a database after verifying that the information is correctly structured, and produces a notification and queues it in another Kafka topic.

After sending a request, the client receives the reply asynchronously when the Notifier component (9) consumes the reply from Kafka and sends it to the client's reply endpoint (10). If there is success, in (11), the app shows the details of the data inputted on the API, in (12).

Fig. 8 presents the detailed functionality diagram of the Front-End application for feeding traceability information onto the middleware platform. In (1), the application starts by presenting to the user relevant information related to the Business Operation (e.g. Activities, Product Lots).

For security purposes, OpenID Connect [34] is used to verify the user's identity, using ValorMar's Identity Provider. In (2), two security related processes occur, authentication and authorization. Authentication refers to the process of verifying the user's identity, and authorization is the process of verifying the permissions the user has for performing operations on the application. After the authentication

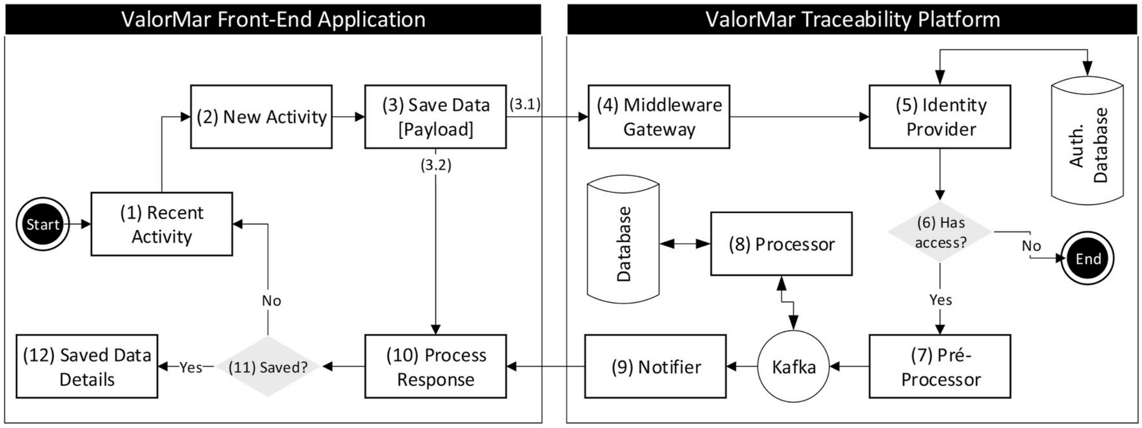

Fig. 7. Interaction Process between front-end applications for feeding traceability data onto the platform, and the Middleware Platform.

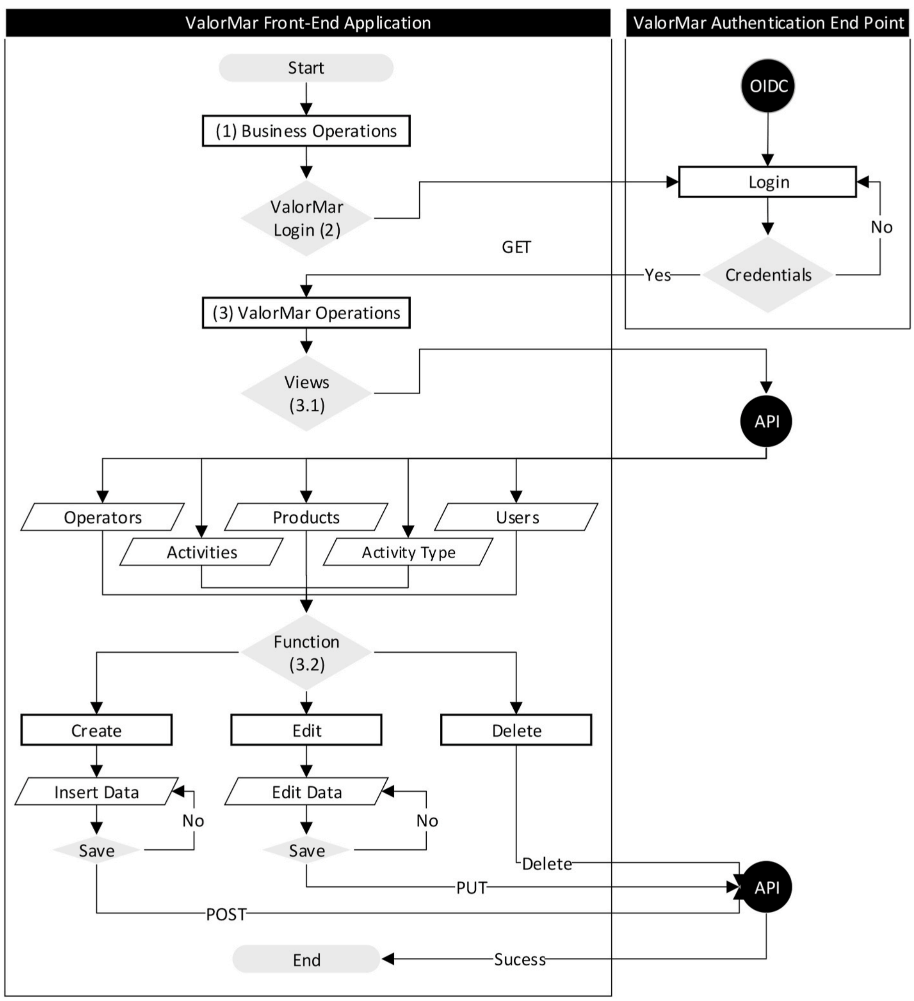

Fig. 8. Flow diagram of front-end applications for feeding traceability information onto the middleware platform.

process is done, a bearer token is returned that must be included in the authorization header of all requests made to the middleware API.

Once authenticated (3), the user has access to the main page of the application and therefore to the categories previously defined. Keeping the categorization in the API data model, the application is grouped into Users, Operators, Products, Activity Type (Event Type) and Activities (Events).

These categories can be selected using a sidebar menu. By selecting a category, a table containing all the information about it (e.g. Activities/ Events, Product Lots) from the current operator is generated (3.2). Note that, for each category, the HTTP verbs (GET, POST, PUT, DELETE) indicate the action that the user is requesting (e.g. Create, Edit, Delete).

If authenticated, in (3.2) the user is able to add, edit or remove entities (Events, Products, Users, etc.) from the previously selected category. Adding or editing an existing entity will generate a payload with information required by the API for the action.

When creating information, i.e. events or operators, some of these categories refer to other categories on the list. The interface will automatically generate lists with the matching operators for these new or updated activities.

6. Retrieving traceability information using ValorMar platform

Traceability information can be consulted through the use of front-end applications. These include an in-store Kiosk, which is presented in section 6.1, and a mobile application for the final consumer, that can be used to consult information about specific product lots, by identifying the lot's reference and the value chain operator. The mobile application is addressed in section 6.2. Both these applications allow the user to consult information about products or specific lots, namely traceability events, nutritional tables, recipes and more.

6.1. In-store traceability kiosk

The kiosk application has been conceived to be used by an in-store administrator, as well as the final consumers at the point of sale. The Kiosk application can then be used with administrative privileges, allowing a administrator to edit the in-store product catalog. Fig. 9 presents the Kiosk functionality flow, divided in 3 different boundaries, and its interaction with the ValorMar middleware platform:

- Non-authenticated users - In order to update current product catalog on the kiosk network, the administrator must first authenticate into the system. This procedure requires login credentials, that will be accepted or denied, from the platform.
- Authenticated users - After authenticated, the platform returns an access token that can be used on specific platform functionalities. An administrator can add, edit or remove products from the in-store kiosks' internal DB.
- Users - The users do not require authentication an can only consult information, either retrieved automatically from the platform or inserted in the Kiosk DB by an administrator.

This kiosk application will search and connect with other kiosk applications in order to synchronize the internal DBs. The in-store kiosks work under a peer to peer *zeroconf* network where all the integrated databases are updated when the administrator provides an update to the catalog. Each kiosk operates as a peer among the same store kiosks, which allows it to synchronize its database to its neighbors', everytime a product or lot is modified, and keep connecting to new neighbors as they are added. This avoids the need to update each kiosk manually and separately from each other. The *Zeroconf* network implemented for the in-stores kiosks is described in [31].

A product's page, in the Kiosk application, presents the final consumer with information from the centralized ValorMar middleware platform. The product information includes its nutritional composition, related recipes and the references of the product lots existing in-store for sale. By selecting one of these lots, the user is presented with lot specific information, from which the traceability information is the most important one. The traceability information is presented through a value chain events graph on a geographical map, as can be seen in Fig. 10. By selecting an event node on the graph, more information is given about that event. The specific information presented depends on the information reported by the value chain operator (only the geographical coordinates are mandatory for each event).

Fig. 11 depicts a diagram representing the functionality implemented for the in-store kiosk application, in a regular usage basis (final consumer perspective).

When the kiosk starts in (1), it synchronizes with the kiosk neighbors,

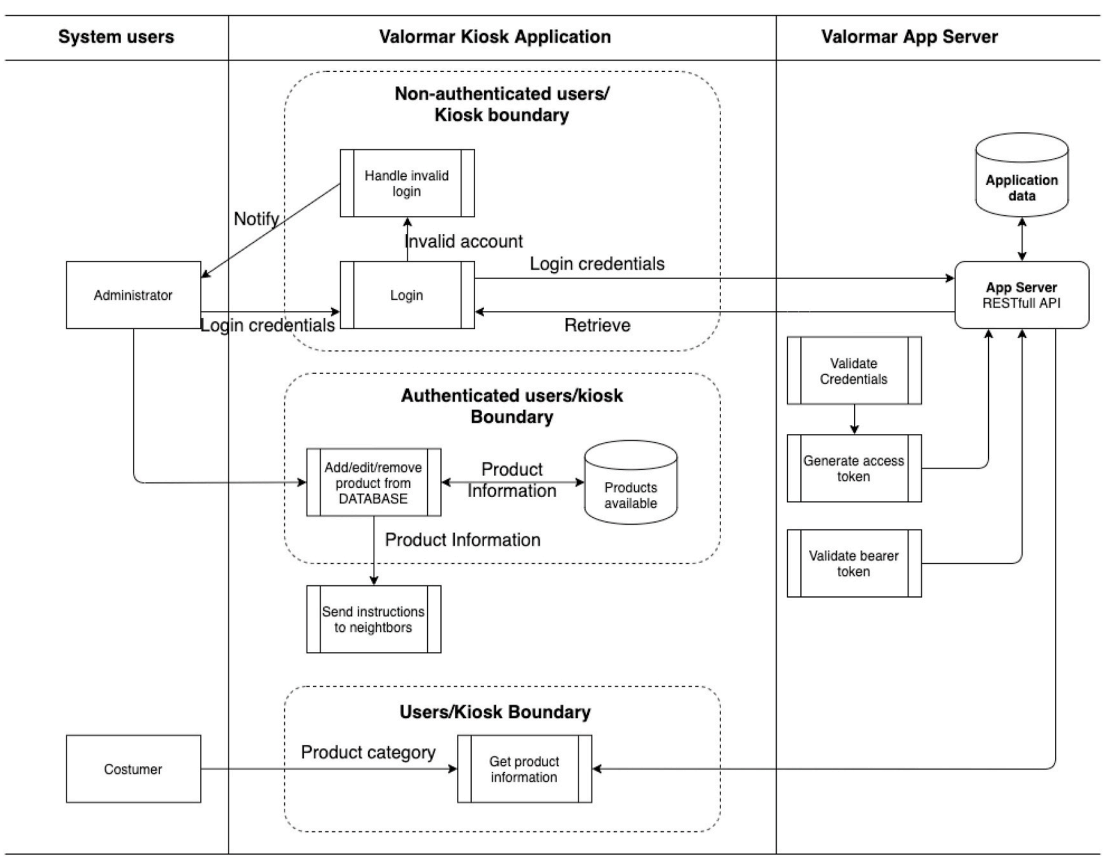

Fig. 9. Data flow of kiosk application illustrating administrative and final customer uses.

Fig. 10. Traceability map, in the Kiosk's product page.

in order to update the built-in DB on every kiosk. As described above, multiple in-store kiosks running the application, connect through a peer to peer network for synchronization.

The products page in (2) shows all the products available on the Kiosk's DB. From the page in (2) the customer selects one product and/or product lot to consult. The kiosk application, then, proceeds to (3), creating payloads with information of the product and/or lot, and makes asynchronous requests to the middleware platform (3.1), while awaiting for a server response, (3.2). A successful response will yield product related information, such as nutritional table and recipes, and product lot value chain events. These events' information will be used to render the traceability map in the kiosk application, providing the traceability information to the final consumer. Similar to the events' insertion, the requests for querying the platform are received and routed by the Middleware Gateway (4), with authentication and authorization of the users, ensured by the Identity Provider (5). Lot tracing requests are routed to the tracking service API, where the Tracking Retrieval component (7) queries the DB, starting from the lot with the reference supplied in the request (the final product lot) and aggregating all the events, regarding any lots that were used or related to the final product lot. This procedure documents the entire history and chain of events, of the given and derived lots, along the value chain. Other types of information are also available in the middleware, as seen in section 4. Other information related with requests are routed by the Middleware Gateway (4) to a Micro Integrator (8), which creates a proxy to a Wiki (9) that transforms and simplifies the native API, provided to extract information from the Wiki pages that describe the products traced by the platform.

The Kiosk's User Interface (UI) was conceptualized and developed with the objective to maximize the User Experience (UX) of any user, when using the kiosk to consult information without assistance. The kiosk application's usability has been assessed through a System Usability Scale (SUS) questionnaire [6]. This usability study can be found in [30] and is further analyzed in section 7.

6.2. *Final consumer mobile app*

In the ValorMar R&D project, a mobile application for the final consumers has also been developed. This application allows the users to view product lots' information, like with the in-store kiosk, but filtered to any individual lot. Fig. 12 shows the mobile application main screen. Given the reference of a product Lot and the product brand or value chain operator, the application queries the platform about that product lot and shows the traceability events as a graph, whose nodes are geographically pointed in a map.

This application was conceived with the objective of be clean and minimal, as the kiosk application, in order that any user easily interact with it. It supports multiple languages such as English, Portuguese and French, and is prepared to show several information, such as nutritional contents and recipes, besides the traceability information.

Fig. 13 shows the traceability map page, similar to the one existing in the kiosk application. As both applications have been developed using the *Cordova*, *VueJS* and *Vuetify* frameworks, the code may easily be ported from one to another.

7. Discussion and analysis of results

In order to validate the software artifacts developed in the ValorMar R&D project, validation tests have been performed and are described in the following subsections. Load tests were performed on the traceability platform and, since the in-store kiosk and the mobile app are meant to be used by the final consumers, it was decided to assess their usability and user experience using a SUS questionnaire, for being answered by the

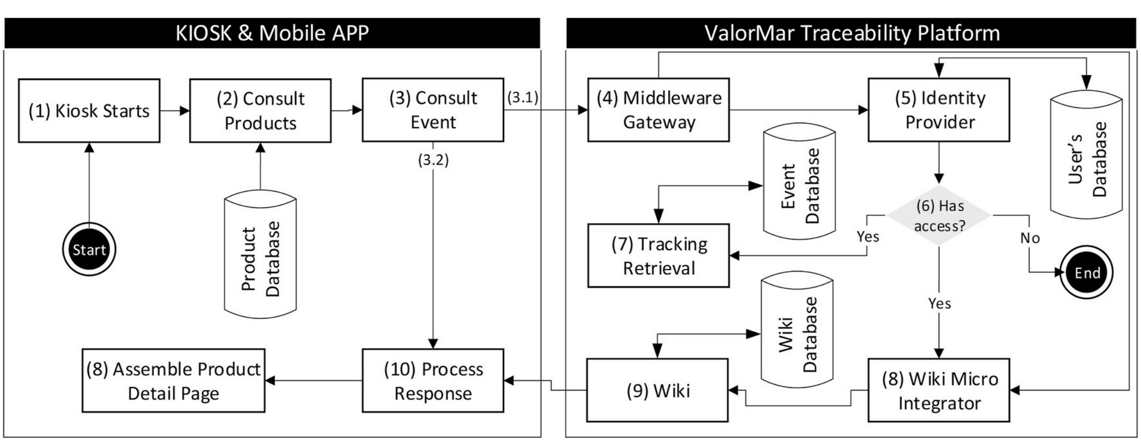

Fig. 11. Interaction Process between the Kiosk application and the Middleware Platform.

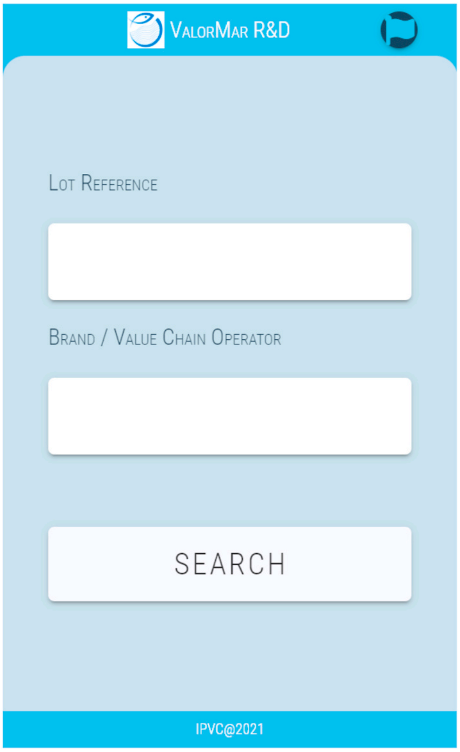

Fig. 12. Mobile application main screen.

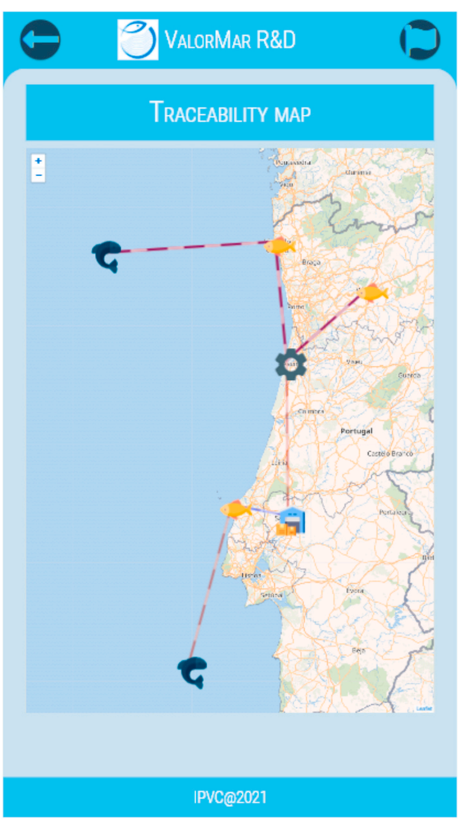

Fig. 13. Traceability map on the mobile application.

users.

7.1. Traceability platform deployment and validation

The deployment of the middleware platform was firstly done using docker containers on a set of Virtual Machines (VMs). The setup acts as a staging environment for developers, who may integrate and test their client applications. For the final production deployment, *Kubernetes* was used to orchestrate all the containers, in a scalable and highly available environment. To validate the deployed middleware platform, a set of load tests has been run. One of the goals was to simulate high usage scenarios and assess the system response. The middleware platform was deployed in a five node Highly Availiable (HA) *Kubernetes* cluster, installed on a private cloud managed by *OpenNebula*. Thus, 16 physical cores divided into 32 virtual units and 32 GB of RAM, were allocated for each node.

Each component of the middleware platform has been deployed with at least two replicas each, except for the components where the performance increases and thus the inclusion of more nodes was tested. Another component was kept separate from the *Kubernetes* environment due to compatibility constraints at the writing process. This was the SQL Server DB cluster. This cluster was deployed on 4 nodes, with 12 cores and 12 GB of RAM each, with three replicas available for the routing of read-only queries. Each of the application nodes was deployed with 0.5 cores and 2 GB of RAM, both dedicated, but the system allowed to further allocate up to 8 cores and 8 GB of RAM with *Kubernetes*' resource allocation configurations. The remaining software, such as WSO2 API Manager for the API management layer, load balancers and other components, has been deployed following vendor recommended settings. The workload has been applied to the middleware, which consists of four JMeter worker nodes with 8 cores and 8 GB of RAM, in order to complete the setup test.

The tests were applied to the Tracking API, which allows for the retrieval of traceability information about a product lot. The DB was loaded with sample data with short event chains, and a JMeter test collection was created, with a request, to retrieve the history of a product lot which included a valid authorization token in the header. Multiple runs of the tests were executed with different amounts of concurrent users and tracking application containers with the goal of evaluating average response times and throughput under load. These tests validate the integration and performance of the API management layer, application and DB.

Each configuration was stress tested for at least 5 min and the results are presented in Table 1. The *Test Configuration* columns' group represents the settings used for that test batch. In this case, the number of concurrent users, represented by the number of threads of JMeter instantiated and the number of nodes, represents the number of instances of the application evaluated during the test. The *Response Times* columns' group contains all the information collected during the test, regarding response times. Here, lower values indicate less delay in fulfilling the requests and a better system performance. The lower the delay is, the lower the perception of the user of that delay is, leading to a better user experience. The *Throughput* of the API is measured in transactions per second, and a higher value indicates that the system can successfully fulfill more requests, leading to more information being inserted or distributed to as many users as possible. The *Network* columns' group indicates the network bandwidth usage for inbound and outbound traffic.

The performance of the events insertion API has not been evaluated due to its asynchronous nature, which prevents us from using the same JMeter setup and will require the development of custom testing suites in the future. Another caveat is that Kafka acts as a buffer for incoming requests, so it should be taken into account, as the size of the buffer and the current occupation will drastically change the performance of the system, and so several scenarios must be tested.

The results (Fig. 14) were slightly lower than expected with the

Table 1
Load test results - Average Response Time.

Test Configuration		Response Times (ms)							Throughput	Network (KB/sec)	
Threads	Nodes	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
2000	1	5927	29	17068	5784	7092	7245	8876.97	329	478.87	72.46
2000	3	2183	26	11139	1225	1803	1922	2501	909	1321.16	199.91
2000	5	1712	26	12423	594	1565.8	2024	3149.99	1156	1679.09	254.07
2000	8	1469	26	7019	1026	2114.9	2824.95	3928.99	1308	1899.82	287.46
400	1	568	26	3006	552	716	799	1102	342	497.84	75.33
400	3	425	27	2910	399	565	614	761	921	1337.39	202.37
400	5	344	26	2454	321	526	585.95	794.98	1130	1641.05	248.32
400	8	302	26	3456	285	445	500	611	1248	1813.03	274.34
200	1	568	26	3006	552	716	799	1102	342	497.84	75.33
200	3	371	26	1860	492	640	693	876	517	751.76	113.75
200	5	363	25	2130	466	667	721	838.99	533	775.16	117.29
200	8	335	25	2366	420	582	642	762.99	574	833.77	126.16

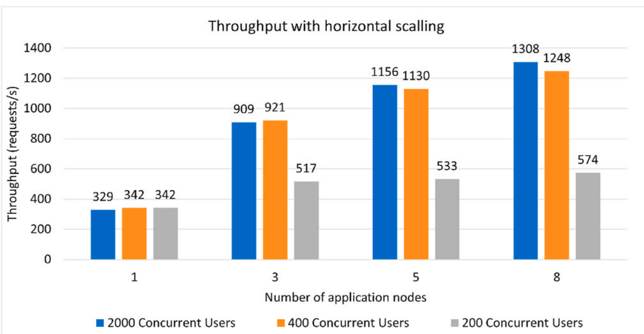

Fig. 14. Load test results - throughput.

resources allocated. Some previous tests executed without data and a draft of the application yielded a much higher transaction throughput. This indicates that, even with multiple DB nodes, the information retrieval portion of the request processing will most likely be the bottleneck in this scenario. During the tests that yielded 1300 requests/sec of throughput, the average throughput of transactions/sec in the DB worker nodes was over 6000 per sec, so in total 18000, which is an average of over 13 DB transactions per request. The differences observed with different quantities of application nodes were much closer to what was expected, with a nearly linear increase with the increase of resources. This increase is not perfect and this might be justified due to the other components being the limiting factor or due to the hardware configuration used, as this private cloud is a shared environment with other projects, and contains physical machines with different specifications.

Fig. 15 reports the results regarding response times, which were

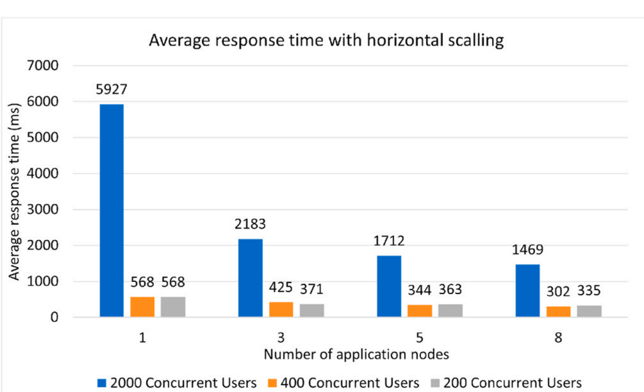

Fig. 15. Load test results - average response time.

similar for both 200 and 400 concurrent users. With a higher load of 2000 concurrent users, the increase in response times is higher, which could be noticeable to users. To solve this, the system would require more resources to handle the work load.

7.2. *In-store kiosk and mobile application deployment and validation*

The kiosk and final consumer mobile applications have been developed using the *Cordova* framework and also *HTML*, *CSS* and *JS*. The *VueJS* JavaScript framework and *Vuetify* Design framework were also added to the project to facilitate the dynamic creation of components.

The in-store kiosk application, previously presented, can run on multiple *Android* devices, but it was conceived to run on larger displays than those of mobile phones, such as tablets. The kiosk application connects to the neighbouring kiosks and, updates the application's built-in DB contents. This results on a broadcast of packages over the network, where the destination are the neighbouring kiosks and the package contents are instructions such as add, edit and remove product information. Delay and loss of network packages should always be considered. The test scenario comprised a custom network with a tablet and a mobile phone. Current results show that the network is functional, as expected typical in-store scenarios having one to three tablet computers. However, load tests have not been made yet for the kiosk or kiosks network.

The kiosk application is designed to be used by any type of user and thus, it was decided to use the SUS method, in order to test its usability. The SUS method consists in a ten question questionnaire. The questions are evaluated from strongly disagree to totally agree, using a *Likert* scale with five points, according to Refs. [2,14].

The general measure of the perceived usability is obtained by calculating the sum of the rating for the ten questions. The obtained result is used to verify the usability of the application, indicated by participants. The range of a SUS questionnaire goes from 0 to 100 and the product is well validated if the final score is above 68 [35]. Additional research has been made to map adjectives such as "worst imaginable", "poor" and "excellent" to range the scores for additional insight [6]. The SUS score can be measured in a few different grades.[1] Below 51, it is considered to be awful, from 51 to 67 it is poor, 68 means the application is decent, between 68 and 80.3 is good and above this last value can be considered excellent. The developed kiosk application here presented had a score of 81 [30].

The questions used on the questionnaire, were:

1. I think that I would like to use this system frequently
2. I found the system unnecessarily complex
3. I think the system was easy to use

[1] https://www.usabilitest.com/system-usability-scale.

4. I think that I would need the support of a technical person to be able to use this system
5. I found the various functions in this system were well integrated
6. I think that there was too much inconsistency in this system
7. I imagine that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I would need to learn a lot of things before I could get going with this system

The SUS questionnaire for the kiosk application has been available online for one week, during which 35 responses were obtained from participants of both genders, between the ages of 18 and 54. The results can be seen in Fig. 16. It is possible to verify, from the obtained results, that most participants indicated that the application is easy to use. Some participants agreed that the application should be less complex, however they considered that it is very easy to learn how to use.

Several suggestions indicated by the participants in the questionnaire were considered and some have already been added to the application, namely the following:

- The navigation bar with the back button should follow the user, while scrolling the product page. This prevents scrolling the page up again to use the back button.
- An interviewee asked for sections on the product page, instead of only having one single page with all the content. The idea was to add several sections, and thus the user can select the one of interest, such as traceability or recipes for that product.
- Some other users prefer the traceability information represented on a map, instead on a graph. Initially, it was decided to represent it on a graph, and the map was only used to represent the relevant geographical points. However, in the latest version of the application, only the traceability map with the graph nodes, geographically positioned, is presented.

The kiosk application was well accepted by the participants, with minor revisions, that were already included in the final version. Another question addressed to the questionnaire interviewees, but not considered for the SUS study, was if they find the application features enough and adequate. 91% of the users agreed with that.

In what respects the mobile application for traceability, its usability has also been assessed through a SUS questionnaire, with the same questions used for the kiosk application. Here, during the five days the questionnaire has been available, 12 responses have been obtained both from male and female participants between 25 and more than 65 years of age. The average results by gender are depicted in Fig. 17. The mobile application for fish lots traceability had a SUS score of 73.3. The SUS

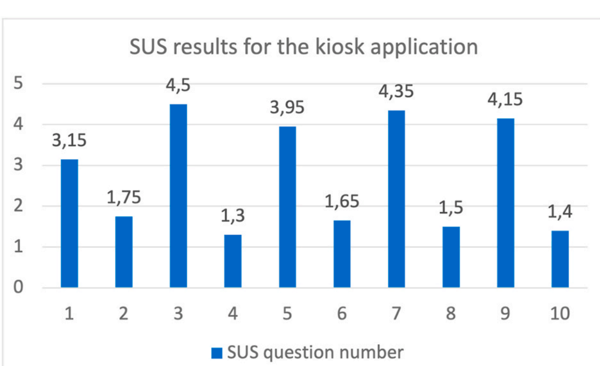

Fig. 16. Average results for the Kiosk application SUS, by gender of participants.

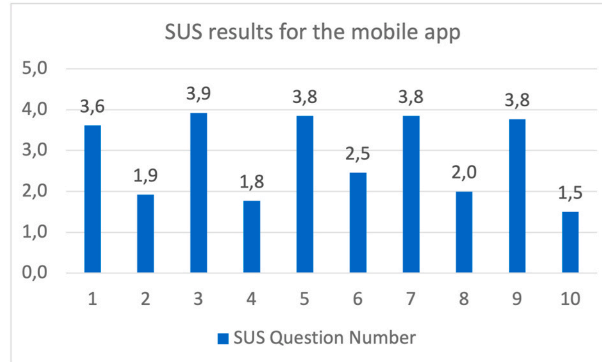

Fig. 17. Average results for the mobile app SUS, by gender of participants.

scores obtained, allow the conclusion that the kiosk application provides an excellent usability for the users, and the mobile app provides good usability.

In a set of questions, aside the SUS questionnaire, the participants were asked their opinion about the importance of knowing the origin of products and their traceability information. With a range from 1 (not important) to 5 (very important), 86.6% considered important/very important (answered 4 or 5) the knowledge about the origin of the food they buy (with an average answer of 4.3); 86.7% considered important/ very important the knowledge about the date of capture of the fresh fish they buy (with an average answer of 4.7); and, when asked about the importance of knowing the traceability information (places and temperature of origin, transport and storage) of the fresh fish and of other manufactured fish products, the percentage of 4 and 5 answers were, respectively, 66,6% and 73,3% (with average answers of 4.2 and 4.1, respectively).

About the questions on the usefulness of the kiosk and mobile applications, 73.3% answered 4 or 5 when asked if the information about the origin of fish given by these applications is important for taking a decision about buying or not buying a given lot of fresh fish (with an average answer of 4.25); 86.6% considers important/very important the information given by the applications about traceability (places and temperature) for taking a decision about buying or not buying a given lot of fresh fish (with an average answer of 4.33); 60% considers important/ very important the information given by the applications about the origin of fish products given by these applications for taking a decision about buying or not buying a given lot of manufactured fish-based product (frozen fish, canned fish, cooked fish-based meals), with an average answer of 3.8; and, 80% considers important/very important the information given by the applications about traceability (places and temperature) for taking a decision about buying or not buying a given lot of manufactured fish-based product (with an average answer of 4.25).

These answers confirm the importance of traceability of both fresh fish and fish-based products to the final consumer, and the usefullness of the proposed platform and applications to that end.

8. Conclusions and future work

With the globalization of markets and value chains, preserving the quality of products throughout the supply chain has become a challenge especially when it comes to food products. The value chains became global, bigger and involving more participants (business partners). Items from the same product lot may be transformed, stored and transported several times between their production (or capture) site until they reach the final consumer. Thus, storing information about what happens in each value chain stage becomes mandatory not only for providing information to authorities, in case of public health threats, but also to provide information to the final consumer.

In this article, the results of the ValorMar project, sub-project

"Technological Platform for Integration of the Fish and Fishery Value Chain", have been presented. The project's main goal was to create a software platform for tracing aquaculture and wild fish, and derived manufactured products, from their capture sites to the final consumer. The developed platform and applications allow the storing of information for quality analysis and for informing anyone in the value chain about when and where a product lot suffers a transformation or passes through one of the previously identified stages of the value chain, and who was responsible for it.

This project, and its results, are especially relevant and useful:

- For food safety reasons, respecting mainly fresh fish: the platform allows the authorities to know the location of a particular fish lot to, if necessary, withdraw it from the market.
- For providing traceability information to any business partner in the value chain: the platform enables any industry or storage/transport intermediary in the value chain, through their ERP/MRP applications integrated with the platform, to know the origin of any given fish product lot, when it was produced, stored, transported, who was responsible for each activity, and which previous lots have been used in producing that lot. This enables any business partner in the fish and fisheries value chain to make informed choices when buying fish-based product lots.
- For providing traceability information to the final consumer: the platform allows assessing the integrity of fish-based products, including the freshness of fresh fish because, through the kiosk and the mobile applications, it enables any final consumer to know when, where and how the fish or fish-based product, that is for sale in a retail store, was caught, manufactured, transported and stored. This enables the final consumer to make informed selection of fish product lots.
- For providing nutritional information, especially to the final consumer: the platform provides informative data (taxonomy and nutritional information) to the value chain users, especially the final consumer, allowing him/her to make informed selection of fish products. The informative data is used to enrich the end stories about the products presented to the consumer. This informative data associated with the traceability information is one of the innovations that distinguishes this platform from other traceability related studies. The already existing taxonomy and nutritional information has been inserted in the platform by a biologist, which provides operators with standard information in the cases where operator specific information is not available or not required.

Another project's goal was that the middleware platform would have ease of extensibility of the knowledge base, enabling new types of information to be added in the future. This future information can be stored by the middleware platform or integrated from outside platforms or databases.

The data collected and stored by this platform enables an analysis of consumption trends and of the species caught over time. Once almost all the wild fish caught by the Portuguese fishing fleet is received and registered by DocaPesca (the company responsible for receiving wild fish and organizing fishery auctions, 100% owned by the Portuguese state), a business partner in this project, this platform also allows to control and verify if the fish quotas defined by the European Union are being met.

Future work will focus on the applications for feeding the platform with traceability information, namely further adapting ERP and MRP applications, used by operators in the fish and fishery value chain.

Another future work involves developing a blockchain-based platform, with the same API as the current working platform, to verify its usefulness and study its advantages and disadvantages when compared with the current platform, which does not rely on BlockChain.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Antonio Miguel Rosado da Cruz reports financial support was provided by European Regional Development Fund (ERDF). Jose Oliveira reports financial support was provided by European Regional Development Fund (ERDF). Jose Evaristo Lima reports financial support was provided by European Regional Development Fund (ERDF). Dimitri da Silva reports financial support was provided by European Regional Development Fund (ERDF). Volodymyr Kuprych reports financial support was provided by European Regional Development Fund (ERDF). Pedro Miguel Faria reports financial support was provided by European Regional Development Fund (ERDF). Claudio Teixeira reports financial support was provided by European Regional Development Fund (ERDF). Estrela Ferreira Cruz reports financial support was provided by European Regional Development Fund (ERDF).

Acknowledgments

The authors would like to thank FoodIntech, especially its CEO Miguel Fernandes, and SONAE, especially Senior Project Manager Ana Machado Silva, and also Docapesca and Fórum Oceano for their ideas and contributions to the ValorMar project. This article has been developed in the context of Project "ValorMar – Valorização integral dos recursos marinhos: potencial, inovação tecnológica e novas aplicações" (reference POCI-01-0247-FEDER-024517) funded by the European Regional Development Fund (FEDER-Fundo Europeu de Desenvolvimento Regional) through Operational Programme for Competitiveness and Internationalization (POCI).

References

[1] L. Alves, T. Carvalhido, E.F. Cruz, A.M.R. da Cruz, Using blockchain to trace PDO/PGI/TSG products, in: 23st International Conference on Enterprise Information Systems (ICEIS), SciTePress, 2021, pp. 368–376.
[2] A. Bangor, P. Kortum, J. Miller, Determining what individual SUS scores mean: adding an adjective rating scale, J. Usability Stud. 4 (2009) 114–123.
[3] G. Baralla, A. Pinna, R. Tonelli, M. Marchesi, S. Ibba, Ensuring transparency and traceability of food local products: a blockchain application to a smart tourism region, Concurrency Comput. Pract. Ex. 33 (2021) e5857, https://doi.org/10.1002/cpe.5857. URL, https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5857.
[4] M. Bevilacqua, F. Ciarapica, G. Giacchetta, Business process reengineering of a supply chain and a traceability system: a case study, J. Food Eng. 93 (2009) 13–22.
[5] K. Biswas, V. Muthukkumarasamy, W.L. Tan, Blockchain based wine supply chain traceability system, in: Future Technologies Conference (FTC), 2017, pp. 56–62.
[6] J. Brooke, SUS: a retrospective, Journal of Usability Studies 8 (2013) 29–40.
[7] C. Cheng, P. Jiang, J. Liu2, A common traceability method for agricultural products based on data center, Sens. Lett. 11 (2013).
[8] E.F. Cruz, A.M.R. da Cruz, A food value chain integrated business process and domain models for product traceability and quality monitoring, in: Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS), SCITEPRESS – Science and Technology Publications, Heraklion, Crete, Greece, 2019, pp. 285–294, https://doi.org/10.5220/0007730502850294.
[9] E.F. Cruz, A.M.R. da Cruz, R. Gomes, Analysis of a traceability and quality monitoring platform for the fishery and aquaculture value chain, in: 14th Iberian Conference on Information Systems and Technologies (CISTI 2019), 2019. https://ieeexplore.ieee.org/document/8760755.
[10] E.F. Cruz, A.M. Rosado da Cruz, Design science research for is/it projects: focus on digital transformation, in: 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), 2020, https://doi.org/10.23919/CISTI49556.2020.9140972.
[11] E.F. Cruz, A.M. Rosado da Cruz, Using blockchain to implement traceability on fishery value chain, in: Proceedings of the 15th International Conference on Software Technologies (ICSOFT2020), SCITEPRESS – Science and Technology Publications, 2020, pp. 501–508.
[12] E.F. Cruz, M.Y. Santos, R.J. Machado, Deriving a data model from a set of interrelated business process models, in: Proceedings of the 17th International Conference on Enterprise Information Systems, SCITEPRESS – Science and Technology Publications, 2015, pp. 49–59.
[13] D. da Silva, J. Costa, B. Assunção, V. Kuprych, C. Teixeira, Microservice-based middleware for collaborative supply chain tracing, in: 2020 15th Iberian Conference on Information Systems and Technologies, CISTI, 2020, pp. 1–6, https://doi.org/10.23919/CISTI49556.2020.9141036.

[14] N.P.I.R. Devy, S. Wibirama, P.I. Santosa, Evaluating user experience of English learning interface using user experience questionnaire and system usability scale, in: 2017 1st International Conference on Informatics and Computational Sciences, ICICoS, 2017, pp. 101–106.

[15] D. Folinas, M. Vlachopoulou, V. Manthou, B. Manos, A web-based integration of data and processes in the agribusiness supply chain, in: EFITA Conference, 2003.

[16] F. Food, A. Organization, The State of World Fisheries and Aquaculture. Technical Report, 2020. United Nations.

[17] C.J. Gardner, S. Rocliffe, C. Gough, A. Levrel, R.L. Singleton, X. Vincke, A. Harris, Value Chain Challenges in Two Community- Managed Fisheries in Western Madagascar: Insights for the Small-Scale Fisheries Guidelines, Springer International Publishing AG, 2017.

[18] A. Hevner, S. Chatterjee, Design science research in information systems - theory and Practice, Integrated Series in Information Systems 22 (2010) (Springer).

[19] A.R. Hevner, S.T. March, J. Park, S. Ram, Design science in information systems research, MIS Q. 28 (2004) 75–105, https://doi.org/10.2307/25148625.

[20] K. Indrasiri, P. Siriwardena, APIs, Events, and Streams, Apress, 2018, pp. 293–312, https://doi.org/10.1007/978-1-4842-3858-5_10 (chapter 10), 10.1007/978-1-4842-3858-5_10.

[21] K. Indrasiri, P. Siriwardena, The Case for Microservices, Apress, Berkeley, CA, 2018, pp. 1–18, https://doi.org/10.1007/978-1-4842-3858-5_1 (chapter 1), 10.1007/978-1-4842-3858-5_1.

[22] K. Indrasiri, P. Siriwardena, Designing Microservices, Apress, 2018, pp. 19–61, https://doi.org/10.1007/978-1-4842-3858-5_2 (chapter 2), 10.1007/978-1-4842-3858-5_2.

[23] K. Indrasiri, P. Siriwardena, Inter-Service Communication, Apress, 2018, pp. 63–88, https://doi.org/10.1007/978-1-4842-3858-5_3 (chapter 3), 10.1007/978-1-4842-3858-5_3.

[24] J. Lin, Z. Shen, A. Zhang, Y. Chai, Blockchain and iot based food traceability for smart agriculture, in: Proceedings of the 3rd International Conference on Crowd Science and Engineering, Association for Computing Machinery, New York, NY, USA, 2018, https://doi.org/10.1145/3265689.3265692.

[25] N. Mai, S.G. Bogason, S. Arason, S.V. Árnason, T. Matthiasson, Benefits of traceability in fish supply chain – case studies, Br. Food J. 112 (2010) 976–1002.

[26] F. Miatton, L. Amado, Fairness, transparency and traceability in the coffee value chain through blockchain innovation, in: 2020 International Conference on Technology and Entrepreneurship - Virtual, ICTE-V, 2020, pp. 1–6.

[27] L.M. Moga, Cloud computing based solutions for monitoring the supply chain of fish and fishery products, in: 2017 Eighth International Conference on Intelligent Computing and Information Systems, ICICIS, 2017, pp. 33–38.

[28] C. Narrod, D. Roy, J. Okello, B. Avendaño, K. Rich, A. Thorat, Public–private partnerships and collective action in high value fruit and vegetable supply chains, Food Pol. 34 (2009) 8–15.

[29] C.G. Nicolae, L.M. Moga, G.V. Bahaciu, M.P. Marin, Traceability system structure design for fish and fish products based on supply chain actors needs, Animal Science LX (2017) 353–358.

[30] J. Oliveira, P.M. Faria, A.M. Rosado da Cruz, User experience in kiosk application for traceability of fishery products, in: C. Stephanidis, A. Marcus, E. Rosenzweig, P. L.P. Rau, A. Moallem, M. Rauterberg (Eds.), HCI International 2020 - Late Breaking Papers: User Experience Design and Case Studies, Springer International Publishing, Cham, 2020, pp. 740–751.

[31] J. Oliveira, A.M. Rosado da Cruz, P.M. Faria, Zeroconf network retail kiosk for fish products traceability, in: 15th Iberian Conference on Information Systems and Technologies (CISTI), 2020. Sevilla, Spain.

[32] A. Regattieri, M. Gamberi, R. Manzini, Traceability of food products: general framework and experimental evidence, J. Food Eng. 81 (2007) 347–356.

[33] A.M. Rosado da Cruz, E.F. Cruz, P.M. Moreira, R. Carreira, J. Gomes, J. Oliveira, R. Gomes, On the design of a platform for traceability in the fishery and aquaculture value chain, in: 14th Iberian Conference on Information Systems and Technologies (CISTI 2019), 2019, pp. 1–6, 10. 23919/CISTI.2019.8760891, https://ieeexplore.ieee.org/document/8760891.

[34] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, C. Mortimore, Openid Connect Core 1.0, The OpenID Foundation, 2014.

[35] Z. Sharfina, H.B. Santoso, An Indonesian adaptation of the system usability scale SUS, in: 2016 International Conference on Advanced Computer Science and Information Systems, ICACSIS, 2016, pp. 145–148.

[36] F. Tian, A supply chain traceability system for food safety based on HACCP, blockchain & internet of things, in: International Conference on Service Systems and Service Management, 2017.