On the Design of a Platform for Traceability in the Fishery and Aquaculture Value Chain

António M. Rosado da Cruz[1,2,3], Estrela Ferreira Cruz[1,2], Pedro Moreira[1], Rui Carreira[1], João Gomes[1], José Oliveira[1], Rui Gomes[1]
[1]ARC4Digit, Instituto Politécnico de Viana do Castelo, Viana do Castelo, Portugal
[2]ALGORITMI, Universidade do Minho, Guimarães, Portugal
[3]Corresponding Author: miguel.cruz@estg.ipvc.pt

Abstract — **Food Traceability is a customer's and other value chain operator's call, driven by increasing product quality demands and customers' awareness. Traceability in fisheries and aquaculture value chain is related to the geographical origin and location of products, but also to the continuous measurement of products transport and storage conditions. This paper follows the methodology that is being carried out in the *ValorMar R&D* project in order to conceptualize and design the overall integrated traceability platform architecture and services API.**

Keywords –Food Traceability; Fish and Fisheries Value Chain; Platform Architecture; Services API

I. Introduction

Overall production of fishery products peaked at about 171 million metric tons in 2016, including aquaculture, according to FAO's *The State of World Fisheries and Aquaculture 2018* report [1]. In per capita terms, food fish consumption grew from 9.0 kg in 1961 to 20.2 kg in 2015, at an average rate of about 1.5% per year. Aquaculture represents today 47% of the total fishery products consumption, and 53% if non-food uses (e.g. fishmeal and fish oil) are excluded. With capture fishery production rather static since the 1980s, aquaculture has been responsible for coping with the progress of consumers tastes [1].

Consumers increasingly demand safe and healthy food. One of the EU's concerns is to ensure that food is of high quality for all European citizens, whether they are produced inside or outside the EU. With regard to food safety, the European Commission has for some ten years established food legislation which introduced the concept of traceability. This concept, which specifies that any food product can be traced from the source to the final consumer, is already implemented in some sectors (e.g. meat), but at the level 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 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. In addition, the growing consumer interest in the origin and treatment of the products they consume has put some pressure on the need to trace the fisheries and aquaculture value chain.

This paper presents the design and specification of a fishery and aquaculture products traceability platform, being developed in the scope of *ValorMar R&D* project, which will be the central platform for integrating with systems from the value chain operators to gather traceability and quality information, and with consumer-oriented applications for providing traceability information to the final consumers. This paper presents the value chain integrated business process model and the supporting integrated platform's domain model, but its focus is on the platform's design and architecture.

The main objective of the platform being developed is to gather and centralize all traceability information (among other information relevant to the consumer, such as the date of catch of the fish or the temperatures to which the fish was subjected throughout the chain). It is also intended that the integrated traceability platform integrates the internal traceability data of each of the operators of the value chain, allowing centrally to perceive the entire flow from the producers/catchers of the raw fishery products, through the production units packaging, conveyors, wholesalers and retailers. In addition, the platform may incorporate qualitative data, food safety, among other information, increasing and making this information available to the final consumer, significantly improving their perception of what they are buying and ultimately eating.

The structure of the presentation is as follows: The next section addresses related work; In section iii, conceptual approach and platform requirements are presented; Section iv presents the platform architecture and design; The services API is presented in section v; and, section vi concludes the paper and presents ideas for future developments.

II. Related Work

EU authorities have long proposed directives requiring the disclosure of information about the origin, quality and location traceability of certain products, improving information to consumers and enabling faster recalls when needed [2].

Several proposals and studies have been made to trace information in a food value chain, either focused on fish and fishery products, or on any other food branch. Some, such as [3], were created with the aim of protecting and certifying the origin of products, others to assure the quality and safety of fresh products in general. Regattieri et al. proposed a platform to support the traceability in all the steps performed in the creation of Italian cheese Parmigiano Reggiano, from bovine farms to the final stage of the value chain, packaging [3]. The main purpose of this platform is to certify the protected designation of origin of the Italian cheese, assuring the quality of the product and to

This contribution 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 FEDER (Fundo Europeu de Desenvolvimento Regional) through Operational Programme for Competitiveness and Internationalization (POCI).

2019 14th Iberian Conference on Information Systems and Technologies (CISTI)
19 – 22 June 2019, Coimbra, Portugal
ISBN: 978-989-98434-9-3

avoid forgeries [3]. The implementation is based on a central database that stores data identified in all steps of the value chain and collected semi-automatically with sensors, bar codes, etc..

A platform to support the aquatic foods supply chain has been proposed by Yan et al. [4]. This platform supports the traceability in the supply chain of aquatic products from production to sales, including distribution, and includes information about production, such as water quality monitoring [4]. The authors present the overall and system-level and network structure of the traceability platform. The implemented platform is based on RFID and Electronic Product Code (EPC), and is applied in a case study in China to tilapia products [4].

In [5] the authors propose a platform to track the aquaculture products from the farm to the consumer. The implementation of the platform is based on web services which are prepared to receive data captured by RFID systems. This data is integrated with data collected with wireless sensor networks (WSN) infrastructure. The paper also presents an analysis of the benefits obtained by the introduction of the created platform, based on predefined objectives and the evaluation of KPIs [5].

In [6] the authors propose a structural design for a monitoring tool to support the traceability in Romanian fisheries supply chain. The proposed tool has the purpose of monitoring the safety and quality of fish and fishery products. The paper presents the general scheme of the traceability system.

In [7] the authors present the first step in a project to design and create a Web platform to support food traceability for dairy products. The first phase of the project includes the design of a reference model for a generic dairy supply chain. The authors identified three main phases in the supply chain, namely natural environment, transformation and distribution, and identified the main entities involved in the supply chain and present a class diagram for each one of the entities. The identified entities are: actor, container, material and sample [7].

Folinas et al. proposed a web-based application for modeling agricultural processes and data. The proposal highlights the collaborative effort in modeling and integrating the logistics processes assuring that all business partners have access and share information [8]. The authors start by identifying standards and specifying the information to be exchanged.

III. Traceability Platform Conceptualization

The fisheries traceability platform aims at registering the traceability relevant information about fishery and aquaculture products from the capture, or aquaculture production, to the plate. Thus, the information to be presented to the different audiences must be identified. The data collected by the platform, will be registered and processed centrally, according to the requirements identified, producing information considered relevant for all target audiences, including the final consumer and all logistics chain operators. One needs to characterize these target audiences, as well as the quality and granularity of the information presented to them. Each audience will make use of specific front-end applications, that are outside of the scope of this paper, and integrate with the platform here presented.

From the point of view of the final consumer it is intended that, at the point of sale and via a web or mobile app, the consumer has access to the relevant information about a particular food product that he has acquired or is thinking of acquiring, for example the origin or capture, the sterilization function and if the cold chain was respected throughout the different players of the chain.

From a value chain operator's point of view, the system should be able to gather geographic, quality and action/event information made on each product lot, including which lots were used in producing another lot, and supply that information to every value chain operator, along with alert information every time the cold chain isn't respected or whenever a lot, or part of it, is considered improper for human consumption.

Our approach to designing the traceability platform started by analyzing the value chain operators' business processes that generate, use or transform, or shoot down product lots. From this bulk of business processes, we have abstracted an integrated business process that serves as template for all the other business processes through an approach in all similar to the one followed in [9]. Then, from this integrated business process, we extracted the domain model, by using the approach presented in [10, 11]. The obtained domain model has then been manually refined and is being used as a basis for the *Model* data structures in the platform architecture. The following subsections detail and clarify each of these steps. From the identified platform requirements, we have then specified the set of services that should be present in the platform API, which is presented in section IV, along with the platform architecture specification.

A. Value Chain Business Process Model

The fisheries and aquaculture value chain combines different operators, and these have several defined business processes, in which activities that influence the products in the value chain in a greater or lesser degree are performed. The operators involved already collect most of the traceability information desired in this project. However, only part of this information is currently flowing between chain operators, and it is not yet possible to trace back a product lot to its initial production or catching. The main identified value chain operators are: Ship Owner; Ship/Vessel; Fishery auction trader; Wild fish trading Company; Aquaculture Company; Canned fish Company; Frozen fish Company; Cod fish capture and trading Company; Cod fish salting and drying Company; Cod fish desalting and freezing Company; Logistics Company (Transportation and Storage); Big retailer distribution Center; and, Retailer.

The analysis of each value chain operators' business processes has followed a method similar to the one described in [9] and is presented in [12]. As result of this process, an integrated BPMN business process model has been built (depicted in Fig. 1), which serves as template for all the analyzed business processes. This business process model represents all activities that may happen to a product lot along its way through the value chain. As we can see in Fig. 1, a fish product's lot may be produced in an aquaculture farm or be captured at the sea, river or lake. Once caught, the fish is registered, and the quality is assessed. When the quality is not within the established parameters, the lot is discarded (downed). A lot can be sold multiple times. Between sales the fish may be stored or transformed. Following a sale, there is typically a transportation. At the reception, after a transportation, there is always a quality

2019 14th Iberian Conference on Information Systems and Technologies (CISTI)
19 – 22 June 2019, Coimbra, Portugal
ISBN: 978-989-98434-9-3

assessment, and a new lot may be registered (being associated to the previous lot). Similarly, after a transformation new lots of derived products are created. A fish product's lot may suffer several transformations, such as salting, freezing, cooking, etc.

Value chain operators must be registered and the information about each activity must be stored in the platform's database.

B. Platform Requirements

The traceability platform being designed is mainly focused on gathering events' data on product lots for being able to provide to the final consumer traceability information on the products of the fishery and aquaculture value chain. In this scope, the next main functional requirements have been elicited:

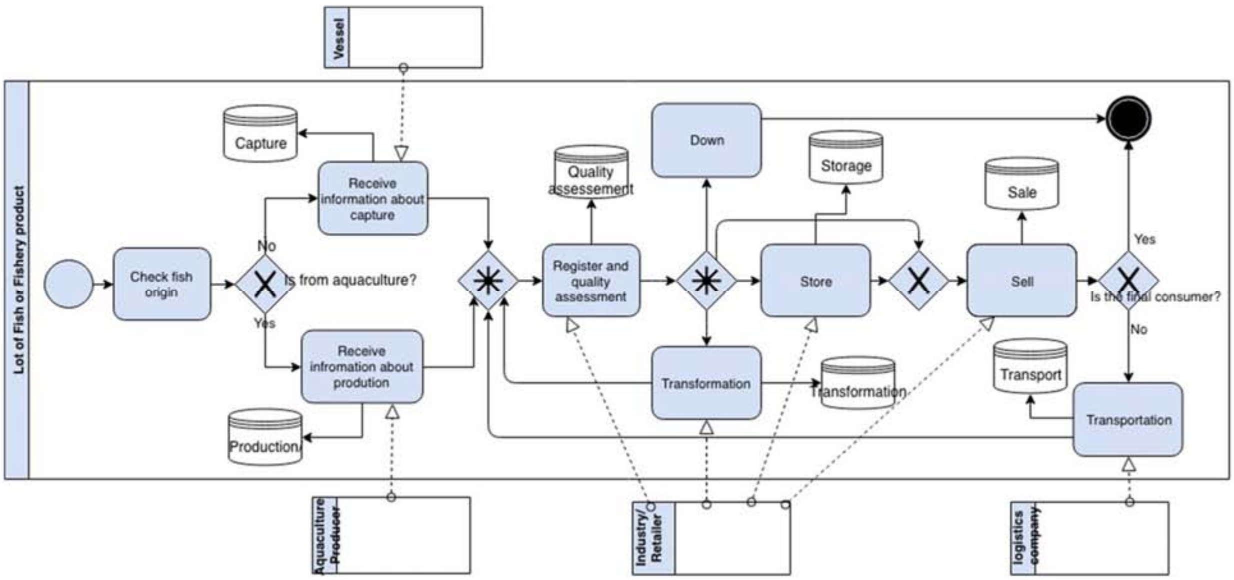

Figure 1. Integrated business process model for the fishery and aquaculture value chain.

There are three types of user profile: **SysAdmin**: system administrator - Administers the shared platform for traceability; **WorkerAdmin**: value chain operator-level system admin - Associated to the operator that they work for and administer; it's a type of Worker; **Worker**: worker from a value chain operator.

1) *A SysAdmin must be able to:*

a) Create, edit and deactivate Value Chain Operators;

b) Create, edit and deactivate WorkerAdmins associated to a value chain operator;

c) Create, edit and deactivate Worker users associated to a value chain operator;

d) Manage global LookUp tables, including product nutrition table information.

2) *A WorkerAdmin must be able to:*

a) Create, edit and deactivate worker users in their own value chain operator;

b) Create, edit and deactivate event records on a lot created or purchased by the value chain operator;

3) *A worker in a value chain operator must be able to:*

a) Create product lots and event records on behalf of the value chain operator to whom they work for;

b) Edit or delete the last event record that they created, in the same session, with the purpose of correcting the information.

Every product has a lot number. This may not be unique at the value chain level, though. Uniqueness rises when the number is combined with the operator that produces the product lot.

4) *A final consumer is an anonymous user that may consult the traceability info of product lot of which they have the lot number and the reference of the operator that produced it.*

5) *A value chain traceability Event may be relative to:* An initial Product Lot registration (Production Registration); Product Lot Sale; Product Lot Transportation; Product Lot Storage; Product Lot Quality Assessment; Product Lot (industrial) Transformation; Product Lot shot down.

6) *A Lot Transformation consumes one or more product lots and produces one or more new product lots;*

7) *A Lot Quality Assessment may give origin to another Lot of the same product (this is because some operators renumber the lots they purchase, upon reception and quality assessment).*

Besides these main functional requirements, a couple of non-functional requirements have been established:

8) *The platform should be able to deal asynchronously with the communication of traceability events.*

9) *The platform should be able to scale horizontally.*

C. Domain Model

From the value chain integrated business process in Fig. 1, we derived the domain model, by using the approach presented in [10, 11] (see Fig. 2). This domain model has then been manually refined taking into account all the structure-related requirements presented in the previous subsection. The resulting domain model is showed in Fig. 3.

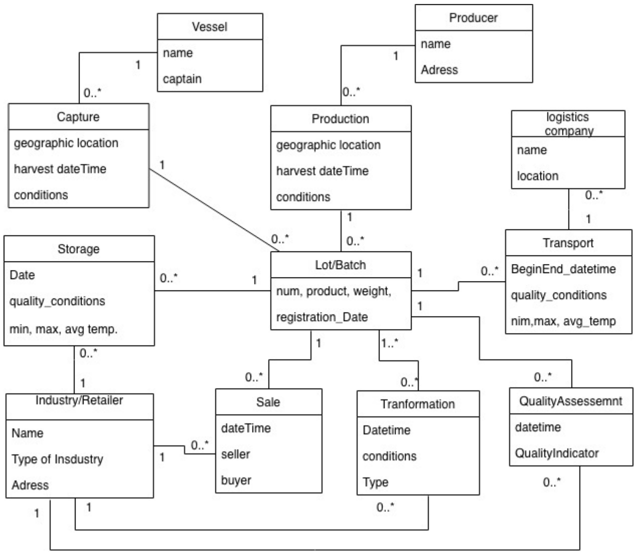

Figure 2. Derived Domain Model.

IV. TRACEABILITY PLATFORM ARCHITECTURE AND DESIGN

Having established the main requirements and the domain model that will drive the design and development of the shared traceability platform, we have made some architectural choices, which have been validated through a developed prototype.

A. *Architectural Decisions*

It is intended that the traceability platform provides a set of services to final consumers' and to value chain operators' applications, which means that it must be able to scale as the number of requests grows. For this:

1) Database technology: We decided to use MongoDB NoSQL database, as it is more apt for simultaneous multiple-user query systems and it is able to scale horizontally, enabling new server nodes to be added to cope with more requests [13];

2) Server technology: The API server technology will be based on Node.js, because of its high performance and efficiency, and again because of its ease of scalability, including to cloud based environments [14];

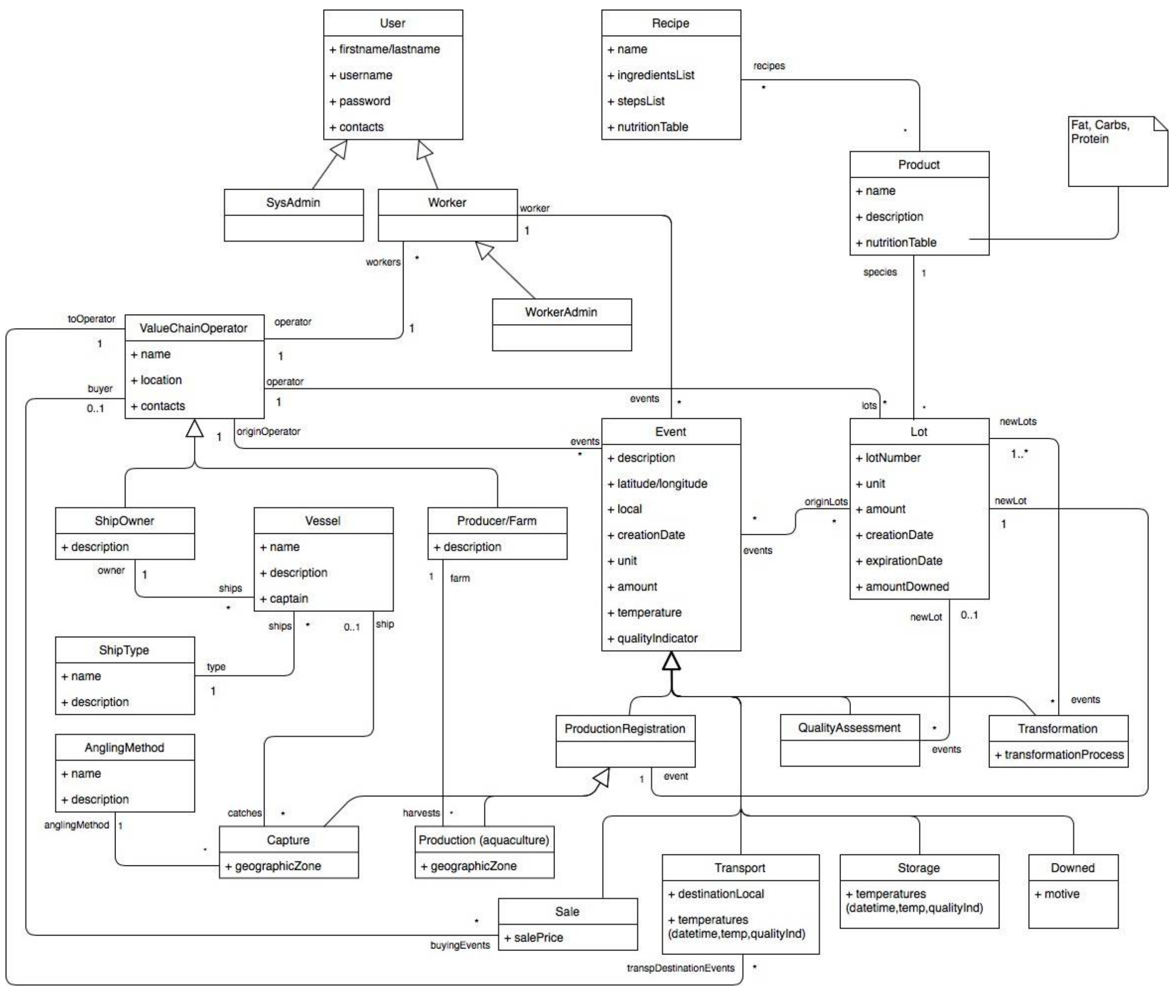

Figure 3. Refined Domain Model.

2019 14th Iberian Conference on Information Systems and Technologies (CISTI)
19 – 22 June 2019, Coimbra, Portugal
ISBN: 978-989-98434-9-3

3) Platform security: Authentication and Authorization at API level makes use of JWT (Json Web Tokens) and of Passport.js as the authorization middleware;

4) API service calls organization: API service calls will be issued in two different ways. Calls for services that create events and lots are pushed onto a Kafka queue for being served by dedicated parallel services that ensure *exactly once* execution of each call; Calls for services that consult information or to less frequent creation/modification services are called directly (synchronously or asynchronously).

The next subsections detail the platform architecture.

B. Platform Architecture - Requests without Apache Kafka

The platform being designed offers a set of directly accessible services. Figure 4 represents the requests that don't use Apache Kafka before reaching the backend platform services. In the figure, there are three servers (implemented with docker containers in the developed prototype), which contain: The Apache Service that will run a frontend client application (e.g. operator or final consumer app); the Node.js backend API application; and, the MongoDB database.

The numbered arrows in the Fig. can be read as follows:

1) Frontend app sends HTTP request to backend endpoint.

2) The backend application starts by checking all the validations and permissions. Then, if everything goes without errors, it executes the requested domain (database) operation.

3) Database returns query results to the backend app.

4) Backend app returns response status to the frontend app.

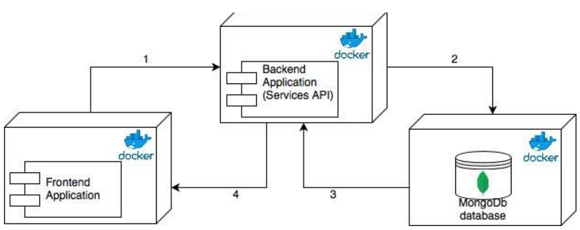

Figure 4. Cloud Architecture without Kafka.

C. Platform Architecture - Requests with Apache Kafka

As mentioned before, the platform being designed makes use of Apache Kafka for frequent creation/modification requests to the API. Figure 5 represents the requests that make use of Kafka before reaching the backend application. In this architectural model, there are five servers (docker containers), which contain: the Apache Service that will run a frontend client application (e.g. operator or final consumer app); the Node.js backend application; the MongoDB database; the Kafka Server; and, a Kafka producer API application that is used to push requests to the Kafka queue, and a Kafka consumer daemon application that sends these requests to the platform backend API.

In this case, the numbered arrows in the Fig. can be read as:

1) Frontend app sends HTTP request to the API service in the Kafka queue producer backend app.

2) Producer app pushes the request into the Kafka queue.

3) Producer responds to the frontend application with the confirmation or non-confirmation that the request has been inserted into the Kafka queue.

4) At any time, the Kafka consumer daemon gets the next request from the queue to send it to the platform backend API.

5) Kafka consumer daemon sends the request to the backend API. Then, waits for the backend app response, before removing the request from the queue. If the request to the backend API times out, the consumer will try the request again. This ensures that all the requests are sent exactly once to the backend app. This doesn't mean that all requests have successful responses, though, because the requested operation pre-conditions may have not been fulfilled.

6) Backend application starts by checking all the validations and permissions. If everything goes without errors, it then executes the requested domain (database) operation.

7) Database returns the query results to the backend app, and this returns to the Kafka consumer daemon, which inserts the request and response status into a domain operations log.

8) and 9) - These two steps represent the user's search for the logs. When a request passes by the Kafka queue, the response confirming the success, or not, of the operation cannot be directly given to the frontend application. To override this problem, a domain operations Log has been implemented. This enables registering the request and response, atached to the user id and request time, every time the platform API responds to the Kafka consumer daemon. With this, a frontend app can list all the logs for taking automatic proper action or just for the user to confirm if a requested operation was successful.

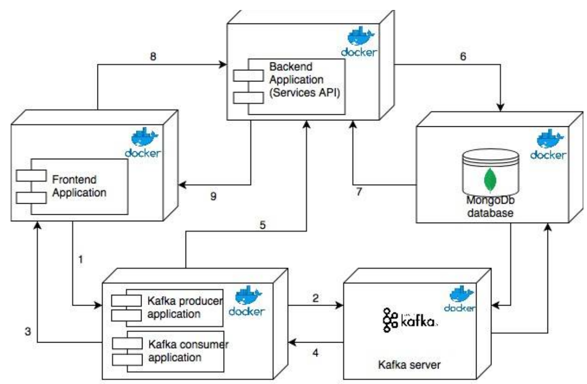

Figure 5. Cloud Architecture with Kafka.

D. Platform Architecture – Authentication and Authorization

When a request is made to the backend API, using or not Apache Kafka, the request will go through a security layer before actually reaching the "core" of the backend application where the request is processed. This security layer consists of Passport.js, an authentication middleware for node.js, that generates access tokens (JWT) for the API users and thereby controls their access time and authorized operations. This access token is generated when a user logs into the system.

2019 14th Iberian Conference on Information Systems and Technologies (CISTI)
19 – 22 June 2019, Coimbra, Portugal
ISBN: 978-989-98434-9-3

Then, whenever a user makes a request to the backend, the token that is sent in the request is decoded and the token access time is checked. The next step is to decrypt the token, verify its signature (its validity), and verify which user is assigned to it. If all the validations are correct, process the request made. If the request is made through Kafka, and an authentication, authorization or session time error occurs, the operation is removed from the Kafka queue, but an operation error will be pushed into the domain operations log.

E. Traceability Platform Services API

The platform API is organized according to the REST architectural style [15]. This entails that services are identified by the handled resource name, and access methods have a uniform semantics for all resources, being associated to the HTTP verbs: **POST**, validates parameters and creates the resource object and objects of other associated resources not directly accessible through the API (e.g. when registering event *ProductRegistration*, a *Lot* is also created); **GET**, gets one or several resource objects, depending on the parameters; **PUT**, updates a resource object; **DELETE**, deletes a resource object.

The security layer ensures that every method starts by looking for the JWT token on the HTTP authorization header, to verify if the user is authenticated and authorized for the respective operation, and only then routes the execution to the proper action. Table II lists the services being developed for the traceability platform.

TABLE I. API MAIN SERVICES

Service Name (resource)	*HTTP Verb*	*Author-ized User Types*[1]	*Parameters*	*Result*
user	POST	sa, wa	User object	The created object
	GET	sa, wa, w	Filtering fields	User or list of users
	PUT	sa, wa	User object	Updated object
	DELETE	sa, wa	objectID	Deleted object
valueChain Operator	POST	sa	valueChainOperator object	The created object
	GET	sa, wa, w	Filtering fields	valueChainOperator or list of valueChain operators
	PUT	sa	valueChainOperator object	Updated object
	DELETE	sa	objectID	Deleted Object
event	POST	wa, w	event object	The created object
	GET	any	Filtering fields	Event or list of events
	PUT	wa, w	event object	Updated object
	DELETE	wa, w	objectID	Deleted object
login	GET	any	username, password	JWT token
traceBack	GET	any	valueChainOperator ID, lot number	List of Events

1. SysAdmin = sa; WorkerAdmin = wa; Worker = w.

V. CONCLUSIONS

This paper presents the design methodology used to develop a services platform for traceability in the fisheries and aquaculture value chain. The platform will be the central piece for integrating with systems from value chain operators, to gather traceability and quality information, and with consumer-oriented applications, for providing traceability information to the final consumers.

Relatively to the state of the art, and to the best of our knowledge, the artifacts presented here are a novelty, including the value-chain level business-process model and the platform informational-structure-supporting domain model, but also the platform's architecture and API specification.

The next steps include the platform implementation and deployment and the development of value chain operators' and consumers' applications that integrate with the central platform.

REFERENCES

[1] FAO. The State of World Fisheries and Aquaculture 2018 – Meeting the sustainable development goals. Rome. Lic CC BY-NC-SA 3.0 IGO, 2018.

[2] UNION, E.. Regulation (EC) no 178/2002 of the european parliament and of the council. Tech. report, Official Journal of the European Com., 2002.

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

[4] Yan, B., Shi, P., and Huang, G.. Development of traceability system of aquatic foods supply chain based on RFID and EPC internet of things. Transactions of the Chinese Society of Agricultural Engineering, 29(15):172–183, 2013.

[5] Alfredo Parreño-Marchante, Alejandro Alvarez-Melcon, M. T. P. F.. Advanced traceability system in aquaculture supply chain. Journal of Food Engineering, Volume 122:99–109, 2014.

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

[7] Ioannis Manikas, B. M.. Design of an integrated supply chain model for supporting traceability of dairy products. Int'l journal of dairy tech., 2009.

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

[9] Cruz, E. F., Cruz, A. M. R.. A Food Value Chain Integrated Business Process and Domain Models for Product Traceability and Quality Monitoring: Pattern models for Food Traceability Platforms. In Proc. of the 21st ICEIS, Heraklion, Crete-Greece, 2019 (to appear).

[10] Cruz, E. F., Machado, R. J., and Santos, M. Y.. From business process modeling to data model: A systematic approach. In QUATIC 2012, Thematic Track on Quality in ICT Requirements Engineering, IEEE Computer Society Press, Los Alamitos, California, U.S.A., 205–210. IEEE Computer Society, 2012.

[11] Cruz, E. F., Santos, M. Y., and Machado, R. J. Deriving a data model from a set of interrelated business process models. In 17th International Conf. on Enterprise Information Systems, pages 49–59, 2015.

[12] Cruz, E. F., Cruz, A. M. R.., Gomes R.. Analysis of a Traceability and Quality Monitoring Platform for the Fishery and Aquaculture Value Chain. In I International Workshop on Digital Transformation in the Enterprise, CISTI 2019, Coimbra, Portugal (to appear).

[13] S. Agarwal and K. Rajan. Analyzing the performance of NoSQL vs. SQL databases for Spatial and Aggregate queries. In Proc. of Conf. Free and Open Source Software for Geospatial (FOSS4G), vol. 17, Art. 4, 2017.

[14] Chaniotis, I.K., Kyriakou, KI.D. & Tselikas, N.D. Is Node.js a viable option for building modern web applications? A performance evaluation study. In journal Computing 97: 1023, Oct., 2015.

[15] Fielding, R. T., Taylor, R. N., Erenkrantz, J. R., Gorlick, M. M., Whitehead, J., Khare, R., and Oreizy, P.. Reflections on the REST architectural style and "principled design of the modern web architecture" (impact paper award). In Proc. of the 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017). ACM, NY, USA, 4-14, 2017.

2019 14th Iberian Conference on Information Systems and Technologies (CISTI)
19 – 22 June 2019, Coimbra, Portugal
ISBN: 978-989-98434-9-3