



The Food Safety Market: An SME-powered industrial data platform to boost the competitiveness of European food certification

D3.1 – TheFSM Open Reference Architecture



DELIVERABLE NUMBER	D3.1
DELIVERABLE TITLE	TheFSM Open Reference Architecture
RESPONSIBLE AUTHOR	Danai Vergeti (UBITECH)



Co-funded by the Horizon 2020
Framework Programme of the European Union

GRANT AGREEMENT N.	871703
PROJECT ACRONYM	TheFSM
PROJECT FULL NAME	The Food Safety Market: An SME-powered industrial data platform to boost the competitiveness of European food certification
STARTING DATE (DUR.)	01/01/2020 (36 months)
ENDING DATE	31/12/2023
PROJECT WEBSITE	www.foodsafetymarket.eu
COORDINATOR	Nikos Manouselis
ADDRESS	110 Pentelis Str., Marousi, GR15126, Greece
REPLY TO	nikosm@agroknow.com
PHONE	+30 210 6897 905
EU PROJECT OFFICER	Stefano Bertolo
WORKPACKAGE N. TITLE	WP3 Platform
WORKPACKAGE LEADER	UBITECH
DELIVERABLE N. TITLE	D3.1 – TheFSM Open Reference Architecture
RESPONSIBLE AUTHOR	Danai Vergeti (UBITECH)
REPLY TO	vergetid@ubitech.eu
DOCUMENT URL	
DATE OF DELIVERY (CONTRACTUAL)	30 April 2021 (M15)
DATE OF DELIVERY (SUBMITTED)	31 May 2021 (M16)
VERSION STATUS	2.0 Final
NATURE	Report (R)
DISSEMINATION LEVEL	Public (P)
AUTHORS (PARTNER)	Danai Vergeti (UBITECH), Iosif Angelidis (UBITECH), Dimitris Ntalaperas (UBITECH)
CONTRIBUTORS	Giannis Stoitsis (Agroknow), Timos Lanitis (Agroknow), Nikos Manouselis (Agroknow), Svetla Boytcheva (SAI), Pavlin Gyurov (SAI), Branimir Rakic (PROSPEH), Tanja Matosevic (AGRIVI)
REVIEWER	Giannis Stoitsis (Agroknow)

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Table of contents	20/04/2021	Danai Vergeti (UBITECH), Dimitris Ntalaperas (UBITECH)
0.3	Introduction, relation to other deliverables, initial text for subsections	25/04/2021	Danai Vergeti (UBITECH), Iosif Angelidis (UBITECH), Dimitris Ntalaperas (UBITECH), Tanja Matosevic (AGRIVI), Timos Lanitis (Agroknow), Nikos Manouselis (Agroknow), Giannis Stoitsis (Agroknow)
0.4	Requirements extraction (functional, non-functional, technical, user stories mapping to technical requirements)	03/05/ 2021	Danai Vergeti (UBITECH), Iosif Angelidis (UBITECH), Dimitris Ntalaperas (UBITECH), Braminir Rakic (PROSPEH)
0.5	Documentation of conceptual and technical architecture	07/05/2021	Danai Vergeti (UBITECH), Iosif Angelidis (UBITECH), Dimitris Ntalaperas (UBITECH), Braminir Rakic (PROSPEH)
0.6	Contributions to architectural components	14/05/2021	Danai Vergeti (UBITECH), Iosif Angelidis (UBITECH), Dimitris Ntalaperas (UBITECH), Braminir Rakic (PROSPEH), Tanja Matosevic (AGRIVI), Timos Lanitis (Agroknow), Nikos Manouselis (Agroknow), Giannis Stoitsis (Agroknow)
0.7	Conclusion	18/05/2021	Danai Vergeti (UBITECH), Iosif Angelidis (UBITECH), Dimitris Ntalaperas (UBITECH)
0.8	Minor corrections and contributions to all sections	23/05/2021	Danai Vergeti (UBITECH), Iosif Angelidis (UBITECH)
0.9	Minor corrections and contributions to all sections	25/05/2021	Danai Vergeti (UBITECH), Iosif Angelidis (UBITECH)
0.95	Internal review	28/05/2021	Giannis Stoitsis (AGROKNOW)
1.0	Final version	30/05/2021	Danai Vergeti (UBITECH)

PARTNERS		CONTACT
Agroknow IKE (Agroknow, Greece)		Nikos Manouselis (Agroknow) nikosm@agroknow.com
SIRMA AI EAD (SAI, Bulgaria)		Svetla Boytcheva (SAI) svetla.boytcheva@ontotext.com
GIOUMPITEK MELETI SCHEDIASMOS YLOPOIISI KAI POLISI ERGON PLIROFORIKIS ETAIREIA PERIORISMENIS EFTHYNIS (UBITECH, Greece)		Danai Vergeti (UBITECH) vergetid@ubitech.eu
AGRIVI DOO ZA PROIZVODNJU, TRGOVINU I USLUGE (Agrivi d.o.o., Croatia)		Tanja Matosevic (Agrivi d.o.o.) tanja.matosevic@agrivi.com
PROSPEH, POSLOVNE STORITVE IN DIGITALNE RESITVE DOO (PROSPEH DOO, Slovenia)		Ana Bevc (PROSPEH DOO) ana@origin-trail.com
UNIVERSITAT WIEN (UNIVIE, Austria)		Tima Anwana (UNIVIE) tima.anwana@univie.ac.at
STICHTING WAGENINGEN RESEARCH (WFSR, Netherlands)		Yamine Bouzembrak (WFSR) yamine.bouzembrak@wur.nl
TUV- AUSTRIA ELLAS MONOPROSOPI ETAIREIA PERIORISMENIS EUTHYNIS (TUV AU HELLAS, Greece)		Kostas Mavropoulos (TUV AU HELLAS) konstantinos.mavropoulos@tuv.at
TUV AUSTRIA ROMANIA SRL (TUV AU ROMANIA, Romania)		George Gheorghiu (TUV AU Romania) george.gheorghiu@tuv.at
VALORITALIA SOCIETA PER LA CERTIFICAZIONE DELLE QUALITA'E DELLE PRODUZIONI VITIVINICOLE ITALIANE SRL (VALORITALIA, Italy)		Francesca Romero (Valoritalia) francesca.romero@valoritalia.it
TUV AUSTRIA CYPRUS (TUV AU CYPRUS, Cyprus)		Sousanna Charalambidou (TUV AU CYPRUS) sousanna.charalambidou@tuv.at

ACRONYMS LIST

A2C	Advantage Actor Critic
ABAC	Attribute-Based Access Controller
ABE	Attribute-Based Encryption
ACL	Access Control Lists
API	Application Programming Interface
CRM	Customer Relationship Management
CSP	Cloud Service Provider
DID	Decentralized Identifiers
DLT	Distributed Ledger Technologies
DoA	Description of Action
DSS	Decision Support System
EPCIS	Electronic Product Code Information Services
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
FSQA	Food Safety and Quality Assurance
IaaS	Infrastructure-as-a-Service
IoT	Internet of Things
JSON	Javascript Object Notation
LOD	Linked Open Data
LSSS	Linear Sharing Secret Schemes
OASIS	Organization for the Advancement of Structured Information Standards
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
RBAC	Role-Based Access Control
RDBMS	Relational Database Management System
RDF	Resource Description Framework
SME	Small & Medium Enterprises
SOML	Semantic Object Model Language
SSE	Symmetric Searchable Encryption
TEE	Trusted Execution Environment
TheFSM	The Food Safety Market
UID	Unique Identifier
UML	Unified Modeling Language
WoT	Web of Things
XACML	eXtensible Access Control Markup Language

EXECUTIVE SUMMARY

The purpose of the deliverable D3.1 “TheFSM Open Reference Architecture” is to deliver the second version of the conceptual architecture of the TheFSM platform. Towards this, the Agile development methodology has been adopted, as already introduced in D3.1 (v1.0, M9), based on which during M9-M15 the second iteration of TheFSM Platform Architecture definition takes place.

Whereas the first version of TheFSM Platform Architecture focused on the methodology definition, the user requirements analysis and the mapping of the technical requirements to the conceptual architecture, the second version of the TheFSM Platform Architecture is to further refine and enhance TheFSM Platform Architecture based on the outputs of T3.1-T3.6 of the first iteration of the TheFSM Platform, which delivered the first version of TheFSM Platform prototype. Additionally, the second iteration, also revisits the user requirements, through the second version of the user requirements, as they are delivered in the second iteration of T1.1-T1.4 and T1.5 and ensures that all the relevant updates on the features of TheFSM Platform are taken under consideration. Within this context, the scope of the current report can be described in the following axes:

- To present a comprehensive documentation of the architecture of TheFSM Platform. A brief description of each component is presented focusing on their positioning within the platform’s architecture. For each component, the platform functionalities that are undertaken by this component are described and the component’s interactions with the rest of the components for the realisation of these functionalities is documented.
- To provide the updated documentation of the components of the TheFSM platform. For each component of the integrated TheFSM Platform, the core functionalities that the component offers are described. In addition to this, the involvement of each component in the TheFSM platform’s services and in the designed platform’s workflows is presented. Furthermore, for each component, the interactions with the rest of the components, as well as the interfaces that are offered in order to facilitate the required exchange of information, are presented. Finally, the technical details of these interfaces are documented and presented in the technical view of TheFSM Platform Architecture.

The design of the architecture is a living and iterative process that will last until M36 as per the Description of Action. Thus, D3.1 constitutes a living document that will include the updates that will be based on further identified functional requirements translated into technical requirements, originating mainly from the evaluation and feedback received from the pilot partners, and will introduce updates and refinements in TheFSM Architecture presented in the upcoming versions of this deliverable.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	6
1 INTRODUCTION.....	11
1.1. SCOPE.....	11
1.2. AUDIENCE.....	12
1.3. STRUCTURE.....	12
1.4. RELATION TO OTHER DELIVERABLES	13
2 METHODOLOGY.....	14
2.1. UPDATES FROM THE PREVIOUS VERSION	14
2.2. WORKING METHODOLOGY	14
2.3. THEFSM DATA GOVERNANCE MODEL	15
2.4. THEFSM USER STORIES	16
2.5. THEFSM STAKEHOLDERS AND INTERACTIONS.....	16
3 THEFSM PLATFORM ARCHITECTURE.....	21
3.1. CONCEPTUAL ARCHITECTURE.....	21
3.2. THEFSM PLATFORM MAIN WORKFLOWS.....	23
3.2.1. Authentication and Authorization	25
3.2.2. Dataset Upload.....	26
3.2.3. Query search	29
3.2.4. Monetization	30
3.2.5. API Gateway	31
3.3. THEFSM ARCHITECTURAL COMPONENTS	35
3.3.1. Data Sources.....	35
3.3.2. Data Curation and Semantic Enrichment.....	37
3.3.3. Data Processing.....	40
3.3.4. Identity Management	46
3.3.5. Automated Contract Negotiation and Monetization.....	54

3.3.6.	Security and Access Control	58
3.3.7.	Data Marketplace and Added Value Services	67
3.3.8.	TheFSM Extended Applications.....	68
4	THEFSM TECHNICAL ARCHITECTURE	72
5	CONCLUSIONS	76
6	BIBLIOGRAPHY	77
	ANNEX I USER STORIES	78
	ANNEX II USER STORIES – RANKING	86
	ANNEX III THEFSM FUNCTIONAL REQUIREMENTS.....	87
	ANNEX IV THEFSM NON-FUNCTIONAL REQUIREMENTS	99
	ANNEX V THEFSM TECHNICAL REQUIREMENTS	101
	ANNEX VI THEFSM TECHNICAL REQUIREMENTS MAPPING TO USER STORIES....	108
	ANNEX VII THEFSM COMPONENTS MAPPING TO TECHNICAL REQUIREMENTS .	111

LIST OF TABLES

Table 1 TheFSM stakeholders involved in supply-chain procedures.....	17
Table 2 TheFSM stakeholders involved in certification procedures.....	18
Table 3 TheFSM Data market stakeholders	18
Table 4 TheFSM Stakeholders and the relevant TheFSM technical solution	20
Table 5 Activity diagrams workflows mapping to data governance model	24
Table 6 TheFSM user stories	85
Table 7 User stories ranking and heatmap – 2nd iteration	87
Table 8 Functional requirements – 2nd iteration.....	97
Table 9 Non-Functional requirements – 2nd iteration.....	101
Table 10 Technical requirements – 2nd iteration.....	107
Table 11 Technical requirements mapping to user stories	111
Table 12 Components mapping to technical requirements	112

LIST OF FIGURES

Figure 1 Working methodology of D3.1, second iteration	15
Figure 2 TheFSM Data Governance Model	16
Figure 3 TheFSM Reference Architecture	21
Figure 4 Authentication and Authorization	25
Figure 5 Dataset upload	28
Figure 6 Query search	29
Figure 7 Monetized dataset purchase.	30
Figure 8 API Gateway request execution activity diagram.....	31
Figure 9 API Gateway request execution sequence diagram.....	32
Figure 10 API Gateway new endpoint registration activity diagram.....	34
Figure 11 API Gateway new endpoint registration sequence diagram.....	35
Figure 12 TheFSM Platform Data sources	36
Figure 13 T Data types supported by TheFSM.....	36
Figure 14 High-level architecture of Data Curation and Enrichment	37
Figure 15 High level architecture of semantic mapper	38
Figure 16 Data ingestion sequence diagram	39
Figure 17 Data retrieval sequence diagram.....	45
Figure 18 DID Architecture	47
Figure 19 DID actions overview	48
Figure 20 Verifiable Credentials ecosystem	49
Figure 21 DID resolution	50
Figure 22 DID provisioning.....	50
Figure 23 DID resolution and verification	51
Figure 24 High level presentation of OriginTrail.....	52
Figure 25 High level architecture of OriginTrail framework.....	53
Figure 26 Sequence diagram illustrating the workflow for publishing traceability events.....	54
Figure 27 Sequence diagram illustrating the workflow of querying traceability data.	54
Figure 28 Activity diagram illustrating the data monetization process.....	57
Figure 29 Sequence diagram illustrating the FairSwap protocol.....	58
Figure 30 PERM meta-model	60
Figure 31 RSA encryption represented as a lock.	62
Figure 32 Symmetric encryption represented as a lock.	63
Figure 33 Pseudonymisation/Anonymisation process sequence diagram.....	65
Figure 34 FSI Data Platform architecture	69
Figure 35 AGRIVI's high-level architecture	71
Figure 36 TheFSM technical architecture	72

1 INTRODUCTION

1.1. Scope

The scope of D3.1 is to document the efforts undertaken during the second iteration within the context of Tasks T3.1 "Architecture and Specifications". The second version of the deliverable D3.1 builds on top of the results of the first iteration of "Task 3.1 Architecture & Specifications", "Task 3.2 User, Identity & Application Management Layers", "Task 3.3 Secure Storage & Information Exchange", "Task 3.4 Blockchain-powered Smart Contracting Layer Secure Storage & Information Exchange", "Task 3.5 Data Management, Indexing & Processing" and "Task 3.6 Technical Verification & Integration Testing", as they are documented in D3.1 (M9), D3.2 (M12), D3.3(M12). Additionally, the second iteration, also revisits the user requirements, through the second version of the user requirements, as they are delivered in the second iteration of T1.1-T1.4 and T1.5 and ensures that all the relevant updates on the features of TheFSM Platform are taken under consideration. Within this context, the scope of the current report can be described in the following axes:

- To present a comprehensive documentation of the architecture of TheFSM Platform. A brief description of each component is presented focusing on their positioning within the platform's architecture. For each component, the platform functionalities that are undertaken by this component are described and the component's interactions with the rest of the components for the realisation of these functionalities is documented.
- To provide the updated documentation of the components of the TheFSM platform. For each component of the integrated TheFSM Platform, the core functionalities that the component offers are described. In addition to this, the involvement of each component in the TheFSM platform's services and in the designed platform's workflows is presented. Furthermore, for each component, the interactions with the rest of the components, as well as the interfaces that are offered in order to facilitate the required exchange of information, are presented. Finally, the technical details of these interfaces are documented and presented in the technical view of TheFSM Platform Architecture.

Towards this end, the current deliverable presents TheFSM conceptual architecture as this is defined through a solid iterative methodology which ensures the coverage of the updated end user needs, as they are defined in Wp1 (D1.1 v2.0, M15) and refines the first version of TheFSM Platform Architecture based on the technical evaluation of the first version of the TheFSM Platform (D3.3, M12). D3.1 v2.0 presents the updated documentation on the design and specifications of TheFSM Platform components towards the successful implementation of the designed workflows. However, all the tasks of WP3 remain active until M36 and during this period, the identification and analysis of additional functional and non-functional requirements, as well as their translation into technical requirements, is a living process and the design and specifications of the TheFSM platform's architecture and components will be constantly updated and documented in the upcoming versions of this deliverable.

The architecture is composed of a set of modular components and addresses the needs of all the different stakeholders of the platform as expressed into the identified technical requirements, while enabling the desired scalability, interoperability and extensibility, and preserving a considerable degree of implementation-platform-independence. Additionally, the current document provides the updated requirements backlog, containing the functional, the non-functional and the technical requirements backlog, that are maintained during the project implementation in order to guide all development tasks.

1.2. Audience

D3.1 targets the consortium members of the TheFSM project and especially the technical partners which participate in WP2, WP3, WP4 with the scope to provide and maintain the conceptual architecture of TheFSM Data Platform which guides the development and the integration of TheFSM Platform. Additionally, an audience outside the consortium, with technical background (s/w engineers, developers, architects etc.) can also follow and understand the methodology, the technical requirements and the architecture of TheFSM Data Platform as it is presented in the current document.

Nevertheless, the document can be easily read by the non-technical partners of the consortium, since it also presents a solid iterative methodology which ensures the translation of the end user needs, as they are defined in Wp1 (D1.1) to technical requirements and the relevant architectural components.

1.3. Structure

The current document is structured as follows:

- Section 1: Introduces the deliverable by presenting its scope, the relevant audience, the document structure and the relation to other deliverables.
- Section 2: The updates from the first iteration of D3.1 with respect to the methodology and the current methodology followed are discussed. Initially, the updates from the previous version are presented, as well as the currently used methodology. Moreover, the data governance model, user stories, stakeholders and interactions are thoroughly documented.
- Section 3: The conceptual architecture of the platform is thoroughly discussed. More specifically, updates to existing components are discussed and new components introduced are thoroughly documented, while both are presented in general, for the sake of completeness. Activity and sequence diagrams illustrating certain key workflows are also provided here.
- Section 4: The technical architecture is presented here. It delves deeper into finer details and is much closer to the actual implementation. To make its understanding a bit easier, we draw comparisons between the technical architecture's components and the conceptual's.
- Section 5: Concludes the deliverable. It outlines the main findings of the deliverable which will guide the future research and technological efforts of the consortium.
- Annex I: Lists the final user stories.

- Annex II: Lists the final user stories – end users and technical partners ranking.
- Annex III: Lists the final functional requirements.
- Annex IV: Lists the final non-functional requirements.
- Annex V: Lists the final technical requirements.
- Annex VI: Lists the mappings between technical requirements and user stories, for the sake of completeness.
- Annex VII: Lists the mappings between components and technical requirements, for the sake of completeness.

1.4. Relation to other deliverables

The current deliverable documents the preliminary efforts undertaken within the context of Tasks T3.1 “Architecture and Specifications”, presenting the overall architecture (both conceptual and technical) of the project’s infrastructure, as it evolved since the previous version of the very same deliverable on M9. The main input of D3.1 is D3.1 v1 - Open Reference Architecture, as well as D1.1 - Report on Requirements for TheFSM and D1.2 - TheFSM Development Roadmap. D3.1 v2 provides updates on the conceptual architecture of the project, while presenting updates on components based on new requirements that appeared throughout the project’s development. Additionally, we report the updated functional, non-functional and technical requirements (compared to the outcome of the requirement analysis of T1.1-T1.4 which was documented in D1.1), the mappings between technical requirements and user stories, while we also present the technical architecture. Finally, the updates on the components of the architecture have taken into consideration requirements of the pilots and the work of WP5 and WP6.

2 METHODOLOGY

2.1. Updates from the previous version

The current deliverable documents the outcomes of the second iteration of T3.1 “Architecture and Specifications. Whereas the first version of TheFSM Platform Architecture focused on the methodology definition, the user requirements analysis and the mapping of the technical requirements to the conceptual architecture, the second version of the TheFSM Platform Architecture is to further refine and enhance TheFSM Platform Architecture based on the outputs of T1.1-T1.4, T3.1-T3.6 of the first iteration of the TheFSM Platform, which delivered the first version of TheFSM Platform prototype. We identify the following updates with regards to the previous version:

- TheFSM Platform Architecture v1.0 (M9) used as an input the Development Roadmap v1.0 (M4). In the current iteration, the Development Roadmap v2.0 takes as input TheFSM Platform Architecture v2.0 and complements the current documentation by providing a more mature and clear view of TheFSM Platform Releases and planning of the technical activities. This is due to the fact that the activities of T1.5 and T3.1 are taking place in parallel and deliver both their outcomes on M15.
- TheFSM Platform Architecture v2.0, apart from the input of T1.1-T1.4, is heavily based on the evaluation of technical evaluation of TheFSM Platform v1.0 (D3.3, M12) and TheFSM Platform Architecture v1.0 (D3.1, M9), making use the first prototype of TheFSM Platform, which was not available during the definition of TheFSM Platform Architecture v1.0.
- TheFSM Platform Architecture v2.0 introduces the Technical Architecture v2.0, as well as, the main workflows that are supported by the platform regarding data processing, data sharing and data monetization.
- TheFSM stakeholders are slightly updated by being mapped to the relevant technical solutions of TheFSM ecosystem.
- TheFSM Data Governance Model is refined
- The functional, non-functional, technical requirements are updated and provided in the document Annexes.
- The main workflows addressing the TheFSM Data Governance Model are documented
- Updates and refinements of the architectural components provided in TheFSM Platform Architecture v1.0

2.2. Working methodology

TheFSM conceptual architecture is the main artefact which will guide the technical development, the integration and the successful delivery of the TheFSM platform. TheFSM platform, besides addressing the technical challenges of an **industrial data platform**, also needs to address the needs of the stakeholders of TheFSM value chain. Towards this, the consortium adopted an **agile methodology** which ensures that the user requirements are taken under consideration in the architectural design and throughout the implementation. The details of this methodology (instruments, roles, process) were presented in D3.1 (v1.0, M9).

The TheFSM agile method allows an architecture refinement process as described in Figure-3.

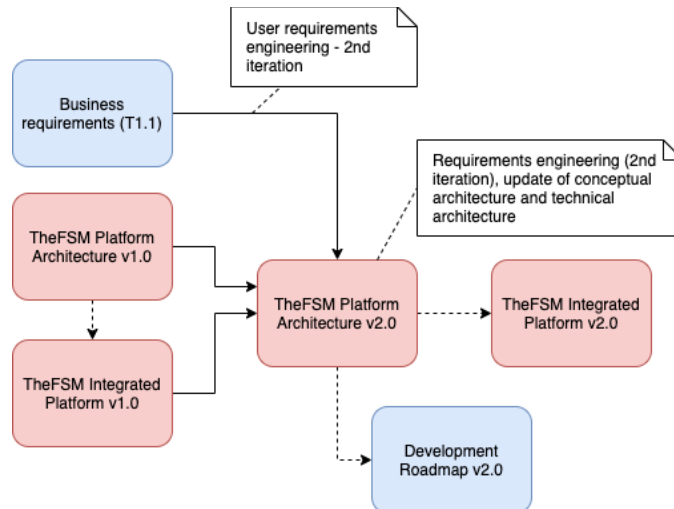


Figure 1: Working methodology of D3.1, second iteration

Following the collaborative and overlapping at work package level approach between WP1 and WP3 that was already introduced in D3.1 (v1.0, M9) the second version of the architecture was identified through an iterative process of requirements engineering and architecture refinement. The **second iteration of the requirements engineering**, deriving from T1.1-T1.4, documented in D1.1 (v2.0, M15), provided the **updated features of the TheFSM Platform and Applications**, while the **technical evaluation of TheFSM Platform** (v1.0, M12) provided with the **technical updates** of the TheFSM Platform that were also taken under consideration during the architecture refinement. The second version of TheFSM Platform Architecture is described by the **Conceptual Architecture**, the **dynamic view of the Conceptual Architecture** and the **Technical Architecture**.

The agile requirements engineering process is proved in D3.1, section 2. Below, the updated outcomes of each agile instrument are presented.

2.3. TheFSM Data Governance Model

TheFSM Data Governance Model provides a well-constructed data management roadmap that is implemented by TheFSM Platform in order to ensure that all data processing adheres to the required technical safeguards (security, curation, provenance, ownership etc.), while at the same time addressing the needs of TheFSM value chain. TheFSM Data Governance Model foresees 5 main phases of data management:

- **Data Collection:** involves collection of data from the supply-chain driven perspective of the data providers. The data asset collection approach that is provided by TheFSM concerns files upload / exchange or data provided through an API.
- **Data IPR and ownership:** addresses the aspects of data asset ownership and licencing that the data provider defines for the shared dataset.
- **Data Curation:** addresses the data quality, data validation, data cleaning, data completion, data harmonization and enrichment perspectives to be applied in TheFSM. In particular, the data harmonization process aims at ensuring that the varying file formats, data

schemas and structures with which the data assets comply are transformed in such a way that they become compatible, consumable, FAIR, reusable using specific standards of the food safety sector and thus valuable for reuse, analysis and knowledge extraction.

- **Data security:** refers to different layers for data security and privacy assurance: (a) end-to-end hybrid encryption for data assets (before, during and after their uploading in the TheFSM platform) and secure tunnels for direct key sharing to authorized data consumers with active data contracts, (b) attribute-based access control policies that formally describe the circumstances under which access requests to data assets should be granted, and are easily interpretable into policy enforcement rules; (c) multiple data anonymization methods and guidelines for data providers.
- **Data monetization:** refers to the configuration, as well as, the wallet management for the data asset trading between the data provider and the data consumer. Transparency, immutability and security of the transactions.

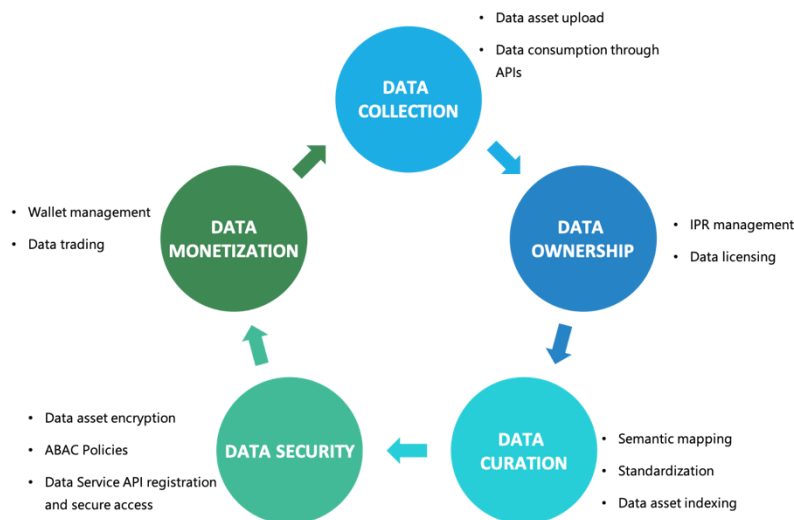


Figure 2: TheFSM Data Governance Model

Each stage described above is implemented from TheFSM Platform and it is documented in the relevant workflows provided in Section 3.2.

2.4. TheFSM User Stories

The list of the user stories and their ranking is provided in Annex I for completeness. More details about the user stories definition and the ranking methodology are provided in D3.1 (v1.0, M9). No major updates were identified during the second iteration in the user stories, since they express the higher expectations of the end users from the TheFSM Platform and the Applications.

2.5. TheFSM Stakeholders and Interactions

During the second iteration, no major updates are identified in TheFSM Stakeholders list. Nevertheless, the Business view and the Data Market view of the stakeholders were further

elaborated and mapped to the relevant technical solutions of TheFSM Project, i.e., TheFSM Applications (WP4) and the TheFSM Platform (WP3).

As mentioned in D3.1 v1.0 (M9) in the **Business View**, the stakeholders are treated as parts of fundamental procedures in TheFSM value chain. The two main aspects in this classification are the **supply-chain procedures** (simply put, the entire process from food producing, processing, distribution etc. till it reaches the customer) and the **certification processes** (all steps required in order for a stakeholder to receive a certificate). Both aspects involve various stakeholders, however not all of them are equally vital to the procedures. This is addressed by further dividing the stakeholders participating in each aspect into primary and secondary (isPrimary column). Sub-categories of stakeholders are included here as well, for the sake of completeness.

Role	IsPrimary	Sub-roles	Description
Producer	Yes	Growers, Farmers, Crop traders, Winegrowers, Winemakers	Creates the product in raw form
Food processor	Yes	Bottlers, Food processing companies	Processes food to make it ready for consumption (e.g., bottler)
Supplier	Yes	-	Suppliers are at the top of the chain and give them products to distributors
Distributor	Yes	-	Distributors take products from suppliers and sell them to wholesalers and retailers
Retailer	Yes	-	Retailers get their products from wholesalers or from distributors
Consultant	No	-	Consultants offer advice to stakeholders for decision-making, during both supply chain and certification processes

Table 1: TheFSM stakeholders involved in supply-chain procedures

Role	IsPrimary	Description
Certification Body	Yes	Issues certifications and oversees supply chain for transparency, conducts audits and issues certificates
Certification Scheme owners	Yes	Issues certifications and oversees supply chain for transparency
Public Authorities	Yes	They conduct inspections and they monitor the risk of the supply chain.
Inspector/auditor	Yes	Inspects and reports on findings via periodic checks of stakeholders upholding to standards
Lab expert	Yes	Conducts lab tests to issue certifications (with the permission of the certification body)
Consultant	No	Consultants offer advice to stakeholders for decision-making, during both supply chain and certification processes

Retailer	No	Retailers get their products from wholesalers or from distributors and produce their Private Label (PL) products from manufacturers
-----------------	----	---

Table 2: TheFSM stakeholders involved in certification procedures

Finally, the second stakeholder addresses the **Data Market perspective**. In this view, we define the aspects each stakeholder should cover when participating in **data exchange**. Our study shows that all stakeholders are meant to have both **data producer** and **data consumer** stakeholders when exchanging data. This is derived from workflow coverage of all Use Cases which were defined throughout the above agile methodology. It also means that extra care must be taken when implementing the functionalities of the platform as the interaction graph is the most dense it can be. Additionally, the platform envisages the potential use of the exposed data services from **food tech companies** and other **ICT companies** which are interested to make use of the data of TheFSM platform and build on top of TheFSM data services additional added value services, which they could also be added into TheFSM platform. Thus, the following table of technological stakeholders is foreseen.

Role	IsPrimary	Description
Data consumer	Yes	All stakeholders mentioned above (tables 10 and 11) are data consumers in TheFSM Data platform through the use of the relevant TheFSM Applications
Data provider	Yes	All stakeholders mentioned above (tables 10 and 11) are data providers in TheFSM Data platform the use of the relevant TheFSM Applications
External data and service providers and consumers	Yes	Developers and other ICT experts from the food tech domain and certification which to make use of the data of TheFSM platform and build on top of TheFSM data services additional added value services

Table 3: TheFSM Data market stakeholders

TheFSM Stakeholders mapping to the relevant technical solutions of TheFSM project is provided below:

Business procedure	Role	TheFSM technical solution	Comments
TheFSM stakeholders involved in supply-chain procedures	Producer	FOODAKAI 2.0, Agrivi 2.0 other external application	Main end users of the applications.
	Food processor	FOODAKAI 2.0, Agrivi 2.0 other external application	
	Supplier	FOODAKAI 2.0, Agrivi 2.0 other external application	
	Distributor	FOODAKAI 2.0, Agrivi 2.0 other external application	
	Retailer	FOODAKAI 2.0, Agrivi 2.0 other external application	
	Consultant	FOODAKAI 2.0, Agrivi 2.0 other external application	

TheFSM stakeholders involved in certification procedures	Certification Body	Food Inspector 2.0, Agrivi 2.0, other external application	Main end users of the applications.
	Certification Scheme owners	Food Inspector 2.0, Agrivi 2.0, other external application	
	Public Authorities	Food Inspector 2.0, Agrivi 2.0, other external application	
	Inspector/auditor	Food Inspector 2.0, Agrivi 2.0, other external application	
	Lab expert	Food Inspector 2.0, Agrivi 2.0, other external application	
	Consultant	Food Inspector 2.0, Agrivi 2.0, other external application	
	Retailer	FOODAKAI 2.0, Agrivi 2.0 other external application	
TheFSM Data market stakeholders	Data consumer	TheFSM Data Market, TheFSM Platform	<ul style="list-style-type: none"> - TheFSM applications (FOODAKAI 2.0, Food Inspector 2.0, Agrivi 2.0, other third party) integrate to TheFSM Platform to consume data - Data analysts, ICT experts in the food safety domain access and purchase data assets
	Data provider	TheFSM Data Market, TheFSM Platform	<ul style="list-style-type: none"> - TheFSM applications (FOODAKAI 2.0, Food Inspector 2.0, Agrivi 2.0, other third party) integrate to TheFSM Platform to provide and trade data - Data analysts, ICT experts in the food safety domain provide and trade data assets
	Service Consumers	TheFSM Data Market, TheFSM Platform	<ul style="list-style-type: none"> - TheFSM applications (FOODAKAI 2.0, Food Inspector 2.0, Agrivi 2.0, other third party) integrate to TheFSM

			<p>Platform to provide services</p> <ul style="list-style-type: none"> - ICT experts in the food safety domain access and trade third party services through TheFSM Platform
	Service Providers	TheFSM Data Market, TheFSM Platform	<ul style="list-style-type: none"> - TheFSM applications (FOODAKAI 2.0, Food Inspector 2.0, Agrivi 2.0, other third party) integrate to TheFSM Platform to provide services - ICT experts in the food safety domain share and trade their services through TheFSM Platform - Third party systems (e.g. ERPs, Supplier Management Systems, Laboratory Management Systems) providers will be able to use TheFSM platform to securely exchange critical information for the certification process

Table 4: TheFSM Stakeholders and the relevant TheFSM technical solution

3 THEFSM PLATFORM ARCHITECTURE

3.1. Conceptual Architecture

The conceptual architecture of TheFSM has been designed by conducting a thorough analysis of the updated technical requirements that were later translated into technological, beyond the state of the art, software modules that are implemented in the context of WP2, WP3 and WP4. Additionally, the technical evaluation of TheFSM Platform v1.0 also provided technical updates in order to further improve the adopted architecture and address the end user needs. The main challenge of the TheFSM architecture v2.0 is to extend the scalable and flexible environment provided by TheFSM architecture v1.0 and further enhance the data security, processing, monetization and analytics functionalities. Whereas the TheFSM architecture v1.0 provided a more detailed view of the security and data processing backbone of the TheFSM Platform, the second version of the architecture revisits the already defined approaches in the aforementioned functionalities, while draws more light to the **data monetization** and **trading services**, as well as, the **data analytics provision**.

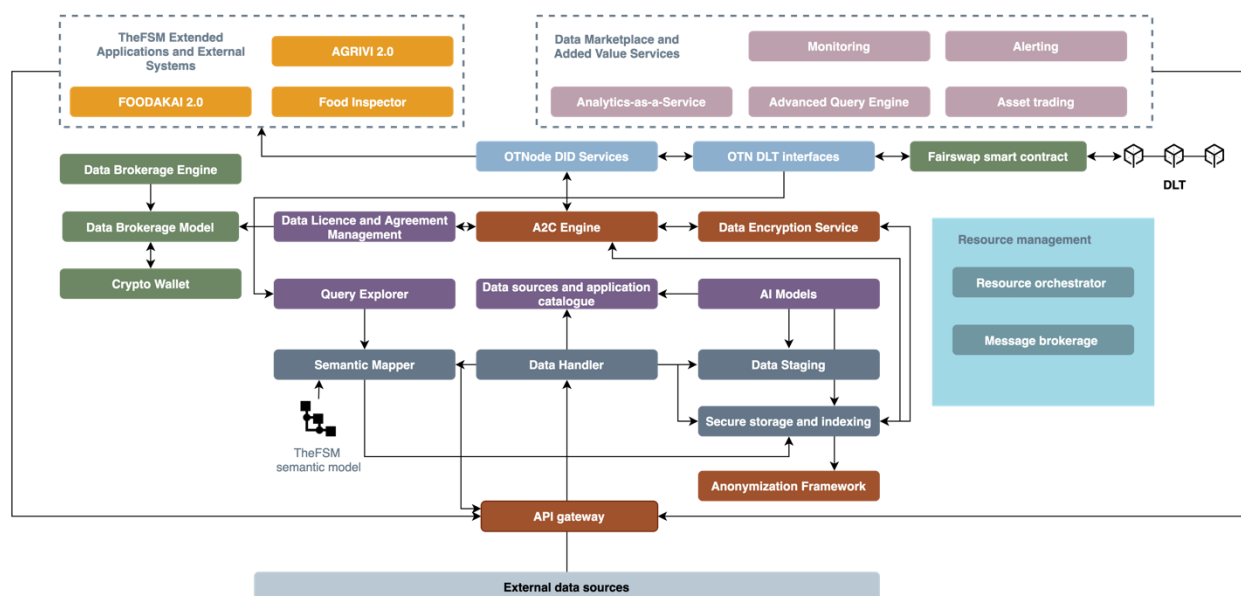


Figure 3: TheFSM Reference Architecture

TheFSM reference architecture consists of a set of loosely coupled architectural components which are organized in three logical architectural layers: **Data curation and enrichment, core services and backend data Platform** and the **applications and marketplace** layer. The data curation and enrichment layer includes all the components which participate mainly in data ingestion, preparation, semantic enrichment and maintenance processes (and are mainly developed under the implementation tasks of WP2). The core services and backend data platform are the components which make use of the data stored and exchanged into TheFSM platform and perform the main data processing, encryption-decryption, analysis, identity and monetization services. Finally, the applications and marketplace layer includes the final offered services of TheFSM platform as they are implemented and provided by the lower architectural layers. These

components include the three main applications which will support the end users (developed in WP4), as well as, the API of the services provided by TheFSM data platform consisting of TheFSM data market (developed in WP2 and WP3). These services are provided to external applications for integration and other data consumers/providers for data asset trading.

Supporting the everyday transactions as well as the data asset trading in food safety and certification requires the harmonization of multidisciplinary data deriving from a number of heterogeneous data sources (see section 5.2.1), while at the same time there's the need to enable the interoperability among different systems and support real-time data exchange. Towards this, the **Data Handler** ingests the data and performs data ingestion functionalities for collecting and storing (aggregated) data from various data streams. Data Handler performs ETL processes and implements a first lever of data transformation regarding a set of supported standards (EPCIS 2.0, GS1 WebVoc, etc...). The collected data will be summarized, preprocessed and stored in the **Data Staging**. Data staging is a collection of data storage systems (such as GraphDB, Mongo) for storing user data and metadata for sharing which is provided on batches or collected and ingested by the Data Handler and are ready for semantic enrichment by the Semantic Mapper. The **Semantic Mapper** provides ETL and semantic enrichment of the data using **TheFSM Semantic Model** and generates the relevant RDF representation of the data which is stored in the **Secure storage and Indexing**. The Secure storage and indexing consists of (big-data enabled) storage solutions which supplements the Data Staging infrastructure and are also capable of storing and managing large amount of data in structured or **unstructured** format. Indexing tools like Elasticsearch are used for better text search performance. TheFSM storage solution provides a set of key characteristics such as horizontal scalability, high availability, high performance and advanced security. Additionally, it provides the indexing capabilities of the platform over multiple complex datasets with flexibility and efficiency.

The enriched data are accessed from the **Query Explorer** which implements complex semantic queries based on a set of parameters in order to access and retrieve information from the Secure storage and Indexing. All data sources, as well as, data services built on top of the data are published to **Data Sources and Application Catalogue** which implements a repository of the TheFSM data applications created in the platform. As such, TheFSM applications can be stored, retrieved, modified and loaded in TheFSM Data Marketplace any time they are required. The purpose of the catalogue is to enable the reuse of the designed datasets and applications through a defined license provided by the **Data License and Agreement Manager**. The component is responsible for handling all processes related to the data licenses and IPR attributes, as well as enabling the drafting, signing, and enforcing the smart data contracts that correspond to data sharing agreements between platform users. The component defines the **Data Brokerage Model** which is used by the **Data Brokerage Engine** which is responsible for generic brokering of datasets in TheFSM platform between different parties and for possible financial compensation. The **Data Brokerage Engine and Model** are closely tied with interactions with the **Crypto Wallet**, which is responsible for transactions between **buyers** and **sellers**. Regarding the authorization of the platform, attribute-based access control on the data is implemented by the **A2C Engine** based on the access policies defined by a combination of policies from data providers in the **Data License and Agreement Manager**, administrators, moderators, API providers (via the **API**

Gateway) and so on. Regarding data encryption at transit for the files stored in the **Secure storage and Indexing**, we have replaced the **ABE Engine** with Hybrid Encryption (renamed to **Data Encryption Service** in the figure), for reasons which are documented in Section 3.3.6. Both engines ensure that only authorized users with specific attributes which fulfil the defined access policies can a) access and b) decrypt the data that they want to access. The **Anonymization Framework** complements the secure attribute-based handling of the data by providing anonymization and pseudonymization of the data. The data access and brokerage mechanisms are supported by state-of-the-art decentralized identity management provided by the **OTNode DID Services**. This component ensures provision and resolution of the Decentralized Identifier Descriptor (DID) and the relevant Verifiable Credentials (VC) of each organization that wants to perform any action on the data (provision, request, update) using DLT. The **OTNode DLT Interfaces** offers an abstraction and data management layer over DLT and facilitates the communication among the OTNode DID Services, the A2C Engine, and the Secure Storage and Indexing in order to manage traceability data exchanges through the platform, as well as, transparency and immutability of the data transactions. **TheFSM Data Marketplace and Added Value Services** provide through APIs a set of added value services to empower the food safety and certification industry which address: a) Provision of data to stakeholders through intelligent query engine, b) Data sharing and monetization services, c) support the analytics algorithms workflow design and execution. These services are used by **TheFSM Extended Applications** which implement a set of intuitive tools and UIs for the stakeholders of TheFSM value chain.

A brand new addition which plays a central role in the updated architecture is the **API Gateway**. This new component was created due to the needs of the platform to allow data providers to provide their data through the platform via APIs. The **API Gateway** is responsible for providing this functionality, working in unison with the **A2C Engine** to protect access to all APIs. The **API Gateway** is thoroughly discussed in this deliverable, both in the next section where some workflows involve it, and throughout its dedicated section later on (section 3.3.6).

Regarding the resource management of the underlying technical infrastructure of the platform, the resource orchestrator is responsible for distributing tasks, load balancing, creation and setup of VM's for independent, isolated tasks etc. by utilizing state-of-the-art solutions like Kubernetes as part of its infrastructure. The message broker will be responsible for forwarding notifications to required users, depending on settings and nature of data updates. Existing solutions such as Apache Kafka are under consideration for the core of this functionality. Last, the dataflow management is responsible for integrating data, converting them in different formats, storing etc.

3.2. TheFSM Platform Main Workflows

In this section, we are going to present the workflows addressing the TheFSM Data Governance Model. These workflows constitute the main workflows taking place in the TheFSM's infrastructure which guide all the different workflows throughout TheFSM. More specifically, they are as follows:

- Request authentication and authorization
- Dataset upload

- Query search
- Dataset Monetization
- API Gateway (both request execution and new endpoint registration)

When inspecting the workflows from the point of view of the Data Governance model, each workflow's coverage of the model is provided in the table below:

Workflow	Data Collection	Data Ownership	Data Curation	Data Security	Data Monetization
Request authentication and authorization				ABAC policies	
Dataset upload	data asset upload	IPR management, data licencing	semantic mapping, standardization, data asset indexing	data asset encryption, ABAC policies	data trading (when uploaded dataset is monetized and ready to be purchased)
Query search			semantic mapping, standardization	ABAC policies	
Dataset Monetization		data licencing		data asset encryption, ABAC policies	wallet management
API Gateway (request execution and new endpoint registration)	Data upload through API	IPR management, data licencing		data asset encryption, ABAC policies, Data service API registration and secure access	

Table 5: Activity diagrams workflows mapping to data governance model

3.2.1. Authentication and Authorization

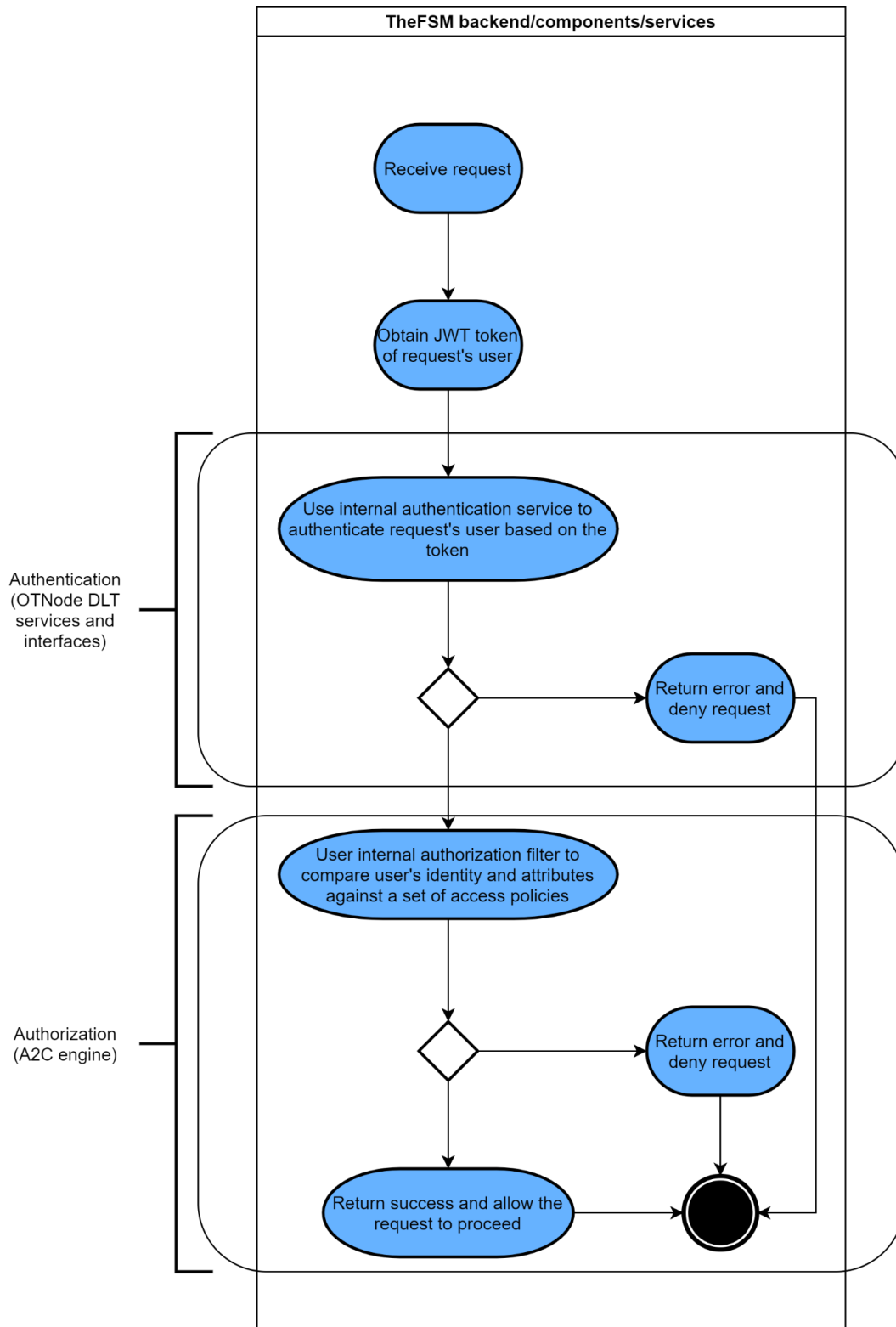
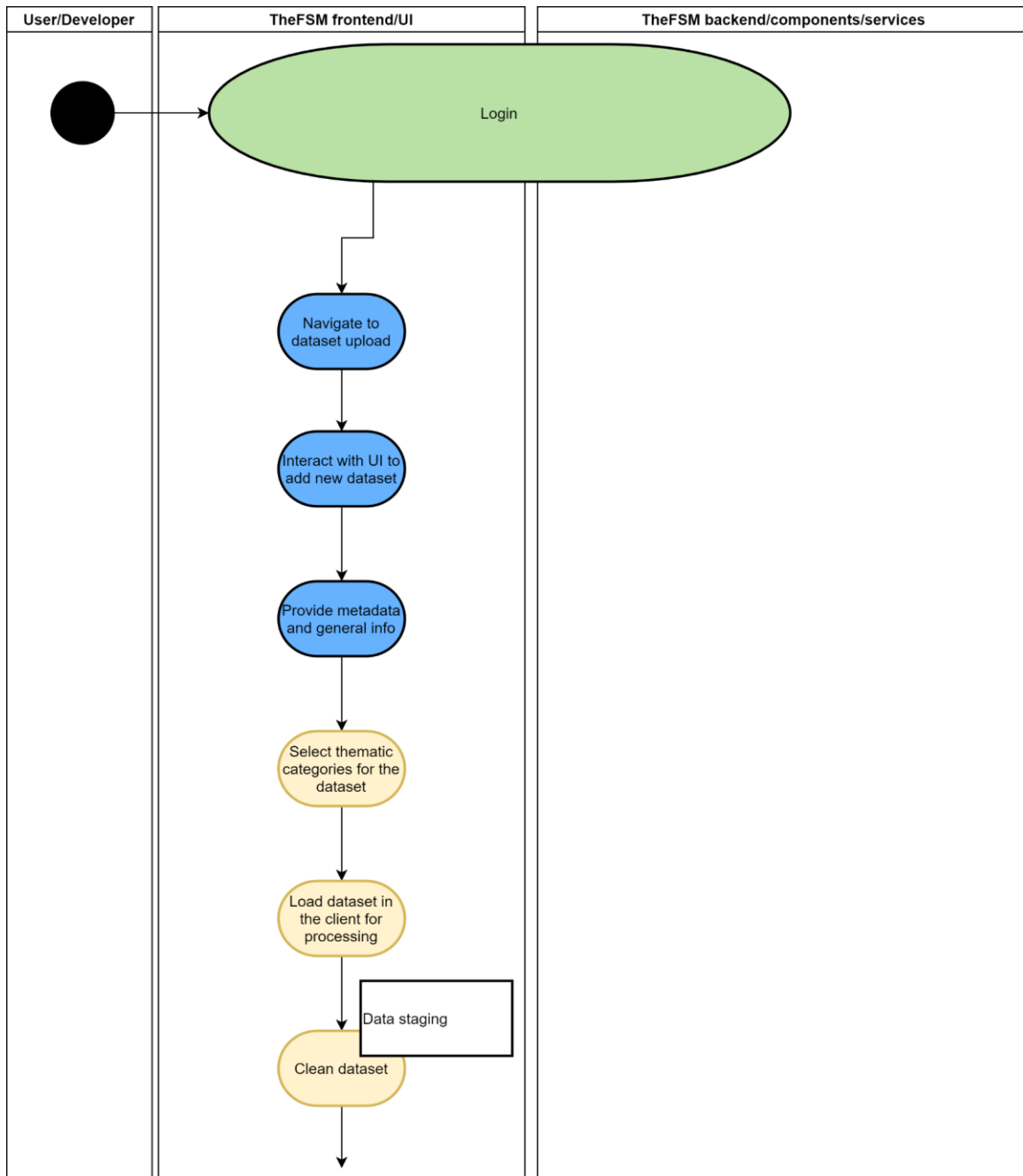
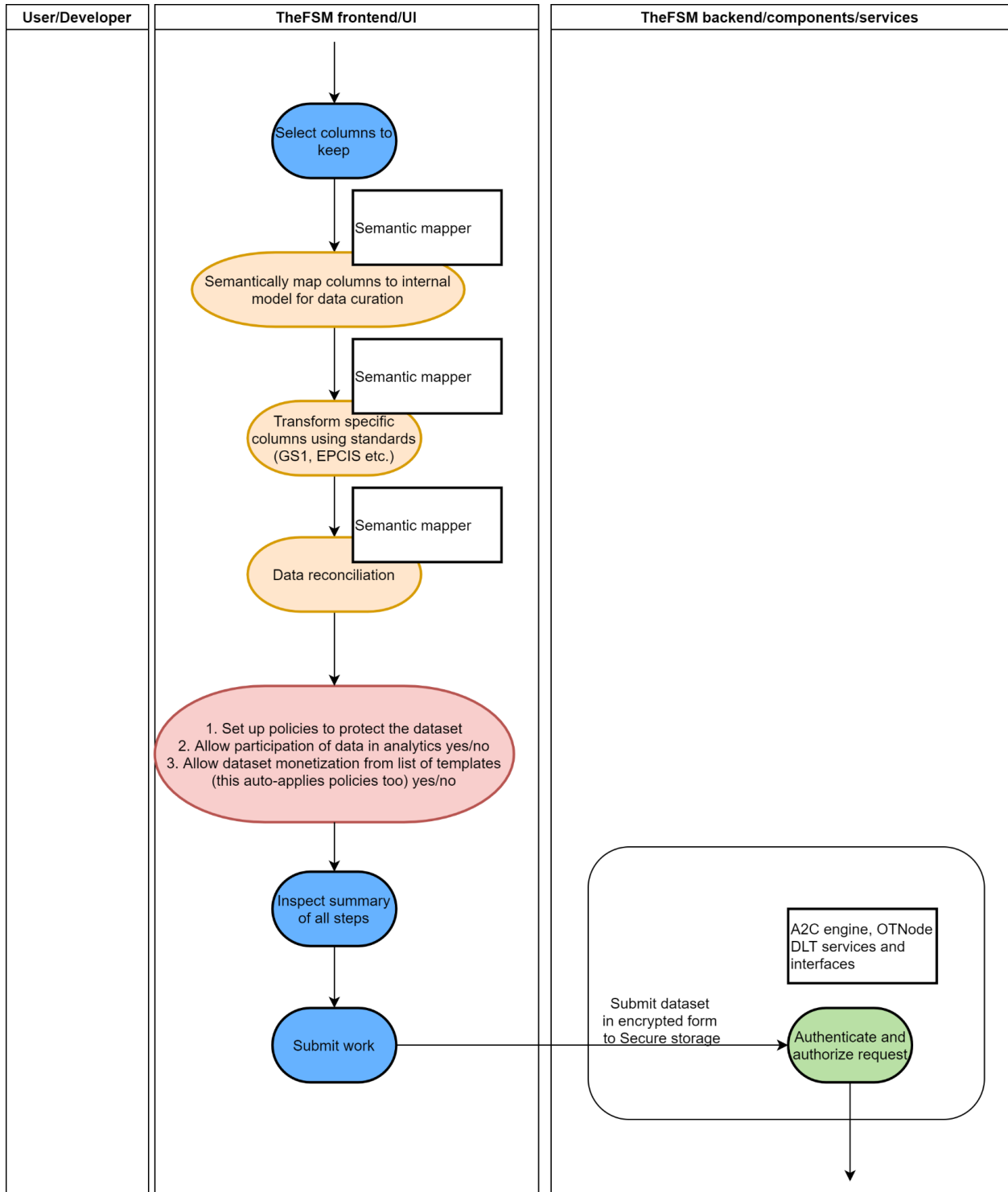


Figure 4: Authentication and Authorization

3.2.2. Dataset Upload





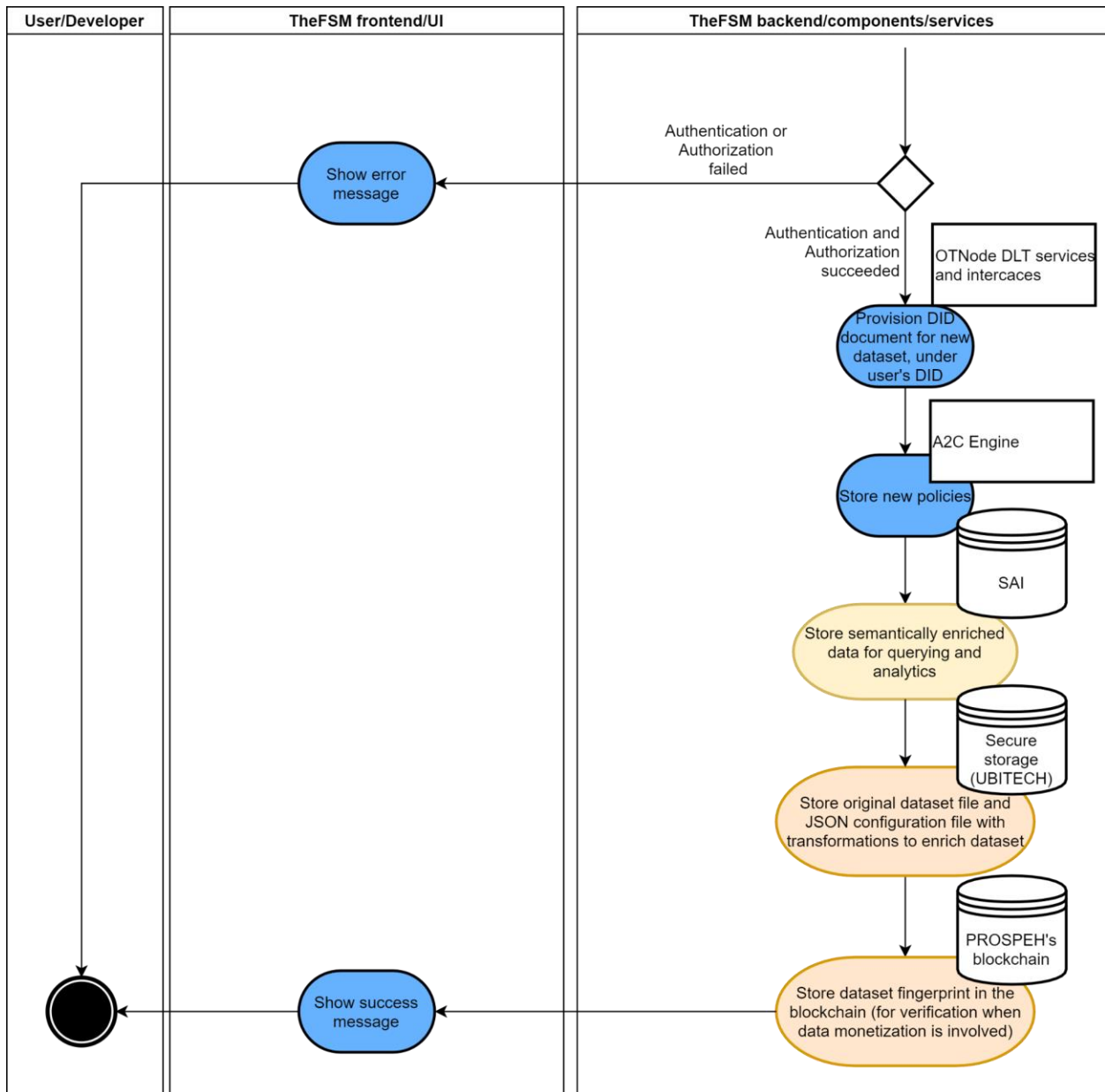


Figure 5: Dataset upload

3.2.3. Query search

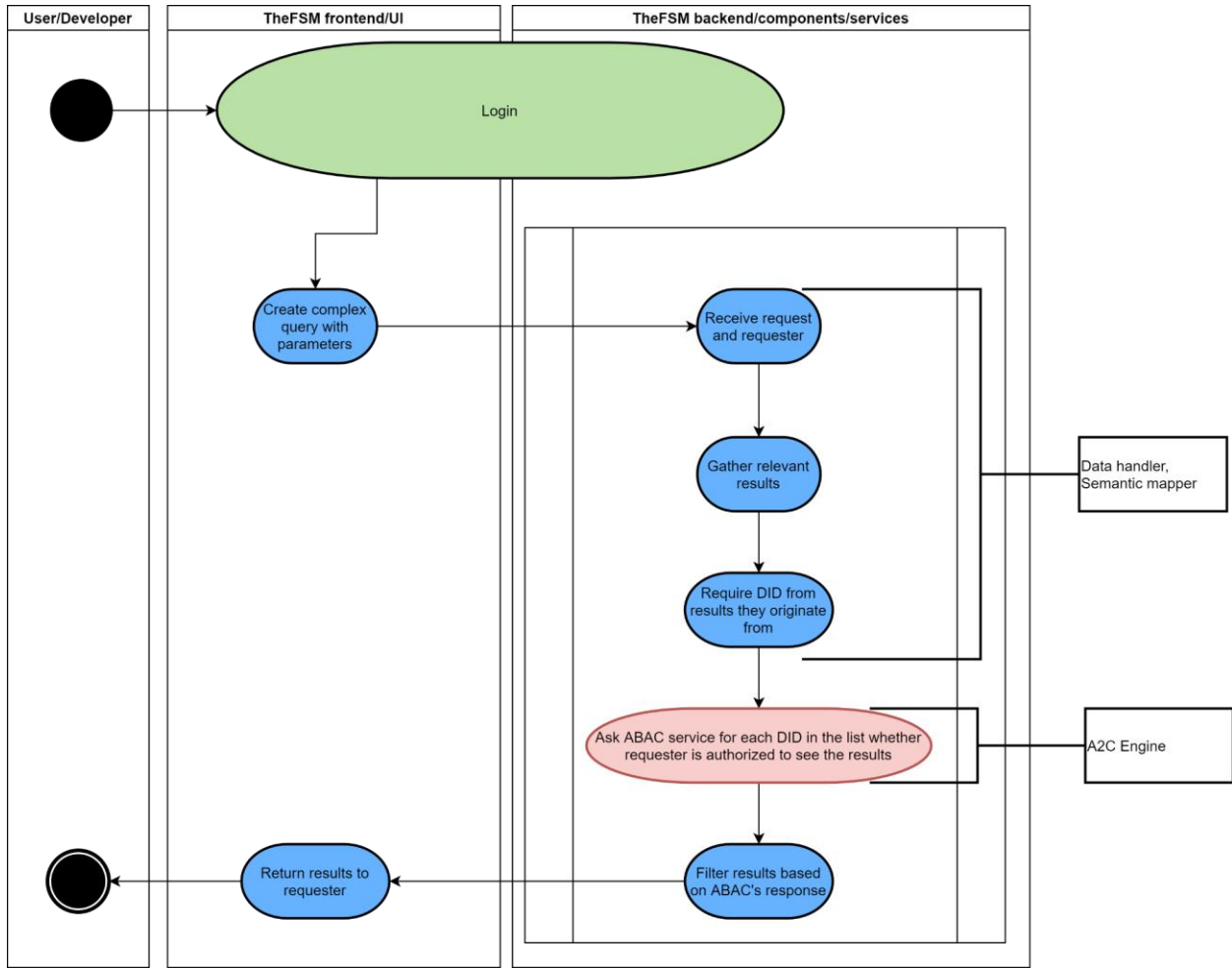


Figure 6: Query search

3.2.4. Monetization

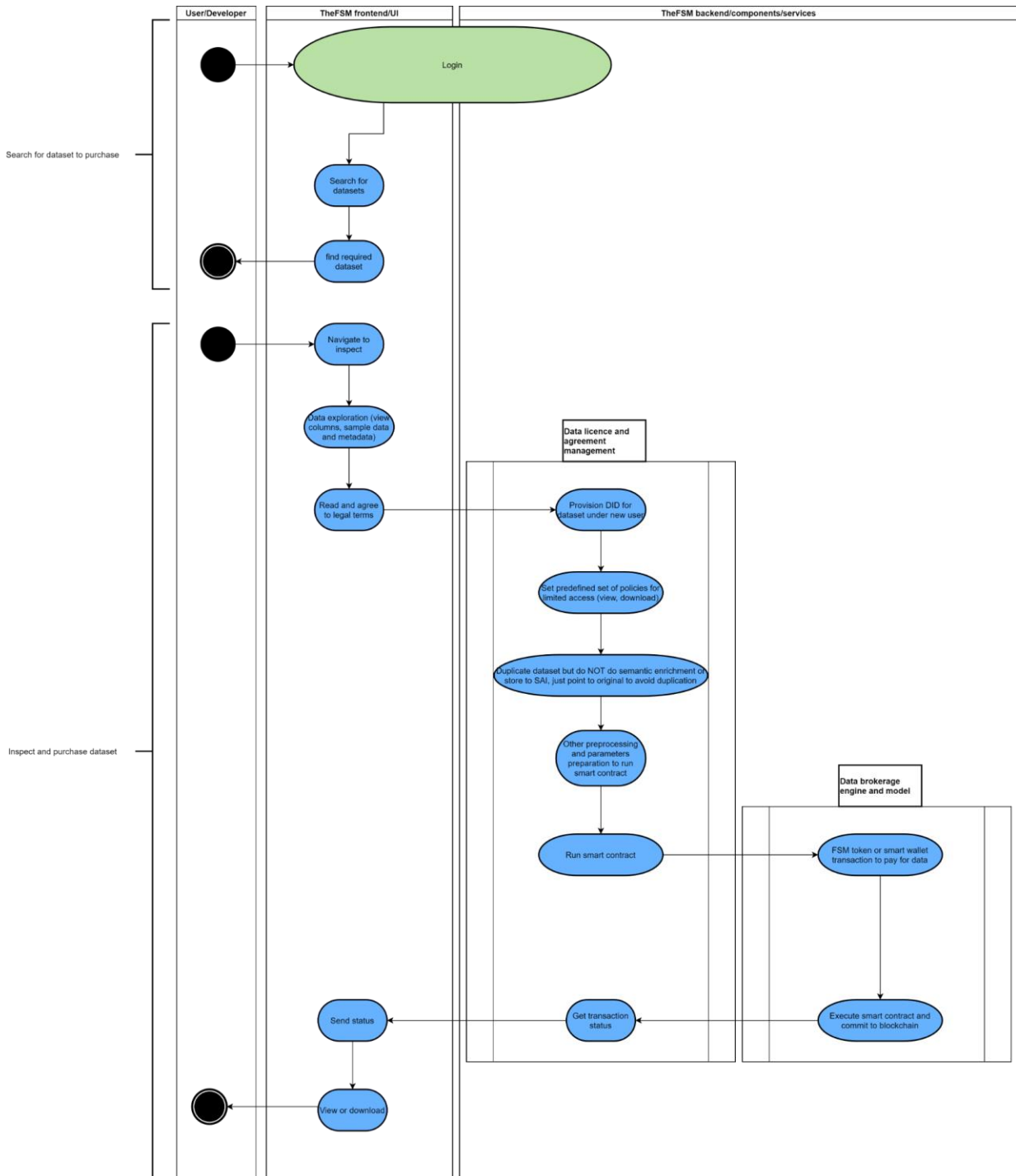


Figure 7: Monetized dataset purchase.

3.2.5. API Gateway

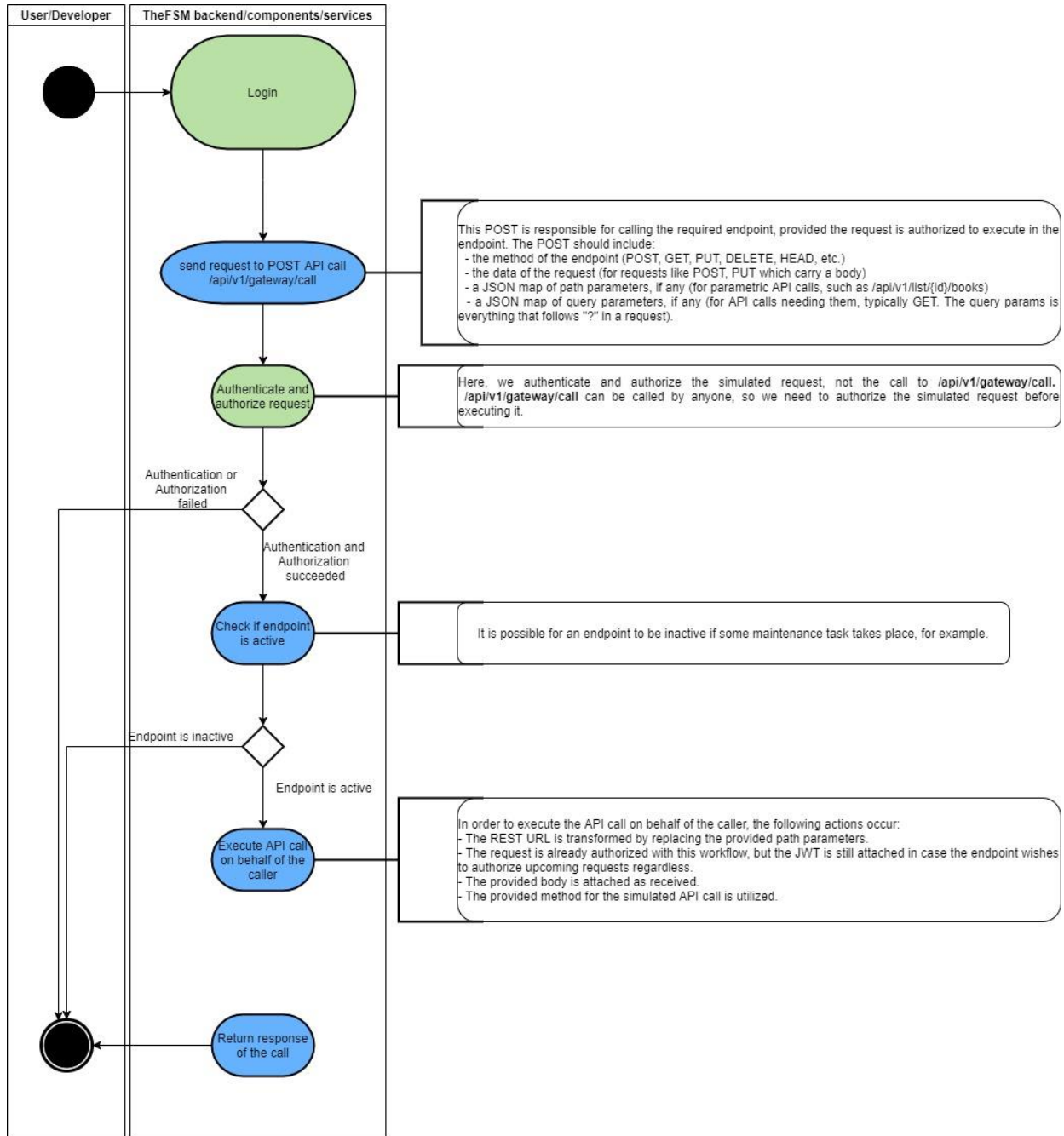


Figure 8: API Gateway request execution activity diagram.

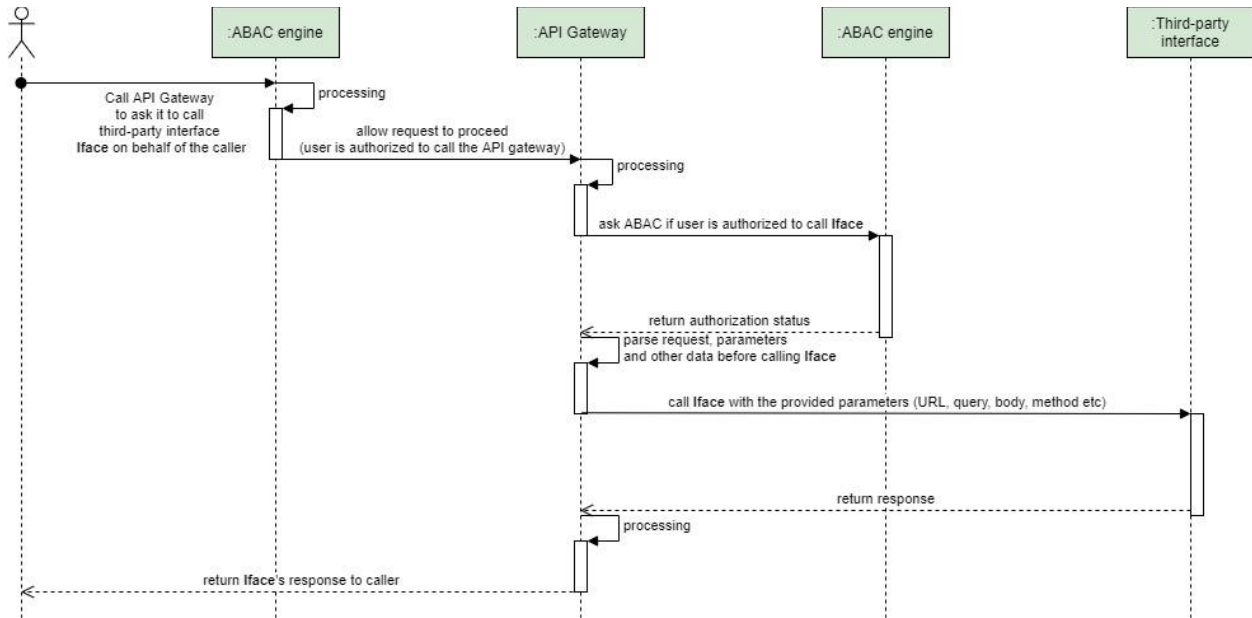
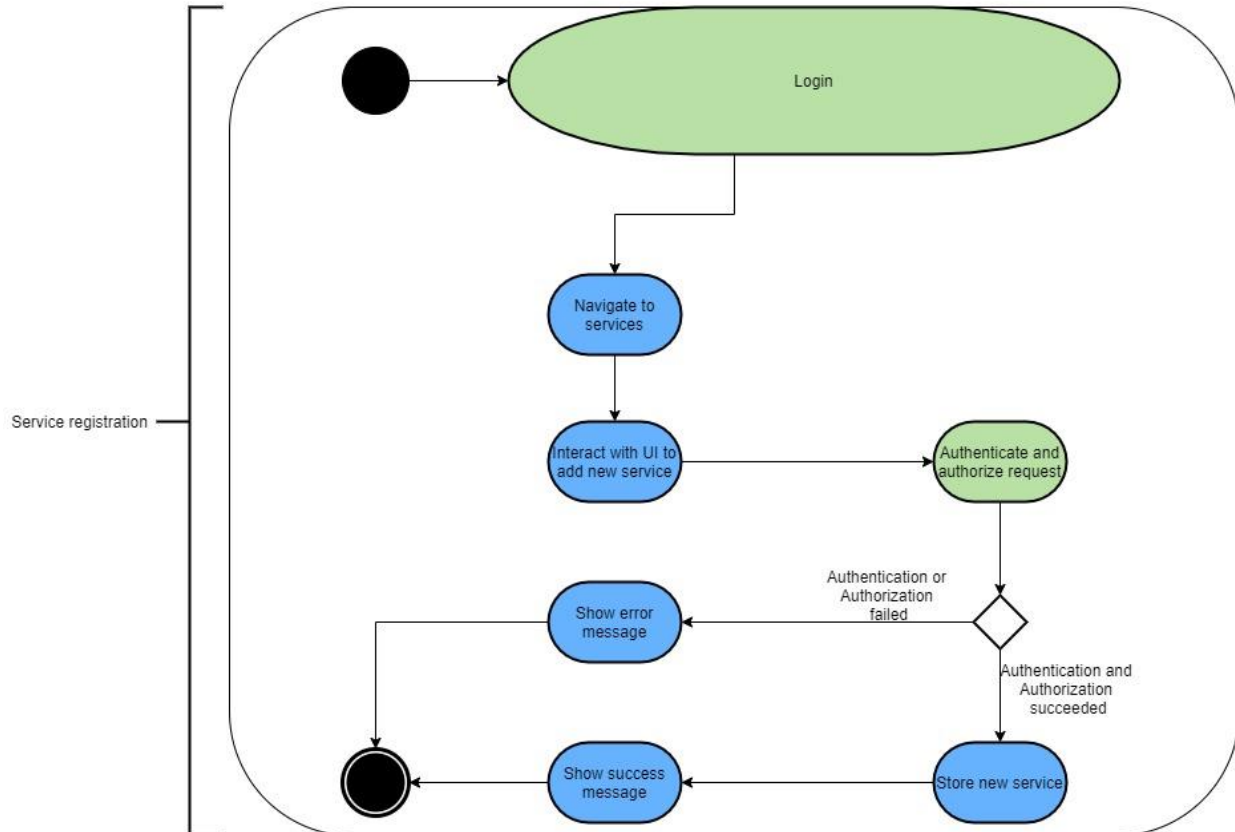


Figure 9: API Gateway request execution sequence diagram.

Note 1:
We assume the person/developer/administrator member executing the following scenario is already registered to the system and potentially has the proper attributes to execute it. It is also possible to execute the scenario by communicating directly with the backend of TheFSM without the UI.

Note 2:
A service must be registered before its endpoints.

Terminology:
Service = A conceptual set of REST API calls provided by a third party and integrated in TheFSM.
Endpoint = A REST API URL (which can also have path parameters, e.g., /api/v1/list/{id}/users).



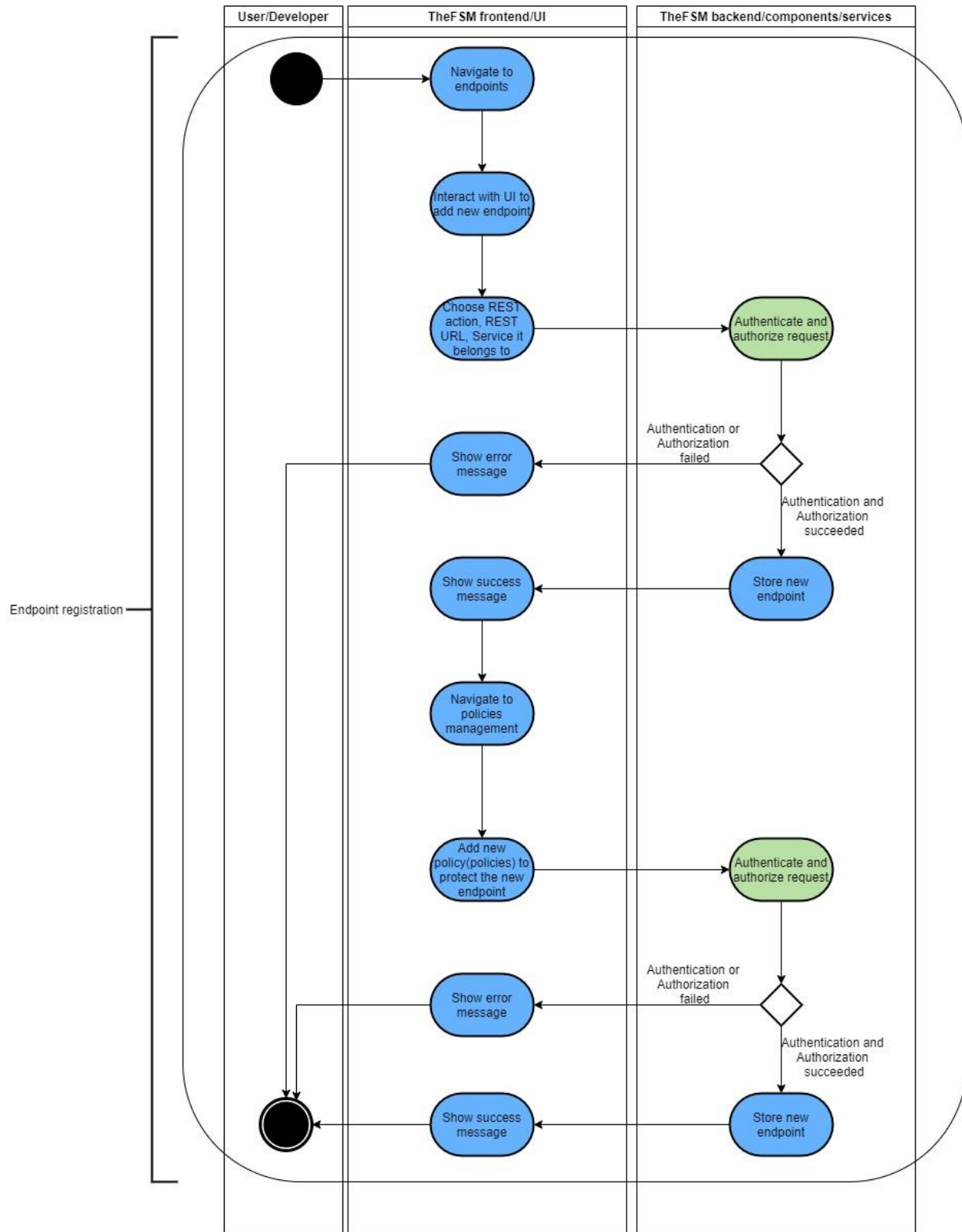


Figure 10: API Gateway new endpoint registration activity diagram.

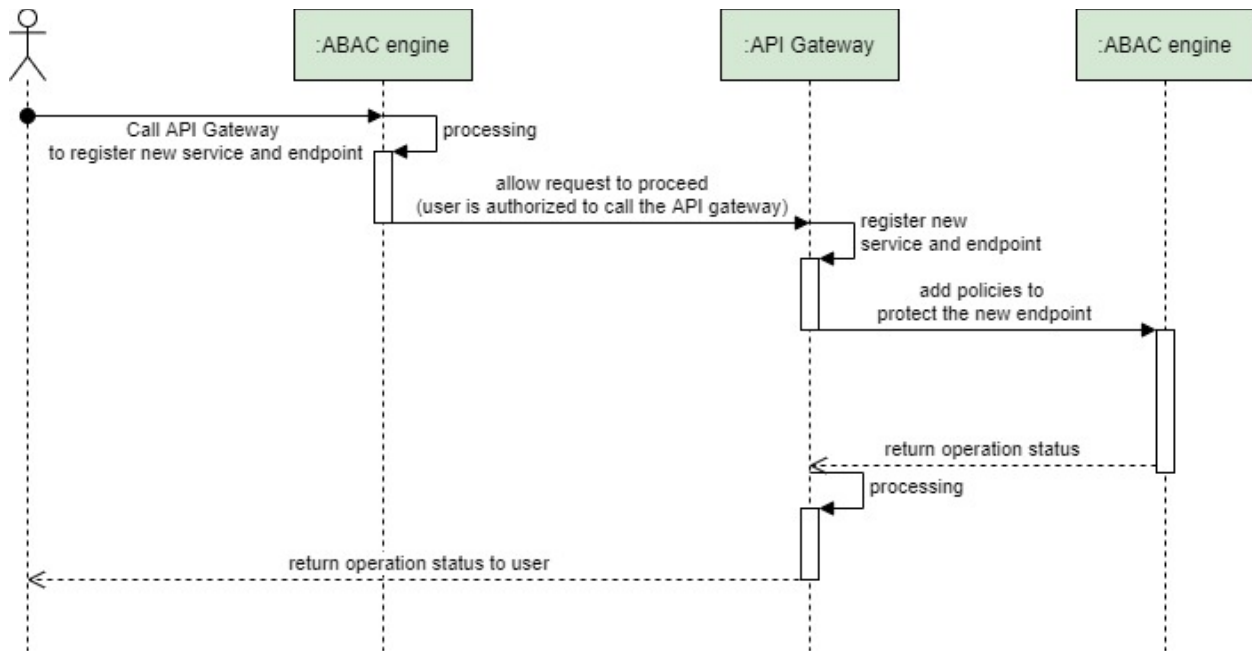


Figure 11: API Gateway new endpoint registration sequence diagram.

3.3. TheFSM Architectural Components

3.3.1. Data Sources

Supporting the everyday transactions as well as the data asset trading in food safety and certification requires the harmonization of multidisciplinary data deriving from a number of heterogeneous data sources. Each of them has its specifics and sometimes access restrictions and requires individual effort before getting accessible by the platform. They are intended to be used on demand, in real time, when necessary for execution of customer queries and taking into account the access privileges (if required) of the initiating user.

Most data sources are expected to be accessed as APIs providing REST interface over http protocol or as SparQL/GraphQL entry points. A detailed list of external data sources will be presented in D2.1 (M12). Nevertheless, below a list with the core data sources which affect the design of the architecture is presented. TheFSM platform is provisioned as a framework where the data exchange and semantic enrichment is provided as a common functionality even though adding each new specific data source will require its definition and likely development of a specific microservice to deal with it. However, the proposed approach facilitates dynamic addition of new data sources.

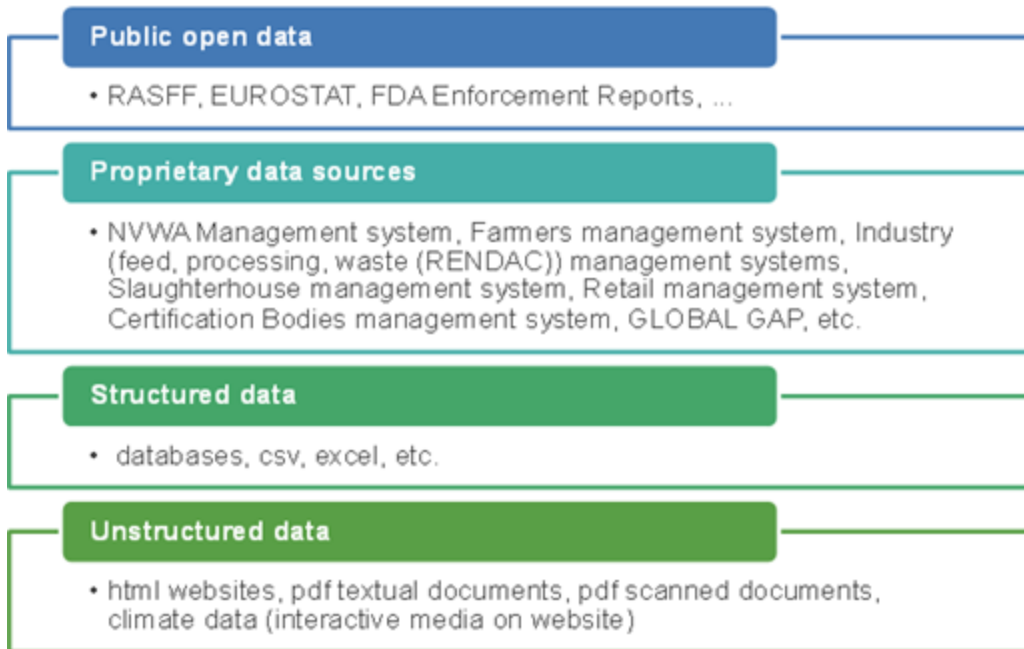


Figure 12: TheFSM Platform Data sources

The following data types are identified to be the input to the platform software:

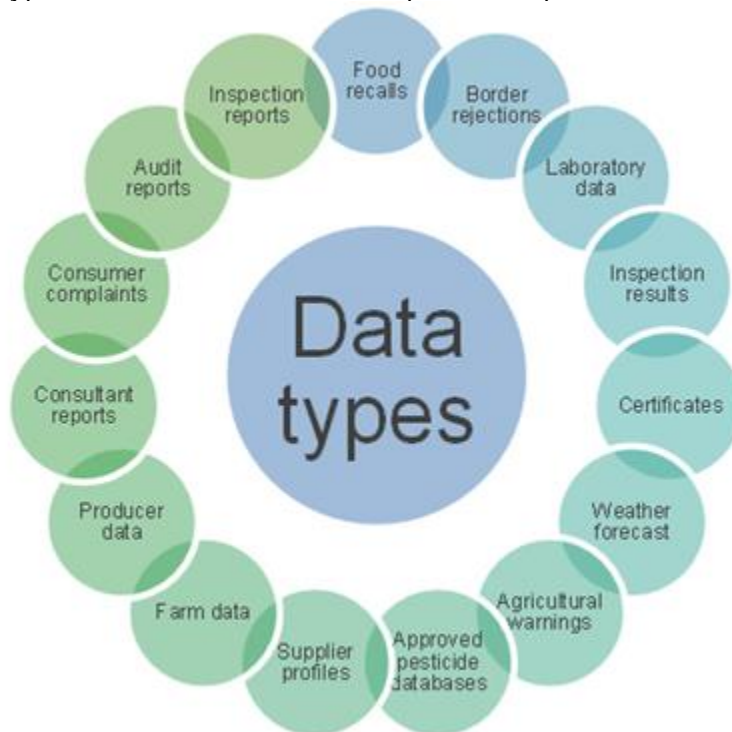


Figure 13: T Data types supported by TheFSM

3.3.2. Data Curation and Semantic Enrichment

This architectural layer provides data ingestion, preparation, semantic enrichment and maintenance processes (and are mainly developed under the implementation tasks of WP2). The layer mainly performs semantic transformation of input data and extracted metadata from the input documents, solving the problem of ambiguity. It also provides semantic enrichment by linking data to various ontologies and external data sources. The layer makes use and extends on the Ontotext Platform¹.

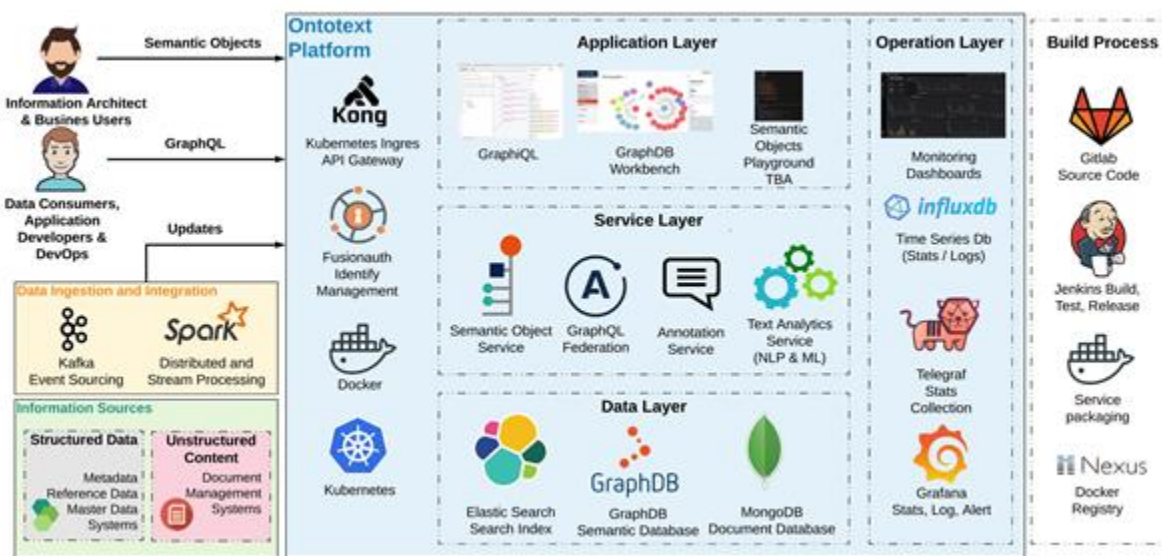


Figure 14: High-level architecture of Data Curation and Enrichment

TheFSM Semantic Model

TheFSM Semantic Model is a set of ontologies and class interrelations featuring the semantic representations of the incoming data objects, received from Data Processing and Analyzing component modules and the other data sources. There is also going to be a set of definitions of the specific input data fields mapping into semantic categories in RDF format as well as the ways of retrieving the corresponding objects' data from their data sources - as API calls, SparQL/GraphQL endpoints or stored data.

Design and Functionalities Overview

TheFSM Semantic Model will be defined using Semantic Object Model Language (SOML) [4] and functioning via the Semantic Objects Service part of the Ontotext Platform, where the data objects used in Use Case Scenarios are represented semantically. TheFSM Semantic Model will provide an ontology which will be defined on top of some common ontologies used in LOD data sources in regards to data needs from business requirements and use cases workflows. It will also include definitions of the other data classes, retrieved from the Data sources and their semantic mappings

¹ <http://platform.ontotext.com/index.html>

into the terms of the ontology. The data fields (objects) from different data sources which will be used for enrichment in the use cases will be identified and the ways of their retrieval to be included in Data Sources and application catalogue.

Semantic Mapper

The Semantic Mapper is a service providing functionality for mapping input data to semantic categories and resulting in their RDF representation.

Design and Functionalities Overview

This service is based on the Ontotext platform² components **OntoRefine tool** and **Apollo Federation** service providing functionality for data mapping from textual data and relational database records to semantic representation in RDF format. To resolve any ambiguity problems in the input data, the Onto Refine service will be used to solve them based on the values of the other properties (fields) and the specific context.

The Federation service will be used to combine the data from the various data sources, retrieved by using the functionality of the Data Handler subcomponent and in regards with the mapping definitions in TheFSM Semantic Model and data retrieval descriptions in Data Sources and application catalogue.

The service is going to receive queries from the Query Explorer and using the mapping definitions in the Semantic Data Model to transform the provided data (query parameters) and extract the needed data from data sources via Data Handler functionality, map them again to semantic categories and via Federation Service to construct the query responses.

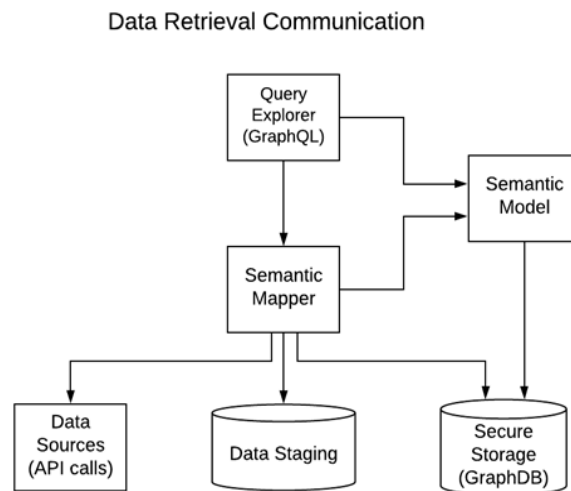


Figure 15: High level architecture of semantic mapper

² <http://platform.ontotext.com/index.html>

Data Handler

The Data Handler is a component providing data ingestion functionalities for collecting and storing (aggregated) data from various data streams. The collected data will be summarized, preprocessed and stored in the Data Staging.

Design and Functionalities Overview

The Data Handler is a set of streaming entry points for receiving WoT and transactional data from third party systems.

It will include WoT parsers, EPCIS parsers and other ETL pipelines for incorporating data from streams from third parties. ETL pipeline is a software service performing data transformation from one format to another. It is very case specific and is usually implemented case by case for transforming parser output into RDF and storing, if necessary. They will be implemented and deployed as microservices.

If a data provider will share static data which to be stored in the system, a corresponding ETL pipeline must be developed for data transformations and storing in the Data Stage component.

Each data source type and even the specific data fields/properties/classes which are to be retrieved, as well as the corresponding ETL pipelines to be used for intermediate processing will be defined in the **Data Sources and Application Catalogue**. When a new data source is added, the corresponding definitions of the data it provides must be added there. If the new source requires development of ETL pipelines, they must be implemented and the corresponding records must also be added to Data Sources and application catalogue.

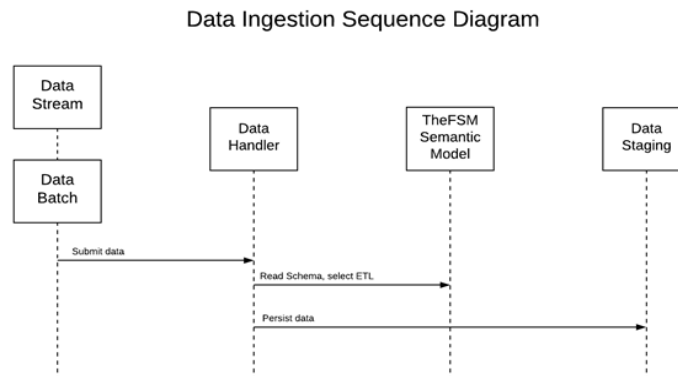


Figure 16: Data ingestion sequence diagram

Data Staging

Data Staging is a collection of data storage systems for storing user data for sharing which is provided on batches or collected and ingested by the Data Handler functionality.

Design and Functionalities Overview

In terms of TheFSM architecture, the Data Staging component consists of data management systems in regards to the stored data types. It will include Ontotext Platform and maybe some other types of storages like JSON storage (MongoDB), and RDBMS (Postgres).

All the data classes in the Data Staging must be included in TheFSM Semantic model SOML schema. The corresponding extensions and reading microservices must be included in the Data Sources and Applications Catalogue. The access restrictions if any must be defined as described further in Data Licensing and Agreement Service.

Secure Storage and Indexing

This component contains the semantic repository of the project where all necessary knowledge for running the platform is persisted. Here is the project specific ontology and downloaded and ingested external static ontologies and dictionaries.

Design and Functionalities Overview

The Ontotext platform featuring GraphDB is expected to be used as a semantic data repository in combination with Elasticsearch for better text search performance. MongoDB could be used as a separate json-data storage, if necessary and some RDBMS like PostgreSQL for local, offline storing of relational data, e.g. if some end user likes to share his static dataset.

All data storage functionality is going to be compliant with GDPR requirements and defined user-object access privileges from Data Licensing and Agreement and Access Management Service. This implies that no sensitive data is stored in the database and most of the data points are retrieved from the data sources when necessary for a specific query from a specific user taking into account his access rights. In cases where some sensitive data is stored, it must be encrypted.

The emphasis, however is on dynamic retrieval of up-to-date information from its native sources, so that dataset copies are not going to be stored in the platform only as an exception. However, ontologies and other metadata, as well as data sources descriptions will be going to be persisted in the system.

The above also means that the functionality of the platform is going to depend on third party API versions and any change in the API and/or available data formats could result in malfunction of the data retrieval from the corresponding source, at least until the necessary changes in DSD and the corresponding ETL are done. This also means that some measures of early detection of API changes should be taken and that the partners must share information about planned changes of API call formats and/or data accessibility in advance in order the corresponding updates to be performed on time without interruption of the functionality.

3.3.3. Data Processing

Data Sources and application catalogue

One of the main objectives of the data semantic processing in TheFSM project platform, is the enrichment of the consumed data with links to external valid data sources so that the data consumers can be provided with up-to-date and valuable data. This enrichment has to be performed real time, at the stage of data consumption, so that only up-to-date information is provided as an output. This approach requires creating an inventory of data sources to be used for the various data objects types and their prioritization in order to solve possible ambiguity problems if the same data object can be obtained from different sources but they also provide different values. So that all the necessary online data sources are identified and rated and their entry-points, API-calls, etc. are collected.

The Data Sources and Applications catalogue is a set of extensions to the platform SOML schema. Each data object which can be retrieved from an external source and more specifically the data type is declared as such an extension as well as the service which must be called to retrieve it. If the source is not an SparQL/GraphQL endpoint a specific service which wraps the external source is implemented in order the two-directional data transformation to be achieved. The Semantic Mapper uses these definitions and additionally implemented service to retrieve the needed data. This approach creates a layer between data consumers and actual data sources, allowing all the data accessible within the FSM platform to be considered as a whole. The only thing the consumer will see is that such data (types) exist, they are accessible and can be used for querying. The actual data sources and all communication details remain hidden to the consumer and he doesn't need to be aware about them. This means that the functionality of the platform will depend on third party API versions and any change in the API and/or available data formats could result in malfunction of the data retrieval from the corresponding source, at least until the necessary changes in the catalogue and the corresponding ETL are done. This also means that some measures of early detection of API changes should be taken and that the partners must share information about planned changes of API call formats and/or data accessibility in advance in order the corresponding updates to be performed on time without interruption of the functionality.

Design and Functionalities Overview

The Data Sources and Application Catalogue consists of two parts:

- Set of microservices (API calls) wrapping the external data sources and providing the Semantic Mapper (Apollo Federations Service) with all the needed data. If any data preprocessing is needed, it will be implemented in the corresponding microservice.
- Set of definitions – extensions to the SOML representing the structure of data pieces retrievable from the corresponding (remote) data source and their mapping to semantic objects.

The access to the collected and ingested data in the Data Stage component will be performed in the same manner resulting in a single data model. The data transformation to RDF will be performed by the corresponding microservice.

The data from applications and prototypes, parts of TheFSM platform, will be accessed by their provided API and will not be stored (doubled) in other places in the system. This allows using the same approach as in connection to the external data sources.

Data License and Agreement Management

The Data License and Agreement Manager is the component responsible for handling all processes related to the data licenses and IPR attributes, as well as enabling the drafting, signing, and enforcing the smart data contracts that correspond to data sharing agreements between platform users. This component is provisioned to handle the data exchange and data transformation between the Data Curation and Semantic Enrichment layer (Data staging), the Automated Contract Negotiation and Monetization layer and the Access and Authorization Control Engine. These interactions and aspects are defined in the next subsections below. The way the following descriptions run in succession is slightly more evident, by inspecting the activity diagram describing the monetization process and the **Fairswap** protocol (section 3.2.4).

This component's scope is dual:

1. It will formalize all aspects with which data assets can be shared, traded or otherwise handled to their acquisition. Such aspects include licences, IPR, sensitivity, privacy concerns and even the integrity of the assets' structure.
2. It will allow the creation of machine-readable data assets. It will define the proper expression of licencing terms in machine-readable format, describing with details the interactions of stakeholders in the context of the data sharing scenarios, as well as documenting the platform's expected behavior.

This component is more complicated than a mere set of definitions and general guidelines; it endeavors to provide a good foundation for the marketplace. Consequently, it will also assimilate insights gained through its initial introduction into the platform, discussions with the stakeholders participating in order to reveal concerns and/or scepticism not properly addressed already, technical limitations that could hinder the component's utilization and general coverage of data markets requirements.

Design and Functionalities Overview

The component has the following main functionalities:

- a. Assist and perform a first level of collection of the environmental attributes needed from the A2C Engine. Such attributes include user DIDs, user roles, details about the specific time of a request (due to policies enforcing time constraints for access).
- b. Handle any requests from the Automated Contract Negotiation and Monetization layer to the Data staging regarding the definition and review of the data licenses attached to datasets using all license-related metadata information. The information defined here will be stored in the core platform's storage and will be made available to all other components that need to query it.

- c. It interacts with the platform's blockchain node to report on the validity of smart contracts for asset monetization. Furthermore, it handles all processes required to prepare a smart contract for each (paid) asset transaction and, finally, uploads it to the blockchain.
- d. Enable users to define their IPRs, terms of use of data (price, duration of access etc.).
- e. Offer predefined data licence templates that the users can review and assign to the datasets they own in the platform.
- f. Enables the users to draft their own custom data licenses and assign them to the datasets they own in the platform.

Data sharing model - licencing templates

In order to facilitate the needs for data monetization, proper licences must accompany each dataset. Those need to strictly define the rights of the data owner (the seller), as well as the rights of the buyer. Additionally, constraints imposed by policies such as time constraints (e.g., a dataset can be downloaded upon purchase after 10 days have passed) need to be explicitly stated and enforced by the platform. To that end, when a user marks an uploaded dataset for monetization, they will be given the chance to select a licence fitting their needs from a list of template licences offered by the platform.

Further on, the user monetizing the dataset will be asked to fill in the information the template needs, especially the policies the licence will need to enforce. The user will be assisted via the graphical query builder used throughout the platform for adding new policies to the system. The policies are based on a set of attributes, such as specific DIDs of users, roles and time constraints. When finished, they will also be asked to verify the policy they set up before proceeding.

Data sharing model - encryption

It should be noted that, before a dataset is uploaded to the platform, it will be encrypted before transiting to the platform. Also, the dataset will be stored in an encrypted format to the platform's database. When retrieved again, it will be re-encrypted anew by the platform until it reaches the client, who will then decrypt it to inspect the content. The encryption part is handled by the Data Encryption service.

Data sharing model - monetization

When a dataset is about to be purchased, a new Automated Contract Negotiation event starts, with the Data Licence and Agreement Management component being responsible for its handling. Initially, the component will communicate with the A2C engine in order to get authentication and authorization status for the requester, as well as the policies enforced by the monetized dataset in question.

Once everything is verified, the component will consult with the licencing template filled in for the dataset and will update the access policies of the system by taking into consideration the transaction process, as it progresses. The dataset sold will need to acquire a new DID which will be correlated with the buyer, while some metadata necessary for the smart contract to properly run will be provided by pre-processing steps.

Further on, this information will be passed to the Brokerage Engine and Model, so that the smart contract can run by executing the **Fairswap** protocol between buyer and seller to ensure fairness (the seller will get the agreed amount of tokens for the data asset they offer, while the buyer is guaranteed to purchase the data they were promised, even verifying the asset's integrity before proceeding). If the process is successful, the monetization process will complete, the requested tokens will be transferred to the seller and the buyer will obtain ownership of the data asset, which will however be subject to the restrictions the licence accompanying the original dataset imposes. The relevant workflows are provided in Section 3.2.4.

AI Models

In the context of the project Agroknow will build a number of AI-powered models and algorithms that will enhance the processing, forecasting and predictive capabilities of the platform, so that its users may generate more value from the data assets they use. The predictive services will be available through the Intelligence API of the data platform that will be hosted, operated and maintained by Agroknow. The deployment of the API will enable the integration with the TheFSM data platform and the applications that will be developed in the context of the project.

More specifically the predictive services will include:

- **Supplier & Product Risk Assessment Models:** we will integrate, train and test prediction models and algorithms that will be used for the estimation of risks associated with products, suppliers and critical control points (Task 2.3.1)
- **Incident Prediction Models:** we will integrate, train and test prediction models that will be used for the calculation of incident trends and estimations on upcoming threats (Task 2.3.2).
- **Risk prediction models:** we will integrate models and algorithms for the prediction of risk for ingredients and finished products.
- **Supplier risk prediction models:** we will develop, test and integrate models that will provide the ability to predict the risk of a supplier.

The predictive services will be used by the FOODAKAI 2.0 and the Food Inspector application which will be developed in the context of the TheFSM project. In addition to that any other application and third party system will be able to use the services by gaining access to the Intelligence API. The prediction models will be developed using Python and Deep Learning frameworks and libraries like Keras and Prophet.

A first version of the intelligence API was developed during the first year of the project to support the FOODAKAI 2.0 application.

Query Explorer catalogue

The query explorer catalogue provides various interfaces for accessing the users' stored data in the platform, their semantic representation and linking to various ontologies and data sources in a consistent and unified manner. It interacts with services in Data Curation and Semantic Enrichment components helping the users to focus on the semantic instead of struggling with

various data sources specifics. It also provides a way for interchanging data between platform users featuring data market functionality and allowing data consumers to use the platform in a data source independent manner.

Design and Functionalities Overview

The query explorer catalogue provides various interfaces for accessing the users' stored data in the platform, their semantic representation and linking to various ontologies and data sources in a consistent and unified manner. It interacts with services in Data Curation and Semantic Enrichment components helping the users to focus on the semantic instead of struggling with various data sources specifics. It also provides a way for interchanging data between platform users featuring data market functionality and allowing data consumers to use the platform in a data source independent manner.

One of possible implementations of Query Explorer is the Ontotext Platform GraphQL Playground where skilled users are able to define their own GraphQL queries in a flexible and convenient way and also to explore the Semantic Object Model Language schema of the project.

Most commonly used queries in regards with identified use case scenarios are going to be provided as predefined parameterized queries where the data consumer users will be able to retrieve the required data in an implementation- and schema-independent way.

All the queries will be run with respect to the calling user access privileges and the results will be filtered (reduced) to those objects the user has rights to see.

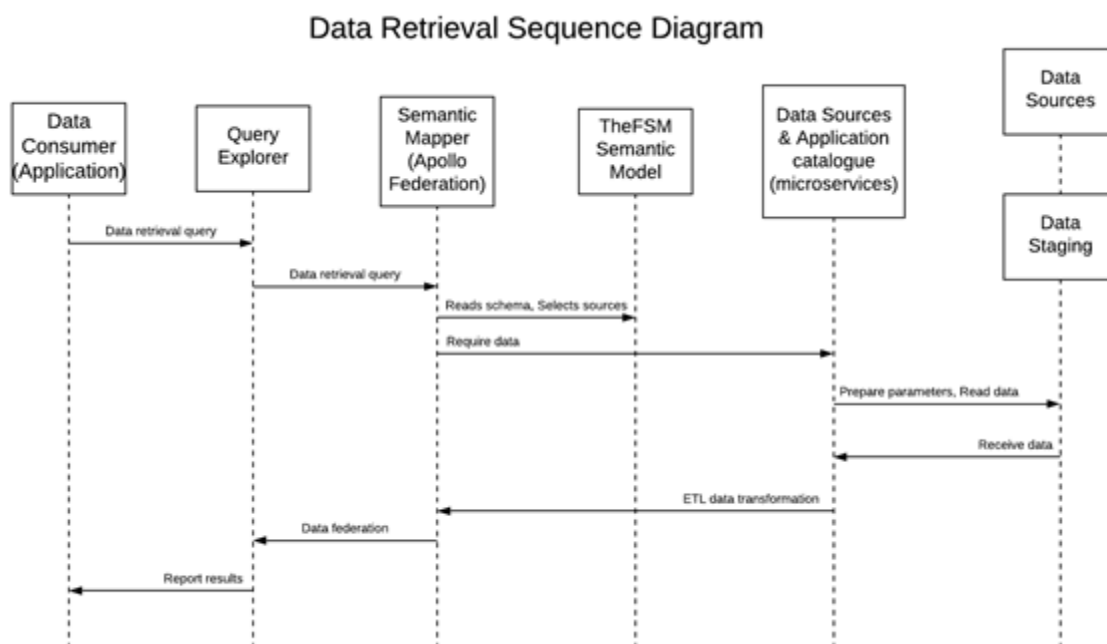


Figure 17: Data retrieval sequence diagram

3.3.4. Identity Management

OTNode DID Services

The **Decentralized Identifiers (DIDs)** are globally unique identifiers designed to enable individuals and organizations to generate self-sovereign identifiers using systems they trust, and to prove control of those identifiers (authenticate) using cryptographic proofs (for example, digital signatures, privacy-preserving biometric protocols, and so on).

A DID identifies any subject (e.g., a person, organization, thing, data model, abstract entity, etc.) that the controller of the DID decides that it identifies. In contrast to typical, federated identifiers, DIDs have been designed so that they may be decoupled from centralized registries, identity providers, and certificate authorities. Specifically, while other parties might be used to help enable the discovery of information related to a DID, the design enables the controller of a DID to prove control over it without requiring permission from any other party.

DIDs are URLs that **associate a DID subject with a DID document** allowing trustable interactions associated with that subject. Each DID document can express cryptographic material, verification methods, or service endpoints, which provide a set of mechanisms enabling a DID controller to prove control of the DID. Service endpoints enable trusted interactions associated with the DID subject. A DID document might contain semantics about the subject that it identifies. A DID document might contain the DID subject itself (e.g. a data model).

A simple example of a decentralized identifier (DID):

did:example:123456789abcdefghi

A DID is a simple text string consisting of three parts, the:

- URI scheme identifier (did)
- Identifier for the DID method (the string “example” in above DID)
- DID method-specific identifier (the string “123456789abcdefghi” in above example)

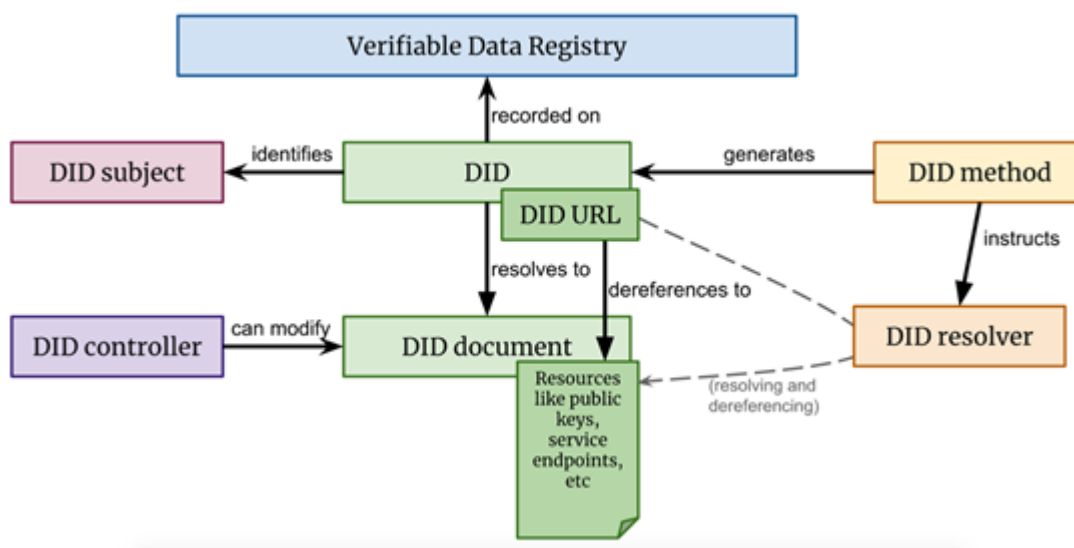


Figure 18: DID Architecture

Design and Functionalities Overview

DIDs and DID URLs

A DID, or Decentralized Identifier, is a fully qualified URI composed of three parts: the scheme "did:", a method identifier, and a unique, method-specific identifier generated by the DID method. DIDs are resolvable to DID documents. A DID URL extends the syntax of a basic DID to incorporate other standard URI components (path, query, fragment) in order to locate a particular resource.

DID Subjects

The subject of a DID is the entity identified by the DID. The DID subject may also be the DID controller. Anything can be the subject of a DID: person, group, organization, physical thing, logical thing, etc.

DID Controllers

The controller of a DID is the entity (person, organization, or autonomous software) that has the capability—as defined by a DID method—to make changes to a DID document. This capability is typically asserted by the control of a set of cryptographic keys used by software acting on behalf of the controller, though it may also be asserted via other mechanisms.

Verifiable Data Registries

In order to be resolvable to DID documents, DIDs are typically recorded on an underlying distributed system or network of some kind. Examples include distributed ledgers, decentralized file systems, databases of any kind, peer-to-peer networks, and other forms of trusted data storage.

DID documents

DID Documents describe the public keys, authentication protocols, and service endpoints necessary to initiate trustworthy interactions with the identified entity. A DID Document is a JSON-LD document that contain the following six, optional components:

- The DID that points to the DID Document, identified by the key id.
- A list of public keys identified by the key publicKey.
- Lists of protocols for authenticating control of the DID and delegated capabilities identified by the key authentication.
- A set of service endpoints, usually URLs, that allow discovery of a way to interact with the entity that the DID identifies by the key service.
- Timestamp indicating when the DID Document was created and updated for auditing the DID Document identified, respectively, by the keys created and updated.
- A digital signature for verifying the integrity of the DID Document identified by the key proof.

The DID Document is the root record for a decentralized identifier that can reference not only what's in the DID Document itself, but also any information from the service endpoints. This is accomplished by adding selectors, paths, query parameters, and fragments to the DID.

DID Methods

DID methods are the mechanism by which a particular type of DID and its associated DID document are created, resolved, updated, and deactivated using a particular verifiable data registry. DID methods are defined using separate DID method specifications.

DID Actions

DIDs are typically used between DID Controllers and relying parties (also called verifiers). The following groups of functionalities are possible with DIDs:

- **Create features** – provisioning DIDs and corresponding DID documents (done exclusively by the DID controller and according to DID method specification)
- **Delete features** – deleting DID material, performed exclusively by the DID controller
- **Update features** – the features for updating DID documents, such as rotating keys, modifying service endpoints, migration and recovery, performed by the **DID controller**
- **“Use” features** – enable presentation, authentication and cryptographic signing by the controller, and verification and auditing by the relying party (the verifier)
- **“Read” features** – corresponding to DID resolution and dereferencing, by definition performed by the relying party



Figure 19: DID actions overview

DIDs features within TheFSM architecture are implemented through the **OTNode DID Services: Identity Hub, DID resolver** and **DLT system** components, explained further below. DIDs may be used according to the DID Auth protocol.

DIDs are utilized in conjunction with the Verifiable Credentials data model, which enable utilization of DIDs in provisioning, sharing and verification of generic verifiable claims in the below illustrated ecosystem. Verifiable credentials are cryptographically secure, privacy respecting and machine verifiable. The recommendation is to implement a verifiable credential scheme for datasets exchanged via TheFSM platform.

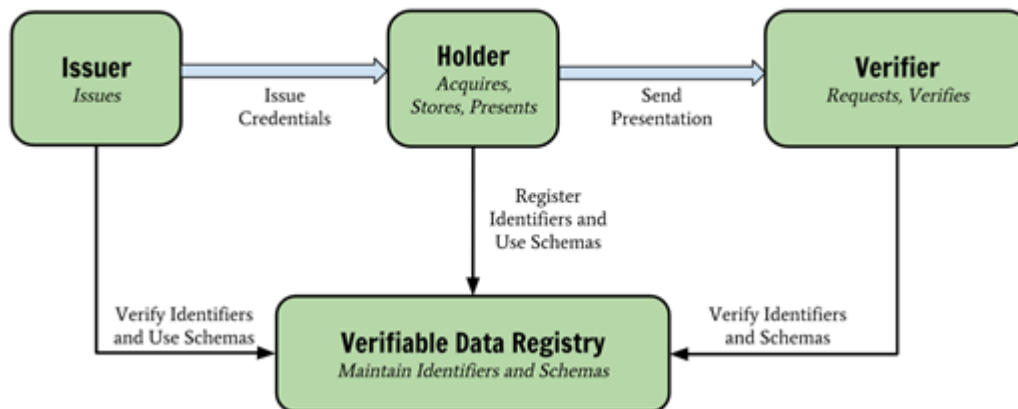


Figure 20: Verifiable Credentials ecosystem

Identifiers Registry (DIF Identity Hub)

The identity hub is used to securely store verifiable credentials. It is a datastore containing semantic data objects at well-known locations. Each object in a Hub is signed by an identity and accessible via a globally recognized API format that explicitly maps to semantic data objects. Identity hubs are addressable via unique identifiers maintained in a global namespace and are associated with the Decentralized Identifier standard described previously in the document.

A single entity (DID subject) may have one or more instances of a Hub, all of which are addressable via a URI routing mechanism linked to the entity's identifier. Hub instances sync state changes, ensuring the owner can access data and attestations from anywhere, even when offline.

DID Descriptor Objects (DIF Universal Resolver)

The DID resolver component is responsible to resolve the conforming DID documents based on the specific DID method and to verify the resolution result. By specification the DID resolver methods are implementation specific (based on the specific DLT). The DID resolution functions resolve a DID into a DID document by using the "Read" operation of the applicable DID method. DIF's Universal Resolver provides a unified interface which can be used to resolve any kind of decentralized identifier.

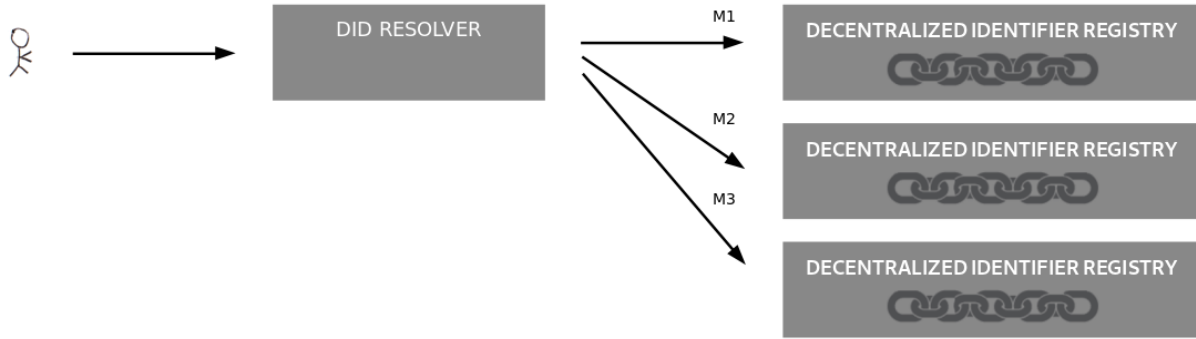


Figure 21: DID resolution

The relevant sequence diagrams for the DID provisioning and resolution are provided below:

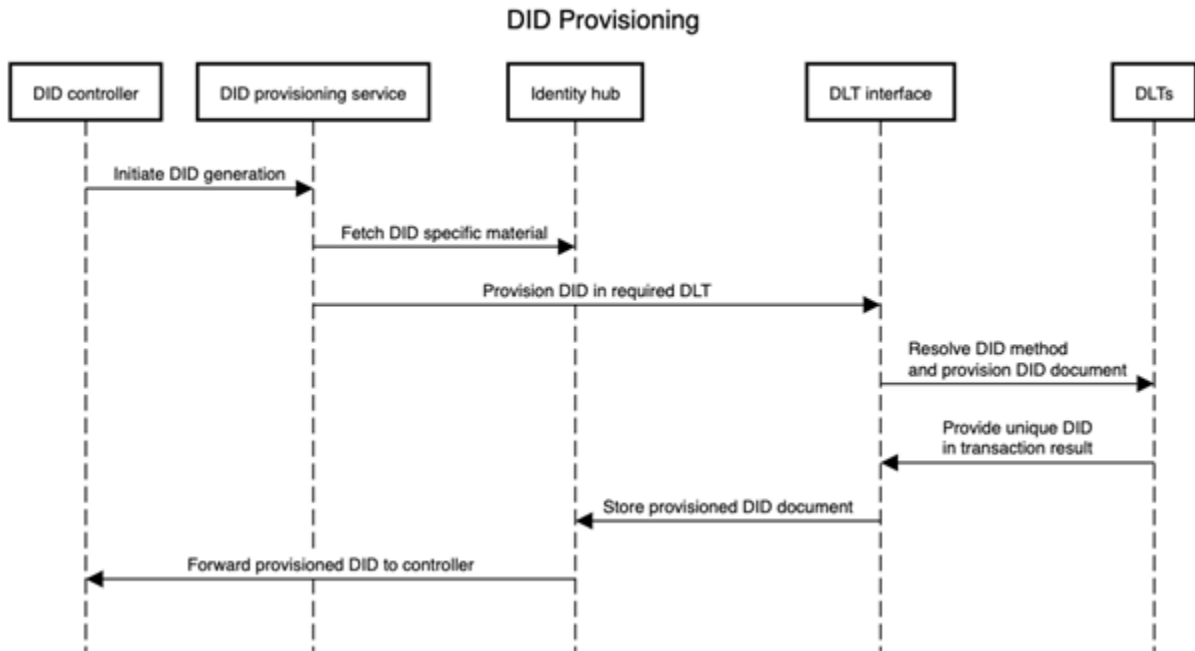


Figure 22: DID provisioning

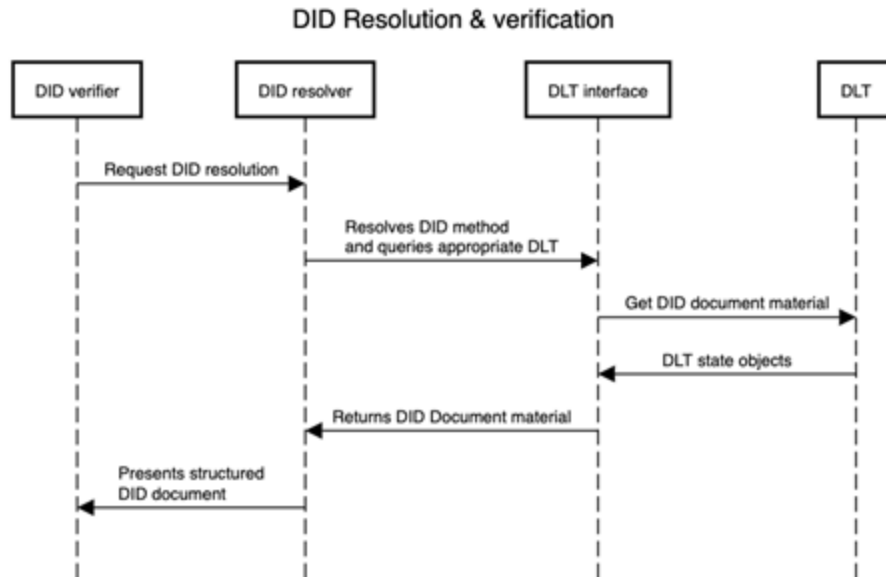


Figure 23: DID resolution and verification

TheFSM Platform is agnostic to the utilization of blockchain for DID implementation and due to the reliance on standards enables a wide variety of DID agents to be used.

OTNode DLT interfaces

The DLT data management interface is used to abstract the details of underlying specific DLT implementation in order to provide common functionality for the Identity hub and resolver components. It implements two generic functionalities – querying DLT state and publishing transactions, which include interactions with DLT smart contracts if the DLT implementation supports it.

The OriginTrail Network node presents an implementation of a multi-DLT interface (ODN, Ethereum, Hyperledger, and other blockchains in the future such as Polkadot) together with a supply chain data specific identity hub, storing semantic data in the standardized supply chain form (according to GS1 standards mostly). TheFSM platform will utilize OriginTrail as is for the DLT, DLT interface and Identity hub components.

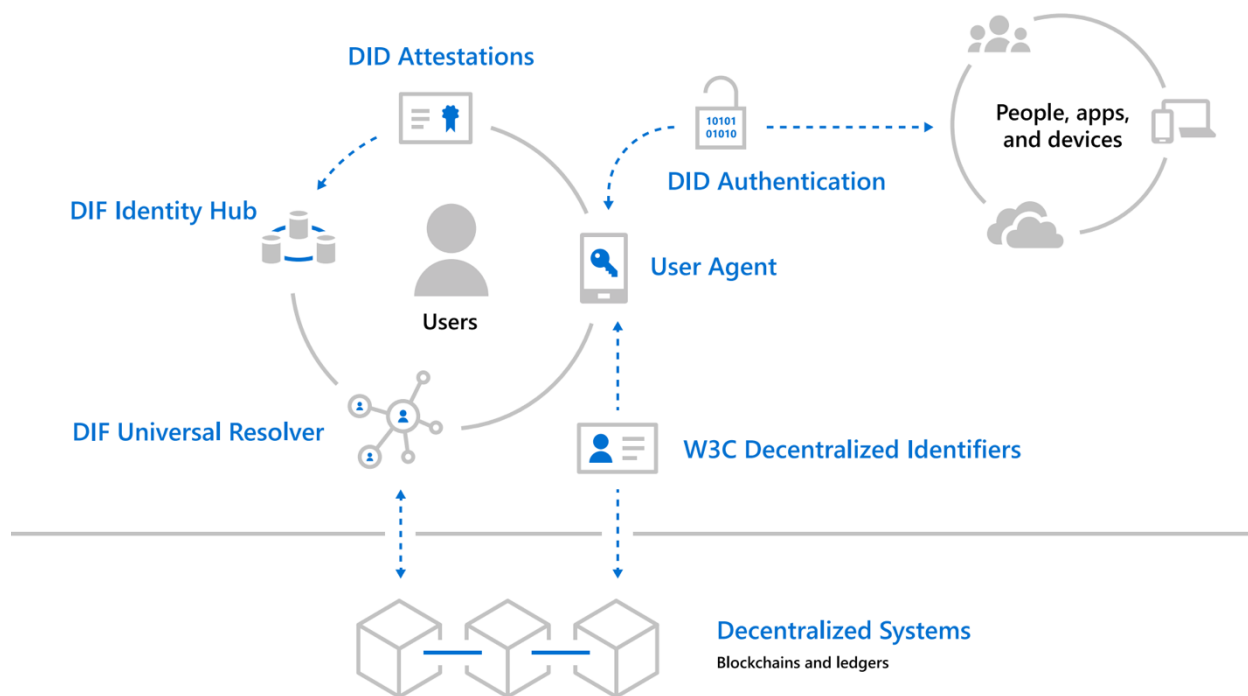


Figure 24: High level presentation of OriginTrail.

Design and Functionalities Overview

DLT

A distributed ledger (DLT stands for distributed ledger technology) is a replicated database that is consensually shared and synchronized with a specific protocol across multiple sites, institutions, or geographies, components of which are typically owned and accessible by multiple entities. The key property of DLT technology is that it is “decentralized”, meaning being maintained in such a way that no central service or authority is needed to operate the system and broker transactions between participants.

One type of DLT technology is blockchain, named by the specifics of the protocol by which data is replicated and shared between DLT stakeholders (as a series of linked, cryptographically verifiable blocks of data, hence block-chain). DLTs key characteristics are achieving high resilience and increased data integrity, which is why it has been utilized in many enterprise use cases such as supply chain visibility and trade finance.

OriginTrail is a purpose-built, open network for cross-organizational data sharing in supply chains, supported by blockchain. The OriginTrail protocol has been designed to support and grow the global linked-data-first decentralized knowledge graph (DKG) to enable interoperable, trusted data exchanges. The OriginTrail DKG is therefore growing according to emerging W3C and GS1 standards to support multiple functionalities for DIDs, verifiable credentials and enterprise data

sharing, supported by consensus-enabling protocols as the trust foundation in data exchange. As such, the design of OriginTrail is based on a blockchain-agnostic approach for the long-term evolution of the technology, being able to leverage the progress of blockchain ecosystems.

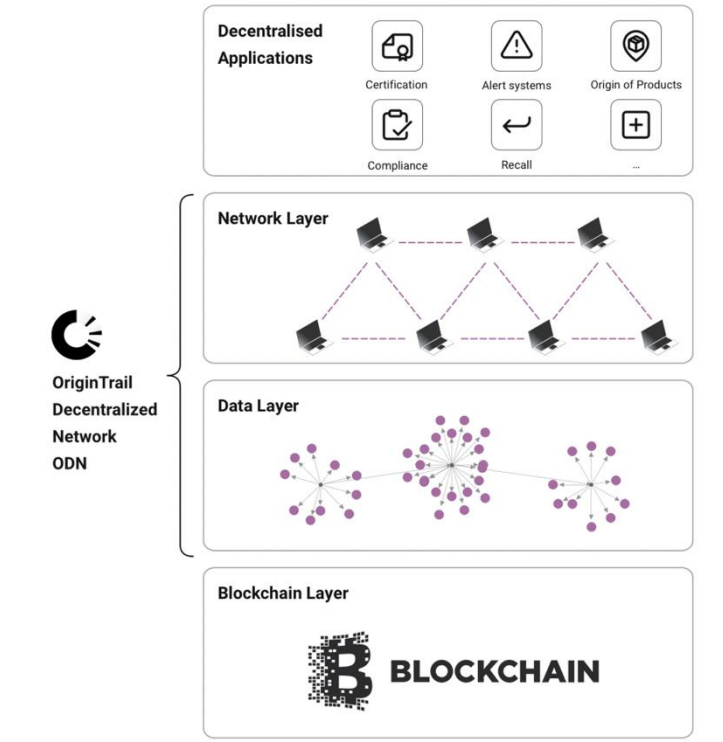


Figure 25: High level architecture of OriginTrail framework

The workflows for publishing traceability data as events and querying traceability info are presented in the next Sequence Diagrams:

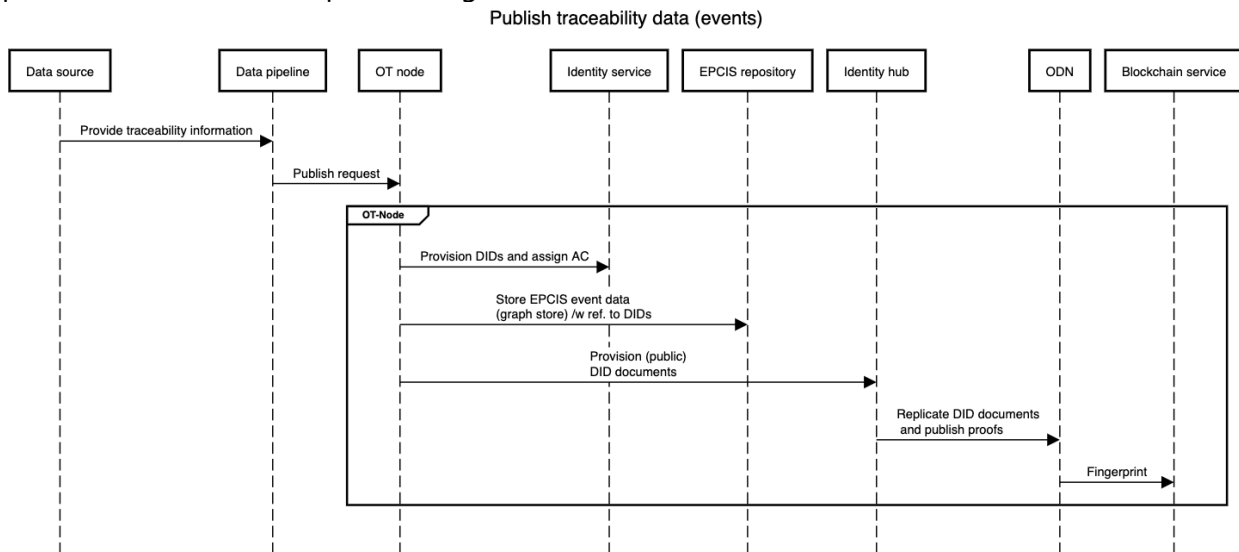


Figure 26: Sequence diagram illustrating the workflow for publishing traceability events.

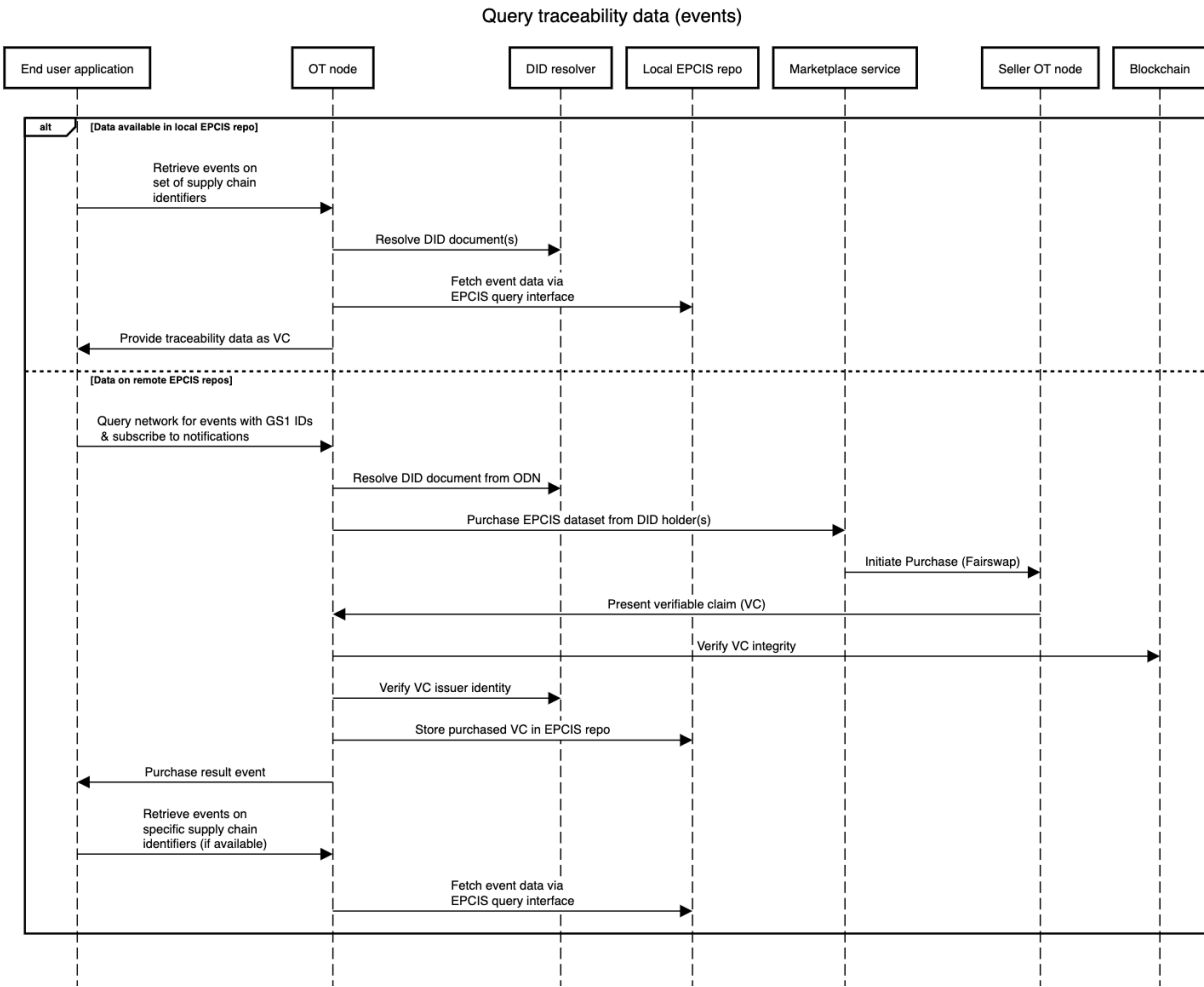


Figure 27: Sequence diagram illustrating the workflow of querying traceability data.

3.3.5. Automated Contract Negotiation and Monetization

Data Brokerage Model

Based on the initially identified requirements, the Framework will be built on top of three core entities, namely the **Data Asset**, the **Policy** and the **Contract** and two supporting entities, namely Attributes and Terms, TheFSM defines a specific dataset from a data provider. A Data Asset, at least in the Framework’s first version, corresponds to a single file which will either be already in or be easily formatted in a tabular form, i.e. comprising rows and columns or text. There is no separate entity to express the concept of DaaS, as these will be offered through sharing agreements that foresee updates and not through real-time data streams.

A Policy is the way all legal, IPR, license, quality etc. terms are expressed. Each Data Asset specifies a number of Policies which control how it can be shared and accessed. A Policy comprises a group of terms and/or attribute guarantees. Terms are specific prohibitions, permissions or obligations

stemming from the above-mentioned aspects, whereas Attributes are expressions of certain facts and/or qualities, e.g. the date a Data Asset was created.

Finally, contracts represent the official data sharing agreements between a data provider and a data consumer in regard to one single Data Asset under specific Policies.

The Data Brokerage Model will be used by the Data Brokerage Engine to execute the data exchange between the two parties and will be instantiated by the Data Licence and Agreement component.

The component will be further analyzed in the second version of the architecture, as soon as, the design and the technical architecture of the security components, Data License and Agreement and identity management will be further elaborated.

Data Brokerage Engine

The data brokerage engine component is responsible for generic brokering of datasets in TheFSM platform between different parties and for possible financial compensation. The implementation of the Data brokerage engine will be based on OriginTrail protocol for data exchange, having OriginTrail nodes perform the function of the semantic store and data marketplace validation operations, interacting on the basis of smart contracts which solve the problem of fair data exchange between the dataset seller and the dataset buyer. This process is performed without reliance on a third party in between to guarantee the fair result of the transaction. The Data brokerage engine will enable the following two guarantees:

- The data seller can be **guaranteed to receive compensation** for a sold dataset
- The data buyer can be **guaranteed to receive a verifiable dataset** requested in the purchase

The OriginTrail data brokerage component implements the *FairSwap protocol*³ with formally proven security, defined and specified by the researchers at TU Darmstadt and the University of Warsaw. The protocol ensures a fair exchange of data for tokens, enabling a data seller to sell a digital commodity for a fixed price of tokens to the buyer.

The Marketplace FairSwap protocol is currently implemented in the form of Ethereum smart contracts, integrated together with the Ethereum identity smart contracts (conforming to the ERC725 standard, conformant to W3C DID standard) and OriginTrail protocol data replication incentivisation smart contracts. The dataset exchange is at the moment partially based on the *W3C Verifiable Credentials* framework, focusing on the broad use case of Verifiable Claims - enabling public viewing of the dataset **metadata** and **proofs**, and private storage of **data claims** which are offered for sale for a specific compensation in tokens.

Cryptographic Wallets

³ <https://eprint.iacr.org/2018/740.pdf>

Crypto wallets are native components of DLT systems used to perform system transactions, most often of cryptographic tokens, however also enabling more complex transactions through the interaction with smart contracts (blockchain native programs). A wallet is implemented as a device or program which holds cryptographic public/private key pairs in a secure way, and usually enables encryption and signing of information using mentioned keys.

Several different wallet types can be observed:

- basic cryptocurrency wallet, which allows for interaction with their native blockchain by sending transactions and running smart contracts. A popular in-browser example is Metamask
- eID wallets, which are designed to extend the functionality to securely store credentials according to the European Self-Sovereign Identity Framework. Popular examples include Hyperledger Indy project wallets
- multisignature wallets, where multiple parties control the wallet operations, which have been designed for additional security
- smart contract wallets, which enable additional custom features of smart contracts

TheFSM platform intends to provide a wide support for crypto wallets, focusing on interoperability and utilization of the emerging wallet ecosystem.

TheFSM Data brokerage has been designed for extensibility based on prior mentioned standards and models. The core of the data brokerage engine is the implementation of trusted smart contracts based on the Fairswap protocol, enabling trusted exchange of data for cryptographic tokens in a decentralized environment without a trusted data broker. However due to the nature of TheFSM platform and its central point in the Food Safety Market, an additional simplified workflow has been proposed to allow for a modified (Fairswap compatible) approach to increase efficiencies of transaction costs and lower the complexity in the scenarios where TheFSM platform is inherently a data brokerage "middle man". The following diagrams showcase the detailed flows of the modified and fully fledged Fairswap protocol implementations.

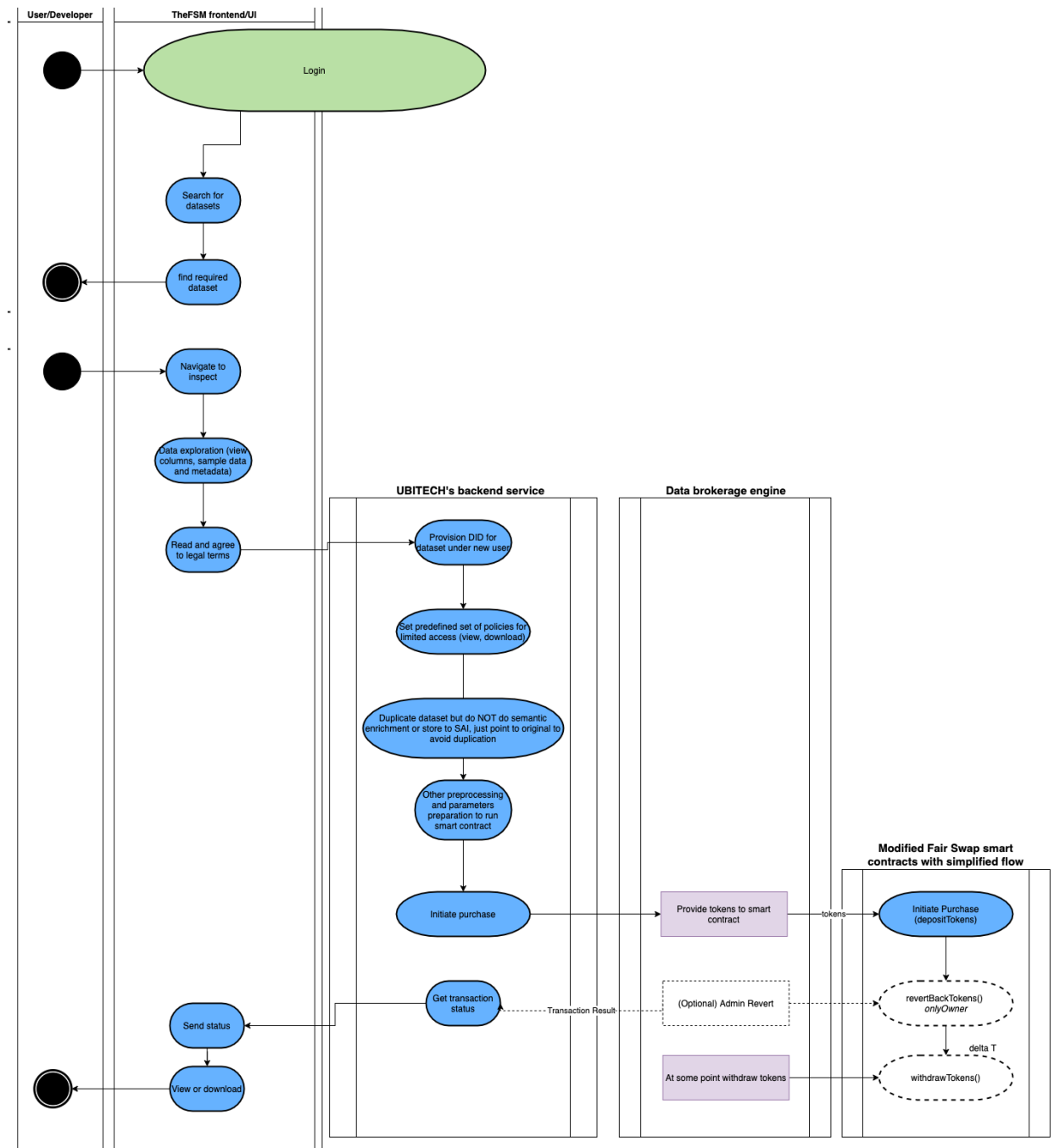


Figure 28: Activity diagram illustrating the data monetization process.

The workflow of the FairSwap protocol which enables fair data exchange for tokens is presented in the next Sequence Diagram:

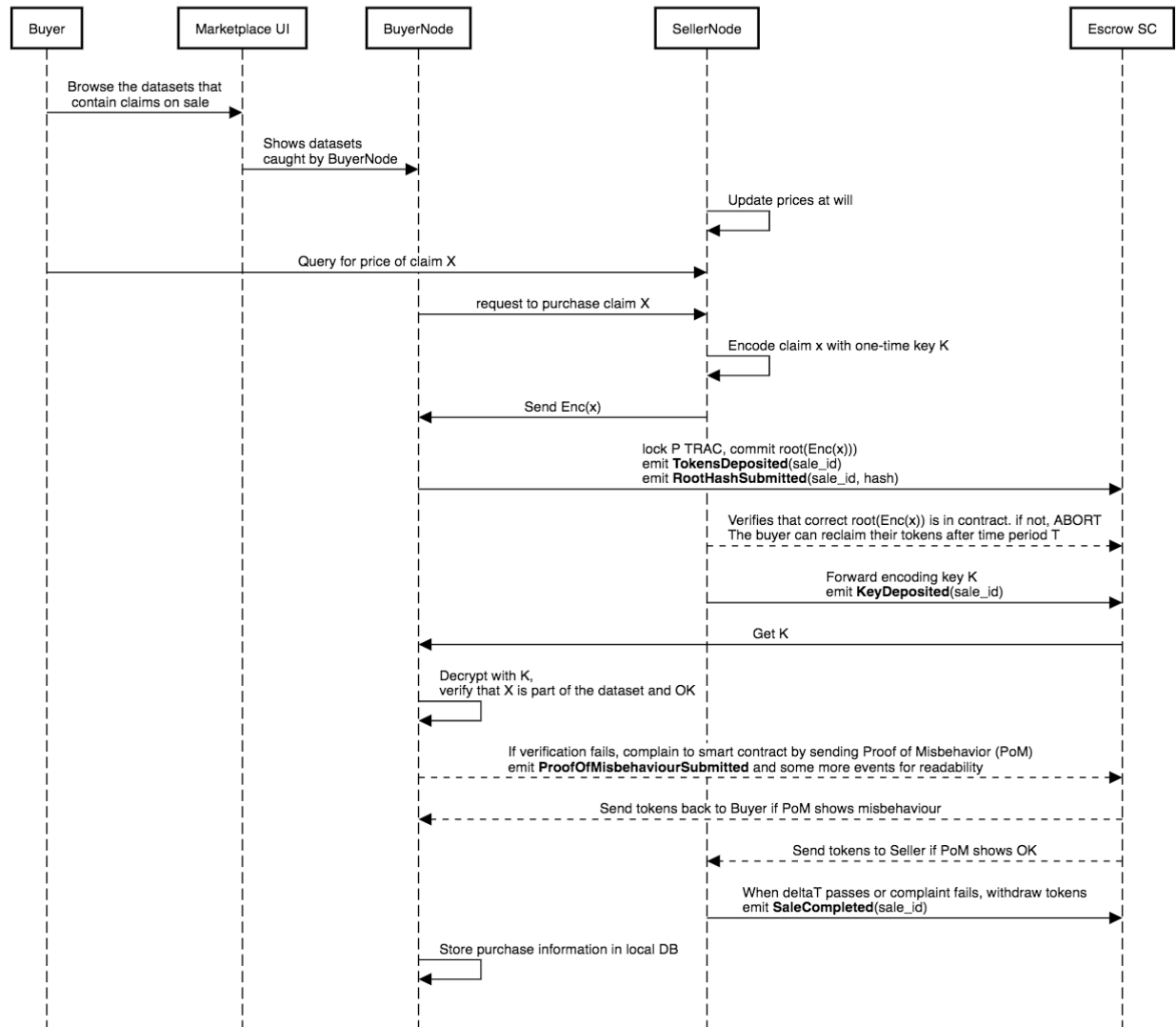


Figure 29: Sequence diagram illustrating the FairSwap protocol.

3.3.6. Security and Access Control

This subsection is dedicated towards describing two important technologies which are utilized in unison, in order to ensure proper authentication and authorization when accessing resources throughout TheFSM platform. The first, ABE (Attribute based encryption) addresses the issue of encrypting documents and data according to a set of attributes, while ABAC (Attribute-Based Access Controller) addresses the authorization aspect of accessing resources, based on both environmental and user-specific attributes. The two technologies will be integrated with each other in the future and we will thoroughly document this process.

Authentication

Technologically, there are well-known industry standards enforcing this, such as OAuth⁴ and JSON Web Tokens⁵ (JWTs).

OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. This mechanism is used by companies such as Amazon, Google, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites. Generally, OAuth provides clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without providing credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner. The third party then uses the access token to access the protected resources hosted by the resource server.

JSON Web Token (JWT) is an Internet standard for creating data with optional signature and/or optional encryption whose payload holds JSON that asserts some number of claims. The tokens are signed either using a private secret or a public/private key. For example, a server could generate a token that has the claim "logged in as admin" and provide that to a client. The client could then use that token to prove that it is logged in as admin. The tokens can be signed by one party's private key (usually the server's) so that party can subsequently verify the token is legitimate. If the other party, by some suitable and trustworthy means, is in possession of the corresponding public key, they too are able to verify the token's legitimacy. The tokens are designed to be compact, URL-safe, and usable especially in a web-browser single-sign-on (SSO) context. JWT claims can typically be used to pass identity of authenticated users between an identity provider and a service provider, or any other type of claims as required by business processes.

ABAC model, Authentication & Authorization Engine

Initially, we had proposed the utilization of the XACML specification to represent the ABAC model in TheFSM platform (for reference, kindly consult with D3.1 v1 - Open Reference Architecture). However, during development, we realized a few shortcomings of XACML:

- It uses too many components that need to be implemented to cover the standard, which adds immense complexity to the project.
- The XACML specification processes rules and policies in XML, which is very verbose. This raises concerns about data storage of rules and policies, as well as efficient parsing and processing.
- Due to the XML format of the policies, they are hard for humans to read.
- The attributes required by TheFSM's ABAC model can be represented in other alternative specifications.

⁴ <https://oauth.net/2/>

⁵ <https://jwt.io/>

To that end, we utilize the PERM meta model which is already built-in in JCasbin, the ABAC library used for authorization in TheFSM.

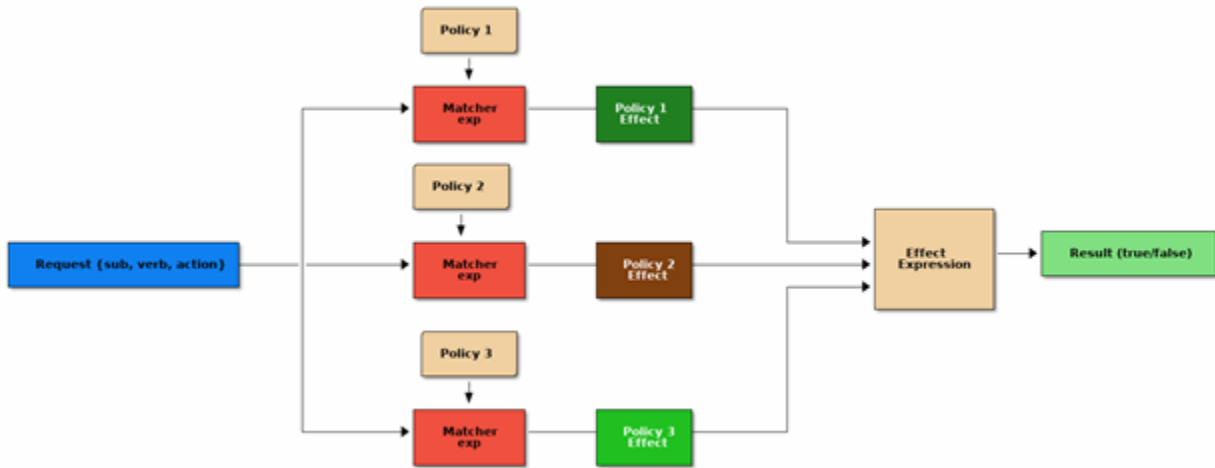


Figure 30: PERM meta-model

In this model, each request is essentially a triplet:

Subject: This is a POJO object containing all attributes uniquely defining the user.

Verb: It represents anything the request is acting on. For our purposes, this is a **resource**. Resources are represented as URLs (due to REST principles).

Action: The desired action of the request.

Casbin’s meta-model contains a set of user-defined policies, each policy including the following:

A condition that must be met, combining boolean conditions of the user’s attributes (not all of them are necessary to appear in the policy, but they can).

- The resource protected by the policy.

- The action involved with this policy.

- The effect of the policy (allow or deny access).

As shown in the figure above, when a request arrives, the engine will filter the request against policies which match a regular expression (simply put, the request must match with relevant policies, i.e. accessing the same resource, with the same action, boolean conditions met). Then, we only keep the matching policies and evaluate the results in order to obtain their effect. Finally, the effects will indicate whether the request will be granted or not, depending on the effect expression.

The effect expression is also known in other schemes as the unification algorithm of the policies. It defines whether the access request should be approved if multiple policy rules match the request. The supported effect expressions are:

- **Allow-override:** If even one matching policy has allow as effect, the request is granted access.
- **Deny-override:** If even one matching policy has deny as effect, the request is immediately denied access.
- **Allow-and-deny:** If a request has at least one matching policy with allow as effect and no deny effects are matched as well, the request is granted access.
- **Priority:** The first effect encountered (allow or deny) determines whether the request will be granted access.

In order to maximize flexibility and ensure robustness of policy evaluation, we use the allow-and-deny effect expression. Additionally, to guarantee the maximum amount of fine-grained control in FSM, we use ABAC. Currently, the attributes accompanying a user are:

- Their DID (unique identifier)
- The UNIX timestamp of the request
- The user's roles
- The year of the request
- The month of the request
- The day of the request
- The day name of the request (this is useful for policies denying access in specific days by name)
- The hour of the request
- The minute of the request

A few of the workflows presented in section 3.2 involve the ABAC engine and how it authorizes requests. Compared to the first iteration of the platform, ABAC has evolved to process and parse parametric resources, such as RESTful resources. "Star" policies (subject, resource, action or effect being anything as an acceptable value) are also introduced into the second iteration of the platform.

Data Encryption

An important facility offered by the platform is data encryption. Due to GDPR regulations and due to dealing with sensitive information, legal contracts and protected datasets, it is paramount to ensure the data remains protected. To that end, we guarantee that while data is being transferred between the end user and the platform, they are always encrypted. In the next parts, we describe encryption schemes, discussing our proposed protocol.

RSA Encryption

When encrypting data, one of the most frequently used ways is RSA encryption, owing its name to its conceivers (Rivest, Shamir and Adleman). In RSA encryption, each member of the protocol has a pair of keys, a private and a public key. The private key must be kept private at all times; only its owner should have it. On the other hand, the public key is available to all interested parties. When needing to send a message to someone, the member asks for their public key and encrypt the message using the public key they received. When the other member receives the message, they decrypt it using their own private key. The stark difference of how both keys are utilized in the protocol is the reason this encryption is also known as Asymmetric Encryption.

To provide a better understanding of how the key pair functions, we can imagine a magical treasure chest which has a special lock and contains a special message. The public and private key interact with the lock differently: the public key can turn the lock clockwise 90 degrees, while the private key can turn the lock counter-clockwise 90 degrees and neither key can interact with the lock in a different way. If the lock is straight up, the chest is unlocked and anyone can see the message. However, if the public key is used, the chest is locked and the message is safely hidden inside. Only by using the private key can the public key's action be reversed, by turning the lock counter-clockwise in order to again unlock the chest.

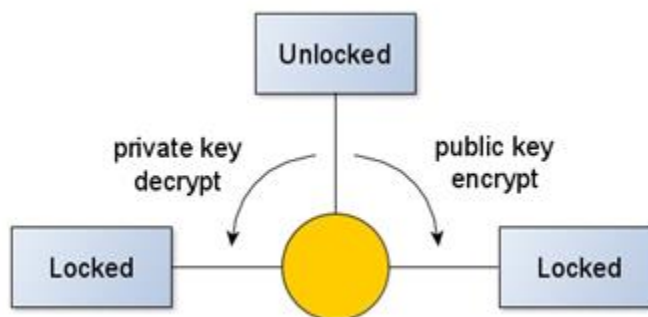


Figure 31: RSA encryption represented as a lock.

Symmetric Encryption

Following the previous description of RSA, it is paramount to explain why it is not enough for the platform's needs. A major shortcoming of RSA is that, in order to encrypt a message, the key needs to be almost as large as the message itself, raising serious issues during data transfer and size.

RSA's shortcoming can be overcome by another scheme, Symmetric Encryption. Instead of having a pair of keys per member of the protocol, all involved parties should use the same key and which they have obtained before the protocol starts. It is called symmetric encryption because the same key both encrypts and decrypts.

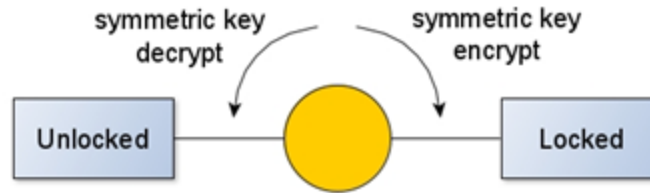


Figure 32: Symmetric encryption represented as a lock.

Attribute-Based Encryption

Finally, for the sake of completeness we present an even more advanced encryption scheme and justify why we opted against it. Attribute Based Encryption (ABE) is a concept introduced in 2004 by Sahai and Waters [1]. Based on IBE, it allows a user to encrypt data so that it can only be decrypted by users with certain attributes. Specialized protocols and adaptations of ABE exist, such as FAME (Fast Attribute-based Message Encryption), DAC-MACS (Data Access Control for Multi-Authority Storage Systems), RD-ABE (Revocable and Decentralized Attribute Based Encryption), CP-ABE (Ciphertext-Policy Attribute Based Encryption) and KP-ABE (Key-Policy Attribute Based Encryption). In many ways, this is similar to the ABAC protocol in request authorization. Once again, a user must have a specific set of attributes validated against a complex boolean policy, being able to decrypt the data iff they match the policy filter. While it adds even more robust data encryption on top of our proposed hybrid protocol, there are a few reasons we opted against it.

The first reason lies in an inherent problem ABE protocols have, which is the immutability of the initial attributes defined by ABE. Once these attributes are set, encryption takes them into consideration and therefore any change to the attributes immediately invalidates the entire protocol.

Furthermore, ABE schemes have difficulties with access revocation due to the encryption depending on a setup phase for the protocol. Once a user has the proper attributes to decrypt a resource, even if they are revoked access, they can still decrypt the resource.

Additionally, due to the nature of ABE schemes, it is difficult to have multiple authorities enforcing the protocol. While some versions of ABE remedy this, it still adds multiple layers of complexity on an already complex protocol.

Moreover, some attributes in policies fitting for a Data Market will naturally involve information such as dates. Keeping in mind that ABE cannot add or remove attributes once set up, it is impossible to have such policies in the encryption level.

API Gateway

The API gateway is a new component introduced in the second iteration of the platform. Generally speaking, an API Gateway is an API management tool which sits between a client and a collection of backend services, which serves as a reverse-proxy accepting all API calls, aggregating services (if necessary) and returns the obtained result.

The necessity of the API Gateway for the project originated from two aspects:

- A way to share data assets via APIs, as well as discovering these APIs themselves was required.
- General services such as analytics, third parties offering their own services as part of the platform for added value can be integrated into the platform via the API gateway.

Apart from those aspects, additional reasons we opted to introduce the API Gateway are the following:

- Protection of APIs from overuse and abuse. This is ensured by authenticating and authorizing requests.
- The potential of running analytics on top of requests.
- The potential of monetized APIs and billing.
- The ability to call multiple micro-services to cover the needs of a single request. Due to having many micro-services throughout TheFSM, it is necessary to call many of them for a single request (e.g., semantic enrichment of data returns to the requester).
- Addition, removal or update of all services is handled at this single point.

The API Gateway is a backend service which can also be handled via the platform's user interface for convenience. When a new API is added to the gateway, the API is defined as a new **endpoint**, requiring the user to provide the URL of the API (parametric URLs for REST APIs are fully supported as well), the method of the API (POST, GET, DELETE, etc.), a small description, the authentication method TheFSM needs to use to call the API (e.g., the API in question could be protected by API key or JWT) and the service under which this API is provided (services are a level above **endpoints** to group APIs of the same provider and to make searching for APIs easier). Once the API is added, the user can then set up ABAC policies to restrict access to that API. These policies are subject to the exact same constraints as the policies enforced everywhere else throughout the platform.

When data assets are offered by API instead of being static and uploaded directly to TheFSM platform, they are going to be automatically added under the hood to the API gateway as services (the user will of course be notified and asked for approval of that action beforehand), enabling data exchange subject to regular ABAC policies.

The robustness of the gateway is evident by the workflow which illustrates how a request is processed, when calling the API gateway (also see Figure 8: API Gateway request execution activity diagram.):

- The user interested in consuming data from a specific API calls the API gateway, asking it to call on their behalf the desired API (they are responsible for providing their own JWT, the URL of the API they want to use, its method and also any body that should be sent, for POST/PUT requests).

- The API Gateway will use the user’s JWT to filter the request before submitting it, to ensure the requester is both authenticated and authorized by the platform.
- Upon successful authentication and authorization, the gateway will use all input provided by the requester and will call the API in question. If the API itself is somehow protected, it will use the declared API key or JWT as part of the request.
- Once the request returns the response to the API Gateway, it will return the response to the caller.

Anonymization Framework

The anonymization framework did not change when compared to the first iteration of the platform, so we merely provide its documentation exactly as is from the first iteration, for the sake of completeness. Due to the sensitive nature of the data hosted on TheFSM, it is paramount to provide pseudonymisation and anonymisation capabilities. Pseudonymisation refers to procedures where sensitive data are mapped to generic values, so that they can be protected, while anonymization maps sensitive data to generic, random values. Assuming “X” was originally ID=3, the main difference lies in the fact that with the former it is still possible to deduce that “X” refers to the same information every time “X” is encountered, while the latter can map 3 to “X”, “Y”, “Z” for every time it is encountered throughout the text.

Pseudonymisation obviously exposes some knowledge about the original data, however this can be useful. For example, risk estimation can take into consideration sensitive data about companies which are pseudonymized and conduct a thorough analysis, without ever exposing their identities.

Design and Functionalities Overview

The initial approach for FSM is the utilization of pseudonymisation, which can then be adapted into full anonymization, should the needs of the project require it.

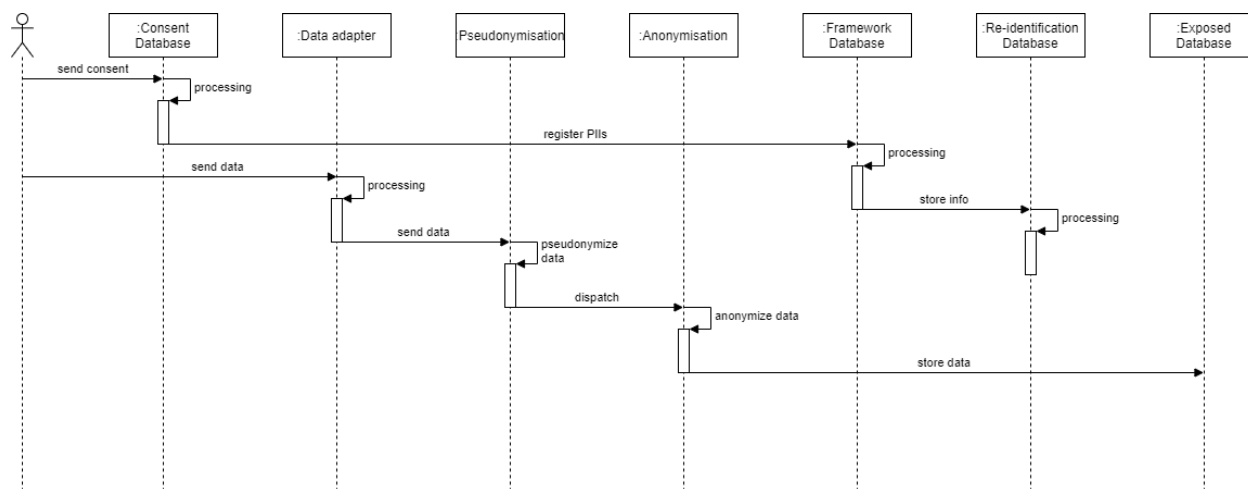


Figure 33: Pseudonymisation/Anonymisation process sequence diagram

The Anonymisation component is responsible to implement the pseudonymisation and anonymisation of the platform data. The component includes the following sub-components:

Consent database: A database which stores the data subjects who have provided consent to the TheFSM platform.

Framework database: A database which contains the PII (Personally Identifiable Information) of all the data subjects.

Re-identification database: A database which contains the original data of the data subjects or other data which can be used to match the pseudonymised (or anonymised) data to the data subjects. These data need to be pseudonymised (or anonymised) and their access is restricted only to the authorised personnel.

Exposed database: A database which contains the pseudonymised data which are accessed and disseminated to the various parties which use TheFSM.

Pseudonymisation: A component which will perform pseudonymisation transformations on the data.

Anonymisation: A component which will anonymise the data.

Data adapter: A software component which is responsible to implement the pseudonymisation of the data.

The pseudonymisation process is briefly described below:

When collecting personal data, the Data Adapter will query the Consent database and the Framework database. The consent database will have stored a map of all subjects that have provided consent to TheFSM. The Framework database will contain the PII of all data subjects. If confirmation from the consent or the framework database occurs, the Pseudonymisation Module will perform pseudonymisation on the data; it will store the pseudonymised data in an open dataset that can generally be accessed by parties being in communication with TheFSM and will store the re-identification data in a separate database; the Re-identification database. The Re-identification database will not be publicly accessed but will be used and maintained by each of the data controller's users. When re-identification is needed at run-time (e.g. when the e-mail of a user needs to be verified), the Pseudonymisation Module will communicate with the Re-identification database to obtain the original data; apart from this case, access to the re-identification database will be restricted.

After storage, an extra *Anonymisation* module will provide the functionality of generating anonymised data from the exposed data set. The implementation of the anonymisation module will be based on the **ARX Framework**⁶ and will produce a data set with high *k-value*, *l-diversity* and *t-closeness* parameters. In case the platform operator imports a population table, the *Anonymisation* module will also produce a low value of δ (for the specifics of *k-value*, *l-diversity*, *t-closeness* and δ -difference). The anonymised data set will contain all useful information regarding user actions and cases and can still be used to compute analytics and provide useful feedback.

⁶ <https://arx.deidentifier.org/>

Since data subjects cannot be de-identified from the anonymised data set, it can be stored or archived regardless of the status of consent forms.

In case that a subject is removed from the framework database or a consent is revoked, the Pseudonymisation Module will remove for this subject the re-identification data from the re-identification database. The pseudonymised data will be automatically converted to anonymous data upon this removal, so they can still be stored in the Exposed database. Upon revocation of consent, the deletion of re-identification data may take some time due to the system having to poll the consent database and the technical expert receiving the notification to delete re-identification data. This will be explicitly noted in the consent form.

The *Pseudonymisation* module will perform a combination of techniques. The administrator of the platform will be able to define which transformations are needed to ensure proper pseudonymisation or anonymisation.

The set of transformations offered will consist of both one-way hashes⁷ and two-way encryption (possibility to encrypt and decrypt the data) as well as all the data masking techniques, except from shuffling. The reason that shuffling is excluded is because it couples data of multiple subjects. If one subject revokes consent, it is difficult to undo the transformation without affecting data corresponding to other subjects.

3.3.7. Data Marketplace and Added Value Services

A value-added service (VAS) is a popular telecommunications industry term for non-core services, or, in short, all services beyond standard voice calls and fax transmissions. However, it can be used in any service industry, for services available at little or no cost, to promote their primary business. In the competitive markets, these services have a significant importance. For instance, in terms of revenue, these services provide a significant amount of money to telecom companies by enabling them to upturn average revenue per user.

On the other hand, they enable operators to establish customers' loyalty. Customers are attracted to platforms companies that offer more VASs. These services also make customers happy. And, customers are more likely to continue using services of the company that makes them happy. Thus, these services play a significant role in ensuring customer satisfaction and retention.

For TheFSM, a main goal is to provide a set of added value services to empower the food safety and certification industry which will address: a) Provision of data to stakeholders through intelligent query engine, b) Data sharing and monetization services, c) Traceability data services, d) support the analytics algorithms workflow design and execution, by making the best known and widely accepted algorithms in the food safety and certification industry available, so as to allow all related stakeholders to analyse and visualize results downstream of big data applications

⁷ A one-way hash function, also known as a message digest, fingerprint or compression function, is a mathematical function which takes a variable-length input string and converts it into a fixed-length binary sequence. Furthermore, a one-way hash function is designed in such a way that it is hard to reverse the process, that is, to find a string that hashes to a given value (hence the name one-way).

and generate new knowledge and insights. Providing analytics of this sort can elevate them to VAS status, as premium features TheFSM will support.

This layer will implement an API regarding the aforementioned services, aiming to provide a common, unified point of data exchange and services execution of TheFSM Data Platform. External applications, developers, end-users will be able to access, consume, exchange data and retrieve analytics.

FOODAKAI 2.0 and Agrivi 2.0 will make use of these data services in order to extend their services.

3.3.8. TheFSM Extended Applications

FOODAKAI 2.0

The FOODAKAI 2.0 application will be developed as an extension of the FOODAKAI platform and will focus on the risk prediction services for supplier assessment (verification). FOODAKAI 2.0 is based on FOODAKAI the Food Safety Data Incidents Platform (FSI Data Platform) developed by Agroknow. The Data Platform is currently used to collect, process, index (for searching) and publish the food safety incidents data. The platform includes the following components:

A data aggregation and processing methodology

A data aggregation and processing software Specifically the aggregation software includes amongst other features

- a. **A data collection component** used to collect dynamically data from several official data sources
- b. **A data processing component that transforms** the information to an internal rich format
- c. **A data enrichment component** that adds missing terms for hazards and products using the textual information of a food safety incident
- d. **A data curation environment** that allow curators to review, organise and enrich the data
- e. **A data indexing and publishing component** (Data Services) that is used by the front end applications to get the data.

A dataset that is continuously updated with new food safety incidents (recalls, food import rejections, alerts) and that includes also Companies and Product Brand information.

The relevant architecture of the FSI Data Platform is provided below:

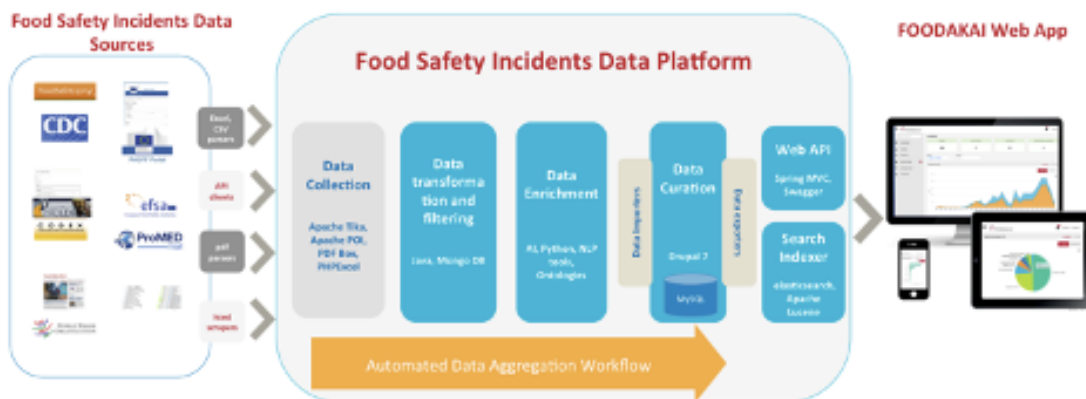


Figure 34: FSI Data Platform architecture

A machine learning API has been created for the enrichment of FOODAKAI's entities using machine learning techniques. This API uses 2 different models based on the annotation that will be attempted:

- one is using the title and textual description of the recall and is trained to identify the hazard which caused the recall, and
- the other one is also using the title and description of the recall and identifies products.

For both of them the SGDClassifier was used, along with a TFIDF vectorizer. This API has endpoints for the generation of the model with a new train dataset and for the identification of hazard and product terms, and is built using the Flask framework for Python. More details on the FSI Data Platform will be provided in D4.1.

The FOODAKAI 2.0 includes two main parts, a backend and a front end part. More specifically:

- The back end part includes mainly the intelligence API of the FSI Data Platform
- The front end part is developed using Ruby on Rails and ReactJS and includes all the functionalities that are provided to the end user for the supplier assessment. The application also includes a PostgreSQL database.

The FOODAKAI 2.0 application will integrate with the TheFSM Platform to securely identify and exchange information relevant to the supplier that is needed for the assessment and prediction of supplier's risk. Currently only data about the food safety incidents are used for the assessment of the supplier. The goal is to get additional information about the supplier, such as inspection results and reports, laboratory test results and certificates in order to create a risk matrix for each supplier that will help in remote verification activities.

FoodInspector

The main goal of this application is to transform the current food safety inspection process that involves paperwork and exchange of files, to a fully digital process with data assets that the inspector can click upon, interact with, and use to get prepared for an audit. The application will provide services that will allow inspectors to dig deeper into data slices and combinations - such as particular ingredients and products in a time period of interest. By having a way to dynamically dig deeper into the results of the inspection audits and the lab testing results, inspectors will be able to perform more accurate and fast assessment of all risk dimensions, in order to select the critical control point that should be of higher priority for a physical inspection. This means that the Certification Body can save time and money from unnecessary inspections and the food producer or manufacturer can avoid redundant product, device and lab testing activities.

The application will include a front end part that will integrate to the front end of FOODAKAI platform and a back end part that will rely on the data exchange services of TheFSM platform. More specifically:

- The front end will be developed using Ruby on rails and ReactJS. It will implement all the functionalities that the inspectors of a Certification Body need in order to assess the risks of a company that they will audit.
- The backend will include a software module that will invoke the APIs of the TheFSM platform, to retrieve information about a food company and inspections' outcomes. The information will include laboratory tests, previous audits, ingredients, traceability data etc.

Agrivi 2.0

Agrivi 2.0 application is an external farm management system intended primarily for producers (farmers) and food processors that are buying the produce from producers (farmers) in the context of supplier risk assessment. Agrivi 2.0 consists of:

- Agrivi FMS platform – existing farm management platform
- Extensions for theFSM project – specific add-ons to the farm management platform that will be developed during the project focused on additional attributes and reports specific for Global G.A.P. certification process

Agrivi 2.0 farm management system will serve primarily to producers (farmers) and food processors in terms of exchanging sourced produce traceability information and applied growing practices in terms of quality standards expected by the food processors.

Goal of the farm management solution for food processing companies is to ensure complete transparency and traceability of sourced produce from farmers and to tackle key challenges faced, such as: quantity volatility from farmer to farmer, applied farming practices that affect quality requirements, lack of produce traceability throughout the vegetation season and lack of efficient collaboration with farmers.

Agrivi 2.0 will also serve consultants and Global G.A.P. certified auditors to perform a virtual and efficient auditing process for farmers based on the farm data available in reports within the farm

management system of the farmer. Agrivi architecture and its components is presented in the diagram, below:

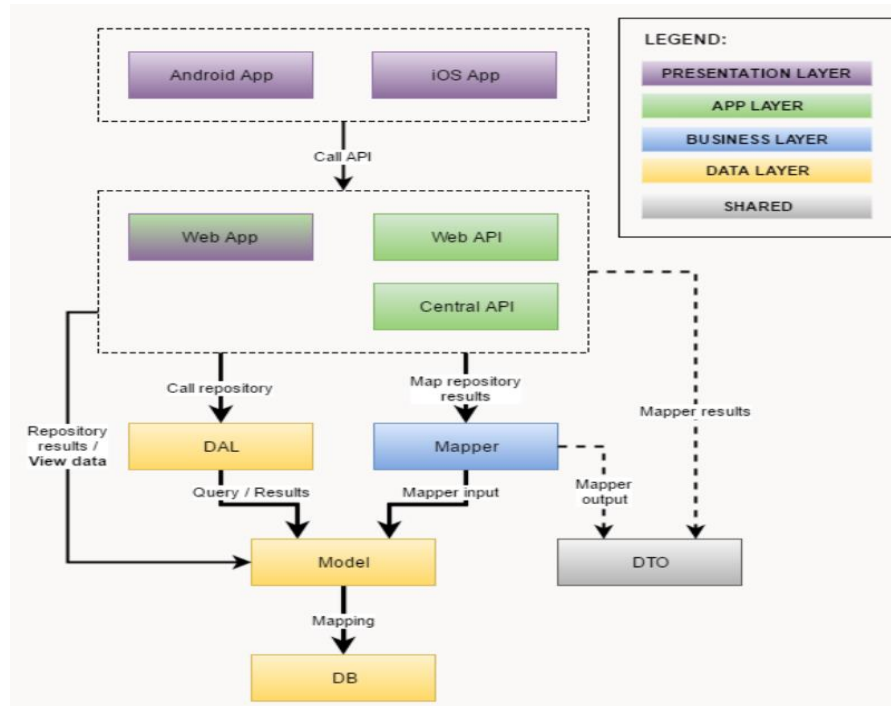


Figure 35: AGRIVI's high-level architecture

AGRIVI's interaction and integration points with TheFSM will be implemented through an API.

4 THEFSM TECHNICAL ARCHITECTURE

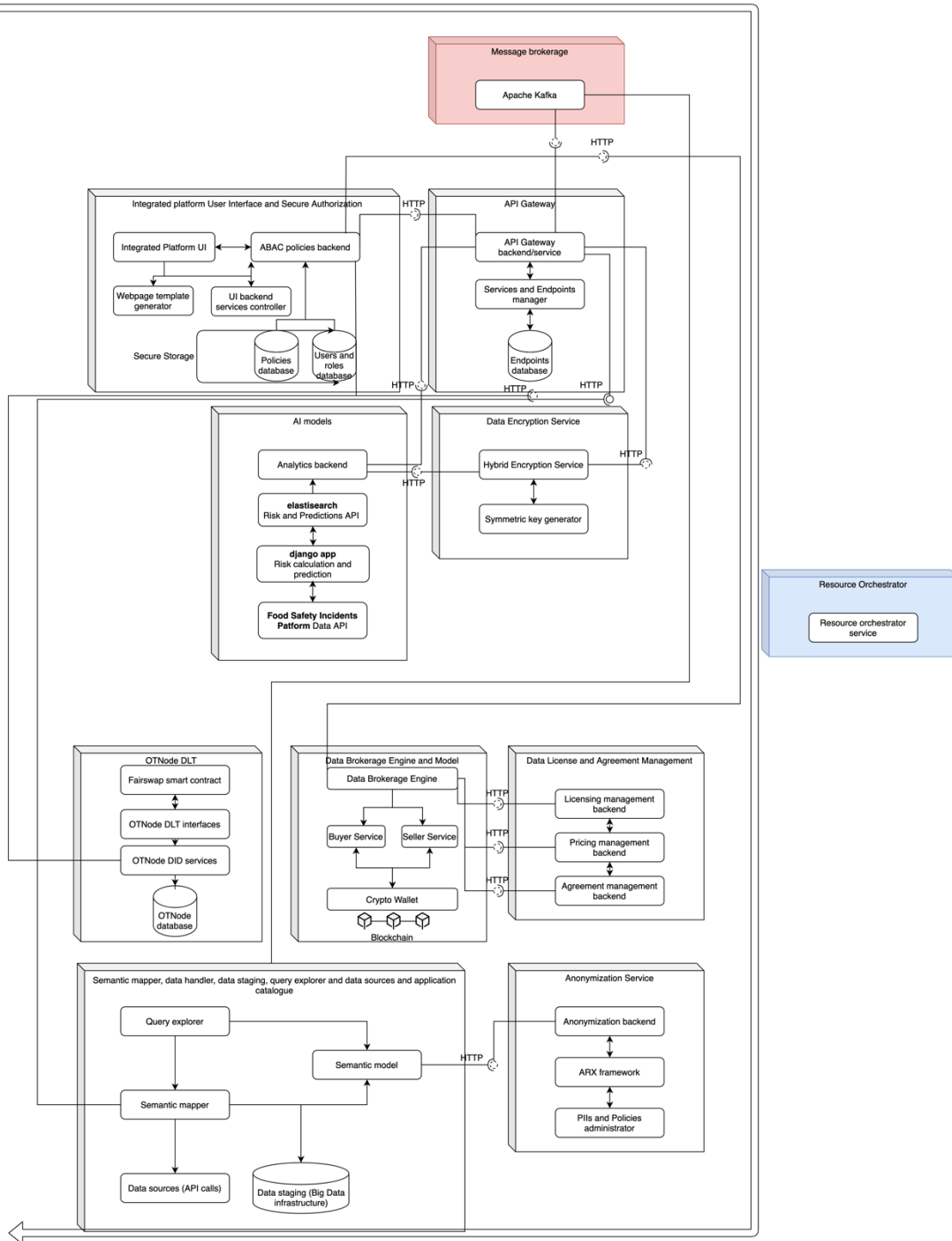


Figure 36: TheFSM technical architecture

The above figure illustrates the technical architecture of the platform, which draws many parallels, when compared to the conceptual architecture. The main difference lies in the fact that the technical architecture delves deeper and exposes some of the internal sub-components which

work in unison to meet the needs of the required workflows. A small description of the sub-components, per main component, as well as the technological stack used is as follows:

- **Message brokerage:** A component which works under the hood. It is responsible for forwarding notifications to required users, depending on settings and nature of data updates. It's built on Apache Kafka, while we are considering better alternatives, should we see it does not meet the project's needs.
- **API Gateway:** The component corresponding to the conceptual component of the API gateway, responsible for the protection of API calling, especially APIs provided by third party platforms integrated into TheFSM. It's built with MySQL as the backend database and Spring Boot, Spring Rest and Spring Security. This component consists of the following sub-components:
 - **API gateway backend/service:** The interface the API gateway exposes.
 - **Services and endpoints manager:** The sub-component responsible for managing all the APIs the gateway handles.
 - **Endpoints database:** A database with metadata and necessary information for the API gateway to properly function.
- **Data encryption service:** The service which encrypts data for transit, both from users to the platform and vice versa. It consists of a Javascript side and a Java side with Spring Security, both of which are needed to encrypt/decrypt on the client and/or the server side. This component consists of the following sub-components:
 - **Hybrid encryption service:** The sub-component responsible for handling hybrid encryption, yielding an encrypted object.
 - **Symmetric key generator:** The sub-component responsible for generating symmetric keys for one-time use, necessary for the hybrid encryption.
- **AI models:** The machine learning models utilized to generate analytics. It's built using Django, state-of-the-art Python libraries for deep learning such as Keras, Tensorflow and Prophet (for time series forecasting), while the data is stored using MongoDB and elasticsearch, in order to scale and handle big data. This component consists of the following sub-components:
 - **Analytics backend:** This subcomponent represents the entire backend infrastructure necessary for the machine learning models to train, run, generate analytics. This component is also responsible for the update of the prediction models on a regular basis.
 - **Big Data database:** Since some of the analytics, especially ones forecasting time series etc have the potential of generating big amounts of data, a big data database is utilized for this very reason.
 - **Intelligence APIs:** This subcomponent will make available the risk estimation and the risk prediction services through a set of web APIs
- **Integrated platform User Interface and Secure Authorization:** The component enforcing ABAC policies which restrict access to everything unless otherwise authorized and also the component providing the main user interface of the platform. The user interface is

oriented towards both users and administrators alike. It's built on Spring Boot, Spring MVC, Spring REST, Spring Security, JCasbin (a Java library for ABAC policies), while its backend databases (for the policies and for the user and role management) are MySQL databases (the data they are holding is limited and therefore not subject to big data requirements). It consists of the following sub-components:

- Integrated platform UI: The user interface of the platform.
- ABAC policies backend: The ABAC policies engine.
- Webpage template generator: The sub-component generating dynamic web-pages based on user-oriented content and queried results. The user interface is designed with performance in mind, providing an almost native-like experience to the user.
- UI backend services controller: The interface responsible for managing the API gateway interfaces via the UI.
- Policies database: Part of the secure storage. This database is responsible for storing all policies currently enforced by the system at any given time. It also stores policies which are temporarily deactivated by an administrator or moderator. Part of the secure storage.
- Users and roles database: This database is responsible for storing all users, their passwords (encrypted for extra security) and their corresponding roles. Part of the secure storage.
- Data flow management: A component which works under the hood. It is responsible for the proper error handling and calling of necessary API calls per workflow. Intended for internal use between the platform's components.
- Data licence and agreement management: This component is responsible for the IPR management, ensuring data monetization takes into consideration legal constraints etc. It's built on Spring boot and Spring REST. It consists of the following sub-components:
 - Licencing management backend: This sub-component manages the legal templates representing the licencing options a user has when trying to monetize their data.
 - Pricing management backend: This sub-component communicates with the data brokerage and engine model in order to do a fair transaction and transfer the paid money from buyer to seller.
 - Agreement management backend: This sub-component is responsible for ensuring monetization transactions were successful, reverting the process and notifying the relevant parties otherwise.
- Data brokerage and engine model: This component is responsible for handling the monetization and transaction procedures between buyer and seller. It's built on a NodeJS and Javascript backend interface. It consists of the following sub-components:
 - Data brokerage engine: This is the core of the backend, exposing the transaction handling.
 - Buyer service: Handling of the transaction from the buyer side.
 - Seller service: Handling of the transaction from the seller side.

- Crypto wallet: The interaction of smart wallets with the DLT itself.
- OTNode DLT: This component is responsible for the generation and provisioning of DIDs, as well as the handling of smart-wallets per user of the platform. It's built on a NodeJS and Javascript backend, while it mainly uses Ethereum as the DLT of choice. It consists of the following sub-components:
 - Fairswap smart contract: This is the interface enforcing the Fairswap protocol to ensure a fair and honest transaction between peers.
 - OTNode DLT interfaces: This is the main interface exposing APIs involving DIDs. One of the more central services this interface exposes is the authentication of users and JWT generation.
 - OTNode DLT services: The corresponding backend services exposed by the interface.
- Anonymization service: This component is responsible for anonymizing sensitive data, as part of the data curation pipelines. It's built with Spring Boot, Spring REST and ARX framework, a Java library for data anonymization. It consists of the following sub-components:
 - Anonymization backend: The interface exposing the service.
 - ARX framework: The backend backbone handling the anonymization procedures.
 - PII's and Policies administrator: The sub-component responsible for correlating anonymized segments with their real values.
- Semantic mapper, data handler, data staging, query explorer and data sources and application catalogue: This component consists of all the data curation, data processing and query exploration functionalities. It's built on Ontotext Platform GraphQL Playground for querying, Semantic Object Model Language schema for semantic modelling, Elasticsearch for fast indexing and search capabilities and MongoDB as the de facto data storage for scalability, It consists of the following sub-components:
 - Query explorer: The sub-component responsible for processing exploration queries and filtering results based on enforced policies.
 - Semantic model: The semantic model representing all entities and their semantic correlations. Utilized by SAI's backend in general.
 - Semantic mapper: The service mapping column information of datasets to be curated to semantic entities.
 - Data sources (API calls): Exposition of semantically enriched data.
 - Data staging (Big Data infrastructure): The actual database the semantically enriched data are stored. Due to semantic data being very verbose, large amounts of data are written and, consequently, the database is oriented towards handling of big data and scaling.
- Resource orchestrator: A component which works under the hood. It is responsible for properly handling the resource management when scaling and performance bottlenecks appear. It's built on Kubernetes.

5 CONCLUSIONS

The purpose of the current deliverable was to deliver a documentation of the second iteration of the platform for M15, while also reporting in an efficient way any changes and updates, when compare to the first iteration. This time, we delve into more detail as well, since we present the technical architecture of the platform alongside the updated conceptual.

At first, we present the methodology followed for the second iteration, documenting in detail all processes, instruments, roles and methods adopted throughout all development phases. Within this methodology, we clearly define the new features, functionalities and workflows we worked on. Also, the requirements definition with respect to key characteristics involving these workflows was presented, alongside the relevant stakeholders and their interactions with the platform.

As seen in the Annex sections, we provide the updated functional, non-functional and technical requirements, while we also provide a missing mapping from D3.1 of the first iteration, mapping the technical requirements to user stories. The new requirements for all types were added by taking into consideration not only the requirements of the pilots and the currently developed workflows, but also general principles modern state-of-the-art data markets should embrace, both for compatibility and for increasing the added value of the project.

The core of this deliverable is the presentation of the updated conceptual architecture, sequence and activity diagrams describing in more detail the current workflows, component interactivity and so on. Each old component is presented once again for the sake of completeness, while any updates are also reported in the corresponding sections. New components are introduced, their necessity is justified and their interoperability with the existing components is also described.

As development of the platform continues, we are planning to augment the data exchange and monetization scenarios so that even external platforms can interact with TheFSM and transact directly. This will help the platform to become more appealing and easier to adopt for potential interested parties. Generally speaking, we endeavor to offer **two ways of data sharing**: regular **data exchange** (subject to ABAC policies) and **data monetization** (subject to ABAC policies, but with monetized data as well). We are working towards expanding them in such a way that even external entities can conduct transactions with TheFSM platform. We also endeavor to incorporate the **IPR management** in **data monetization** and **data trading** as much as possible.

It should be stressed at this point that the current deliverable presents the second version of TheFSM conceptual architecture, as well as the first update on requirement analysis we conducted during the first iteration. These outcomes will drive the implementation phase of TheFSM platform that will be performed within the context of WP2, WP3 and WP4. However, as the design of TheFSM architecture and the identification and analysis of the functional and non-functional requirements, as well as their translation into technical requirements, is a living iterative process that will last until M36, the third version of this deliverable will include updates on both the architecture and the components of the architecture based on the feedback received.

6 BIBLIOGRAPHY

- [1] Cohn, M. (2010). Agile Softwareentwicklung: mit Scrum zum Erfolg!. Pearson Deutschland GmbH
- [2] Ericson, C. A. (2015). Hazard analysis techniques for system safety. John Wiley & Sons.
- [3] "Non Functional Requirements" [Online] Available: <https://www.scaledagileframework.com/nonfunctional-requirements/> [Accessed:16-10-2020]
- [4] Johan ter Bekke (1992). Semantic Data Modeling. Prentice Hall.
- [5] Shashank Agrawal and Melissa Chase, FAME: Fast Attribute-Based Message Encryption., 2007
- [6] Kan Yang, Xiaohua Jia, Bo Zhang, and Ruitao Xie, DAC-MACS: Effective data access control for multiauthority cloud storage systems., 2013
- [7] Amit Sahai and Brent Waters, Fuzzy Identity-Based Encryption., 2005
- [8] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Wate, Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data., 2006
- [9] John Bethencourt, Amit Sahai, and Brent Waters, Ciphertext-Policy Attribute-Based Encryption., 2007
- [10] Vincent Hu et al., Guide to Attribute Based Access Control (ABAC) Definition and Considerations., 2014
- [11] Vincent Hu, D. Kuhn, and David Ferraiolo, Attribute-Based Access Control., 2015.

ANNEX I USER STORIES

The following table provides TheFSM user stories, as they are defined by the business requirements provided in D1.1.

ID	Business requirement no	Category	User Story		
			As a <user>	I want to	So that
US_1	BR_nr2_1	Analytics	Producer	Collect different data related to business characteristics and the final product	I can have a better view of products I am interested in / i can confirm (reveal) that my production process conforms with the certification requirements
US_2	BR_nr2_2	Analytics	Producer	Be able to manage and evaluate data from different heterogeneous sources	I can draw conclusions for analysis, legislative requirements, etc.
US_3	BR_nr2_3	Notifications	Producer	Be constantly updated about information shown to me	I can make valid decisions
US_4	BR_nr2_4	Logging	Producer	Record up to date data assets for exploitation	I can catalogue data which is of interest to me
US_5	BR_nr2_6	Certification	Producer	Be able to validate GLOBALGAP certificates	I can be certain about the credibility of the certificates I am viewing
US_6	BR_nr2_7	Logging	Producer	Archive agreed specifications on the delivered product	I can make cooperation easier
US_7	BR_nr2_8	Integration	Producer	Be able to interconnect with the food processor's recording system	I can view terms of critical product data, tracking per batch and certificates of conformity
US_8	BR_nr2_12	Profiling	All	Be able to personalize the data I will use in my daily operations	I can have a more personalized experience

US_9	BR_nr2_14, BR_nr5_10, BR_nr4_1, BR_nr4_5, BR_nr4_8, BR_nr4_12, BR_nr4_18, BR_nr4_21, BR_nr5_2	Data, Certification	Producer, Food processor, Industry, Inspector/auditor	Replace physical documents with a complete digital collection	I can organize daily work faster and more efficiently
US_10	BR_nr2_16, BR_nr4_11, BR_nr4_15	Decision making	All	Use real-time data	Reduce decision-making time
US_11	BR_nr2_20	Decision making	Food Processor	Be able to segregate critical control points data (regarding product safety) from functional control points data	I can better assess both
US_12	BR_nr2_21	Monitoring	Food Processor	Be constantly updated about information that directly or indirectly affects food safety	I can ensure better quality for the product
US_13	BR_nr2_22, BR_nr2_35	Data, Traceability, Certification	Food Processor, Certification Body	Be able to easily access aggregated data from various sources (e.g., suppliers)	I can assess compliance with food safety standards and with the requirements of certified schemes

US_14	BR_nr2_24	Monitoring	Food Process or	Be immediately notified about any non-conformity raised for the producer and their certified product	I can take proper action
US_15	BR_nr2_25	Certification	Food Process or	Be able to easily access valid info to operational licenses for actors I interact with, as well as info regarding the accreditation of different kind of labs	I can validate my working collaborators and verify the effectiveness of the FSMS
US_16	BR_nr2_27	Profiling	Food Process or	Be able to categorize, modify and transfer my data in a common point of protected and controlled access	I can ensure my data is safe
US_17	BR_nr2_33	Monitoring	Food Process or	Have the ability of finding new partnerships and cooperation, via accessing information relevant to the current market needs	I can produce products which will cover above needs
US_18	BR_nr2_36	Certification	Certification Body	Be able to access up-to-date data from different sources and access to new and amended legislation	I can ensure the transparency of the certification process
US_19	BR_nr2_37	Certification	Certification Body	Be able to use a representative sample of the processed data	I can evaluate compliance with product specifications

US_20	BR_nr2_39	Certification	Certification Body	Be able to collect needed documentation prior to the certification decision	The decision can be properly certified
US_21	BR_nr2_40	Certification	Certification Body	Have different methods of sending and receiving information	I can collect documentation during the certification process
US_22	BR_nr2_45, BR_nr4_2	Monitoring, Certification	Certification Body, Inspector/auditor	Have direct and official information on findings of the National Audit Authorities in certified Producers, Food processors and Retailers	I can consult this information for decision making
US_23	BR_nr2_46, BR_nr3_6, BR_nr3_14	Certification	Certification Body, Producer, Food processor	Be able to easily obtain evidence for the justification of compliance criteria for the actors I am supervising	I can ensure transparency
US_24	BR_nr2_48	Auditing, Certification	Certification Body	Be able to obtain on-demand immediate stakeholder profile in terms of certification history	I can easier analyze the audit risk and for control/validation
US_25	BR_nr2_50	Risk estimation, Auditing	Certification Body	Be able to use and re-examine previous customers' audit findings, grouped into certain categories	I can highlight areas of high risk for subsequent audits
US_26	BR_nr2_51, BR_nr3_5, BR_nr3_13,	Traceability	Retailer, Producer, Food processor, Distributor	Be able to access detailed information about final shelf product, as well as correlation with	I can maintain a robust traceability and be able to efficiently withdraw products, should the need arise

	BR_nr3_21		tor	critical factors	
US_27	BR_nr3_8, BR_nr3_16, BR_nr3_22, BR_nr3_24, BR_nr3_32, BR_nr2_53	Traceability	Retailer, Producer, Certification Body, Distributor, Food processor	Be able to access fully traced information	I can have transparency and ensure no unfair trade practices effect consumers
US_28	BR_nr2_55	Profiling	Retailer	Be able to present important data relevant to QA actions taken by my company	I can enhance customer's trust on my brand name
US_29	BR_nr1_1	Decision making	Retailer	Be able to access information regarding findings of the inspection of suppliers in the food chain	I can make better decisions based on evidence
US_30	BR_nr1_2	Certification	Retailer	Be able to access current status of food supply actors, as far as audit results of certify organizations are concerned	I can validate their credibility for cooperation
US_31	BR_nr1_3	Risk estimation	Retailer	Have access to innovative tools	I can have enhanced risk monitoring capabilities
US_32	BR_nr3_2, BR_nr3_10, BR_nr3_18	Certification	Producer, Food processor, Distributor	Be able to locate with precise criteria required certificates and seals of approval, as requested by a customer/ Be able to access detailed	I can speed up certification and validation

				information regarding certificate validity and scope of certification	
US_33	BR_nr3_3, BR_nr3_11, BR_nr3_19	Decision making	Producer, Food processor	Have a way to view an estimation of costs and expenditures regarding the certification process/Be able to choose appropriate certificate scheme	Have more information when considering/ I can meet the requirements of different organizations (retailers, distributor, processors)
US_34	BR_nr3_4, BR_nr3_12, BR_nr3_20, BR_nr3_30	Decision making, Auditing	Producer, Distributor, Certification Body, Food processor	Be able to support remote audits	I can reduce decision-making under difficult situations
US_35	BR_nr3_7, BR_nr3_15, BR_nr3_23, BR_nr3_31	Decision making, certification	Producer, Distributor, Certification Body, Food processor	Have access to validated data of all stakeholders	I can support decision-making processes
US_36	BR_nr3_21		Distributor	assess data	I Can conduct fact driven management
US_37	BR_nr3_26	Certification	Certification Body	Be able to understand the specific requirements of an organization	I can speed up the certification process without grey areas

US_38	BR_nr3_27	Certification	Certification Body	Be able to directly interact with organizations requesting certification	To speed up the certification process
US_39	BR_nr3_33	Certification, Auditing	Certification Body	Be able to have a better overall view of the ability of an audited organization	I can have better audit results
US_40	BR_nr3_35, BR_nr3_36	Risk estimation	Retailer	Reduce the number of product recalls	I can improve efficiency
US_41	BR_nr4_4, BR_nr4_7	Monitoring	Inspector/Auditor	Be able to interact with data of different Certification Bodies	The data has increased reliability
US_42	BR_nr4_5	Certification	Certification Committee	Be able to verify a digital report	
US_43	BR_nr4_9, BR_nr4_10, BR_nr4_13, BR_nr4_14, BR_nr4_16, BR_nr4_17, BR_nr4_19, BR_nr4_20, BR_nr4_22	Traceability	Farmer/Producer, Distributor	Be able to trace input suppliers	Ensure the quality of my product

US_44	BR_nr4_23		Public authorities	Be able to check and verify product data with respect to compliance with certification regulations	I can ensure transparency
US_45	BR_nr5_1	Monitoring	Public Authorities (NVWA)	Be able to predict when/what/where to check	I can ensure food safety and efficiency
US_46	BR_nr5_2	Monitoring	Public Authorities (NVWA)	Be able to have access to the digital format of the inspection	I can ensure efficiency
US_47	BR_nr5_3	Monitoring	Public Authorities (NVWA)	Be able to search past audit performance per actor (producer, supplier, etc.)	I can ensure food safety and inspection efficiency
US_48	BR_nr5_4	Risk estimation	Public Authorities (NVWA)	Be able to conduct risk-based monitoring	I can conduct efficient sampling
US_49	BR_nr5_8	Certification	Public authorities	Be able to assess the performance of the producers in complying to the certification standards	I can decide to what extent they comply with law and certification standards
US_50	BR_nr5_12	Monitoring	Industry	Be able to inspect market needs and new clients	I can better supervise the supply chain process
US_51	BR_nr5_13	Monitoring	Industry	Be able to establish an up to date communication channel with traders	I can ensure communication

Table 6: TheFSM user stories

ANNEX II USER STORIES – RANKING

ID	Vector value (end-user)	Vector value (technical)	Final ranking value
US_1	3	4	12
US_2	3	4	12
US_3	3	4	12
US_4	3	4	12
US_5	3	4	12
US_6	3	4	12
US_7	3	3	9
US_8	4	4	16
US_9	4	4	16
US_10	4	5	20
US_11	3	4	12
US_12	4	4	16
US_13	4	5	20
US_14	3	4	12
US_15	3	4	12
US_16	4	5	20
US_17	3	4	12
US_18	3	4	12
US_19	3	4	12
US_20	3	5	15
US_21	4	3	12
US_22	4	3	12
US_23	3	4	12
US_24	4	4	16
US_25	4	4	16
US_26	3	4	12
US_27	4	5	20
US_28	3	4	12
US_29	4	4	16
US_30	4	4	16
US_31	4	4	16
US_32	3	4	12
US_33	3	3	9
US_34	4	4	16

US_35	4	5	20
US_36	3	4	12
US_37	3	4	12
US_38	3	4	12
US_39	4	4	16
US_40	4	3	12
US_41	4	3	12
US_42	4	4	16
US_43	3	4	12
US_44	3	4	12
US_45	4	3	12
US_46	4	4	16
US_47	4	4	16
US_48	4	4	16
US_49	4	4	16
US_50	3	4	12
US_51	3	3	9

Table 7: User stories ranking and heatmap – 2nd iteration

ANNEX III THEFSM FUNCTIONAL REQUIREMENTS

Reference ID	Business scenario	Business Requirement ID	Title	Actors	Category
FR_1	nr_1	BR_nr1_2, BR_nr1_1	Create user account	All	User account
FR_2	nr_1	BR_nr1_2, BR_nr1_1	Create company profile	All	Company profile
FR_3	nr_1	BR_nr1_2, BR_nr1_1	Upload certification info	Producer	Certification validation
FR_4	nr_1	BR_nr1_2, BR_nr1_1	Upload laboratory analysis test info	Producer	Certification validation
FR_5	nr_1	BR_nr1_2	Estimate risk	Retailer (FSQA expert)	Risk assessment
FR_6	nr_1, nr_2	BR_nr1_2, BR_nr2_8, BR_nr2_9	Request an audit by third party	Retailer (FSQA expert), Producer	Alerting, Auditing
FR_7	nr_1, nr_2	BR_nr1_2, BR_nr2_8, BR_nr2_9	Receive audit request	Certification Body	Alerting, Auditing

FR_8	nr_1	BR_nr1_2	Submit audit results	Certification Body	Certification validation
FR_9	nr_1	BR_nr1_2, BR_nr1_1	Upload lab results	Lab expert	Certification validation
FR_10	nr_1	BR_nr1_2, BR_nr1_1	New audit notification	Retailer (FSQA expert)	Alerting, Auditing
FR_11	nr_1	BR_nr1_2, BR_nr1_1	New test results notification	Retailer (FSQA expert)	Alerting, Auditing
FR_12	nr_1	BR_nr1_2, BR_nr1_1	View traceability report for a cultivation	Retailer (FSQA expert)	Product details, Traceability
FR_13	nr_1	BR_nr1_2, BR_nr1_1	View IoT data (from farm, production) of a specific product	Retailer (FSQA expert)	Product details, Monitoring
FR_14	nr_1	BR_nr1_2, BR_nr1_1	Provide IoT data (from farm, production) of a specific product	Producer	Product details
FR_15	nr_1	BR_nr1_1	View findings of the inspection of suppliers in the food chain	Retailer (FSQA expert)	Product details
FR_16	nr_1	BR_nr1_2, BR_nr1_1	Select specific suppliers/professionals of interest to monitor their status	Retailer (FSQA expert)	Monitoring, Traceability
FR_17	nr_1	BR_nr1_2, BR_nr1_1	Show product history based on traceability unit id (LOT number)	Retailer (FSQA expert)	Product details
FR_18	nr_1	BR_nr1_2, BR_nr1_1	Predict an increasing risk for a supplier	Retailer (FSQA expert)	Risk assessment
FR_19	nr_1	BR_nr1_1	View specific certification for a producer	Supplier	Monitoring
FR_20	nr_1	BR_nr1_1	View traceability history for cultivation	Supplier	Product details, Traceability
FR_21	nr_1	BR_nr1_2, BR_nr1_1	View lab results and the certification of analysis for a specific producer	Supplier	Company profile, Traceability
FR_22	nr_1	BR_nr1_1	View information for a supplier (Name, products, location)	Supplier	Company profile
FR_23	nr_1	BR_nr1_2, BR_nr1_1	View information about audits and inspections for a specific producer/grower	Supplier	Auditing, Monitoring
FR_24	nr_1	BR_nr1_1	View information for the food recalls, border rejections and inspections for a specific supplier	Supplier	Monitoring
FR_25	nr_1	BR_nr1_2, BR_nr1_1	Access to Producer-entered data (production plans, progress, practices, risks, deliveries)	Supplier	Company profile

FR_26	nr_1	BR_nr1_2, BR_nr1_1	Access to Retail-entered data (production plans, progress, practices, risks, deliveries)	Supplier	Company profile
FR_27	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14	View real-time data related to cultivation conditions	Producer	Monitoring, Product details
FR_28	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14	View additional data related to cultivation conditions	Producer	Monitoring
FR_29	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14	View characteristics of plots (plots' distribution and their topographic features)	Producer	Monitoring
FR_30	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14	Collect data related to characteristics relevant to the agricultural plots (soil), to the plantation etc., before implementing agricultural practices	Producer	Monitoring
FR_31	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14	Extract insights about the plots status	Producer	Monitoring
FR_32	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14	Get notified about potential risks	Producer	Monitoring, Alerting
FR_33	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14	Provide/view available resources of producer's business	Producer	Monitoring, Company profile
FR_34	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14, BR_nr2_16	Share measurement of the concentration (residues) of Plant Protection Substances in the final	Producer	Monitoring, Certification validation
FR_35	nr_2	BR_nr2_1, BR_nr2_2, BR_nr2_3, BR_nr2_10, BR_nr2_14, BR_nr2_16	Share measurements of characteristics relevant to the agricultural plots (soil), to the plantation etc., before implementing agricultural practices	Producer	Monitoring, Certification validation

FR_36	nr_2	BR_nr2_8, BR_nr2_16	Share data from all correlation stages with the food processor	Producer	Monitoring, Traceability
FR_37	nr_2	BR_nr2_8, BR_nr2_16	Share certificate history of a specific product	Producer	File management, Certification validation
FR_38	nr_2	BR_nr2_8	Validate certificate from GLOBALGAP database	Producer	Certification validation
FR_39	nr_2	BR_nr2_9, BR_nr2_8, BR_nr2_7,	Support negotiation with food processor about the characteristics of the product	Producer	Traceability
FR_40	nr_2	BR_nr2_5, BR_nr2_9, BR_nr2_8,	Share directly production data related to the traceability units with processor	Producer	Traceability
FR_41	nr_2	BR_nr2_6, BR_nr2_9, BR_nr2_8, BR_nr2_5	Share of the GlobalGAP Number (GGN) with the processor for certificate validation	Producer	Certification validation, Company profile
FR_42	nr_2	BR_nr2_6, BR_nr2_9, BR_nr2_8, BR_nr2_5	Correlate the certification with tracking batches	Producer	Traceability, Certification validation
FR_43	nr_2	BR_nr2_8, BR_nr2_9	Share production data with the Certification Body	Producer	Certification validation
FR_44	nr_2	BR_nr2_8, BR_nr2_9	View production data of producer	Certification Body	Monitoring
FR_45	nr_2	BR_nr2_8, BR_nr2_9	Support negotiation with certification body about the financial offer	Producer	Certification validation, Traceability
FR_46	nr_2	BR_nr2_8, BR_nr2_9	Receive an alert that my company is uploaded in the GLOBALGAP database	Producer	Certification validation, Alerting
FR_47	nr_2	BR_nr2_8, BR_nr2_9	Upload and manage audit data and files	Certification Body	Auditing, File management
FR_48	nr_2	BR_nr2_8, BR_nr2_9	Create audit report	Certification Body	Auditing
FR_49	nr_2	BR_nr2_8, BR_nr2_9	Share audit report with producer	Certification Body	Auditing, Authentication/Authorization
FR_50	nr_2	BR_nr2_8, BR_nr2_9	Upload final audit data in the GLOBALGAP database	Certification Body	Auditing
FR_51	nr_2	BR_nr2_8, BR_nr2_9	Issue a certification for producer	Certification Body	Certification validation
FR_52	nr_2	BR_nr2_21, BR_nr2_20, BR_nr2_28	View/access farm data on the traceability of a particular batch from producer	Food Processor	Monitoring, Traceability

FR_53	nr_2	BR_nr2_20, BR_nr2_32, BR_nr2_31, BR_nr2_28, BR_nr2_21	View/access product safety verification data from producer	Food Processor	Risk assessment, Traceability
FR_54	nr_2	BR_nr2_20, BR_nr2_32, BR_nr2_31, BR_nr2_28, BR_nr2_21	View/access food recall data from producer	Food Processor	Risk assessment, Traceability
FR_55	nr_2	BR_nr2_31, BR_nr2_24, BR_nr2_34, BR_nr2_33	View/access retailer's requirements Data from retailer	Food Processor	Product details, Risk assessment
FR_56	nr_2	BR_nr2_31, BR_nr2_24, BR_nr2_34, BR_nr2_33	Define product data requirements to food processor	Retailer	Product details, Risk assessment
FR_57	nr_2	BR_nr2_31, BR_nr2_24, BR_nr2_34, BR_nr2_33	Provide feedback about a specific product to food processor	Retailer	Product details, Risk assessment
FR_58	nr_2	BR_nr2_31, BR_nr2_24, BR_nr2_34, BR_nr2_33	View feedback about a specific product from retailer	Food Processor	Product details, Risk assessment
FR_59	nr_2	BR_nr2_31, BR_nr2_24, BR_nr2_34, BR_nr2_33	Access to certificates generated in previous phases	Food Processor	Certification validation, Product details
FR_60	nr_2	BR_nr2_31, BR_nr2_24, BR_nr2_34, BR_nr2_33	View suppliers who hold a specific certificate	Food Processor	Company profile
FR_61	nr_2	BR_nr2_32, BR_nr2_20	View food recall data and receive alerts	Food Processor	Monitoring, Alerting
FR_62	nr_2	BR_nr2_32, BR_nr2_20	View supplementary product data	Food Processor	Monitoring, Product details
FR_63	nr_2	BR_nr2_31, BR_nr2_24, BR_nr2_34, BR_nr2_33	Share raw materials specifications to producer	Food Processor	Product details, Negotiation support
FR_64	nr_2	BR_nr2_31, BR_nr2_24, BR_nr2_34, BR_nr2_33	Share final product safety verification data to retailer	Food Processor	Product details, Certification validation

FR_65	nr_2	BR_nr2_22, BR_nr2_21, BR_nr2_39, BR_nr2_36, BR_nr2_30, BR_nr2_46	Share compliance Data, which evidently reveal conformity of the packaging process against the requirements of the food safety standards (FSSC22000, IFS) to the Certification Body	Food Processor	Product details, Certification validation
FR_66	nr_2	BR_nr2_22, BR_nr2_21, BR_nr2_39, BR_nr2_36, BR_nr2_30, BR_nr2_46	Share certification data with the Certification Body	Food Processor	Product details, Certification validation
FR_67	nr_2	BR_nr2_21, BR_nr2_20	To monitor data from all stages of food processing	Food Processor	Product details, Monitoring
FR_68	nr_2	BR_nr2_35, BR_nr2_36, BR_nr2_37, BR_nr2_38, BR_nr2_39	To receive/view/access data from producer, processor, retailer in order to check if the stakeholder complies with the certification schemes	Certification Body	Certification validation, Monitoring
FR_69	nr_2	BR_nr2_35, BR_nr2_36, BR_nr2_37, BR_nr2_38, BR_nr2_39	To receive/view/access laboratory analysis reports	Certification Body	Certification validation, Traceability
FR_70	nr_2	BR_nr2_6, BR_nr2_9, BR_nr2_8, BR_nr2_5	Share of the GlobalGAP Number (GGN) and relevant valid certificate	Producer	Certification validation, Company profile
FR_71	nr_2	BR_nr2_6, BR_nr2_9, BR_nr2_8, BR_nr2_5	Have access on a the FSSC certification checklist	Certification Body	Certification validation
FR_72	nr_2	BR_nr2_38, BR_nr2_39, BR_nr2_46	Share my evaluation data from the National Accreditation Council	Certification Body	Data management
FR_73	nr_3	BR_nr3_1, BR_nr3_3	Provide relevant data to the planned comprehensive database (FSM)	Producer, Food Processor, Distributor	Certification validation, Data management
FR_74	nr_3	BR_nr3_1, BR_nr3_2, BR_nr3_3	Retailer provides specific requirements regarding certification and seals of approval	Retailer	Certification validation
FR_75	nr_3	BR_nr3_1, BR_nr3_2, BR_nr3_3, BR_nr3_7	CB provides detailed information regarding standards, seals of approval and certification process	Certification Body	Certification validation, Alerting
FR_76	nr_3	BR_nr3_1, BR_nr3_3, BR_nr3_5	Data exchange between orgs	All	Data management, Alerting

FR_77	nr_3	BR_nr3_1, BR_nr3_2, BR_nr3_3, BR_nr3_6	Find all necessary information for planning and realizing the certification process	Certification Body	Certification validation, Data management
FR_78	nr_3	BR_nr3_1, BR_nr3_2, BR_nr3_6	Provide specific customer requirements, standards, seals of approval to producers	Producer	Certification validation
FR_79	nr_3	BR_nr3_1, BR_nr3_3, BR_nr3_6	Determine which standards and seals of approval should be implemented to meet all customer requirements to raise competitiveness	Producer	Certification validation
FR_80	nr_3	BR_nr3_1, BR_nr3_3, BR_nr3_6	Notify producer on prerequisites to fulfill the specific requirements and which resources must be allocated	Producer	Certification validation, Alerting
FR_81	nr_3	BR_nr3_1, BR_nr3_6	Provide a way for producers to determine which certification body can certify which elected standards	Producer	Certification validation, Alerting
FR_82	nr_3	BR_nr3_1, BR_nr3_6	Manage high amounts of data, from heterogenous sources	All	Data management
FR_83	nr_3	BR_nr3_1, BR_nr3_6	Regular data exchange between producer and food processor	Producer, food processor	Data management, Alerting
FR_84	nr_3	BR_nr3_1, BR_nr3_6	Regular data exchange between producer and retailer	Producer, retailer	Data management, Alerting
FR_85	nr_3	BR_nr3_1, BR_nr3_6	Presale data of producer and company	Producer	Company profile, Certification validation
FR_86	nr_3	BR_nr3_1, BR_nr3_6	Search and request certification from distributor	Producer	Data management, Alerting
FR_87	nr_3	BR_nr3_1	Request certification data from certification body	Producer	Certification validation, Data management
FR_88	nr_3	BR_nr3_1, BR_nr3_6	Producer send requirements to food processor, or food processor directly contacting producer	Food Processor, producer	Certification validation, Alerting
FR_89	nr_3	BR_nr3_1, BR_nr3_6	Producer send requirements to retailer, or retailer directly contacting producer	Retailer, producer	Certification validation, Alerting
FR_90	nr_3	BR_nr3_1, BR_nr3_6	Regular publication of distributor certificates	Distributor	Certification validation, Data management
FR_91	nr_3	BR_nr3_1, BR_nr3_4, BR_nr3_6	CB provide audit reports to producer	Producer	Certification validation, Auditing
FR_92	nr_3	BR_nr3_1, BR_nr3_4, BR_nr3_6	Allow consultants to collect data, conduct internal audits, for decision support	Consultant	Certification validation, Auditing
FR_93	nr_3	BR_nr3_1	Enable labs to post certificates after lab tests	Lab expert	Certification validation, Data management

FR_94	nr_3	BR_nr3_1, BR_nr3_6, BR_nr3_8	Allow public authorities to extract information, ensure the legality of the procedures, as also utilize traceability data and analytics	Public Authorities	Monitoring, Traceability
FR_95	nr_3	BR_nr3_9, BR_nr3_14	Regular data exchange between food processor and food processor	Food processor (x2)	Data management, Alerting
FR_96	nr_3	BR_nr3_9, BR_nr3_14	Regular data exchange between food processor and retailer	Food processor, retailer	Data management, Alerting
FR_97	nr_3	BR_nr3_9, BR_nr3_14	Presale data of distributor and company	Food processor	Data management, Alerting
FR_98	nr_3	BR_nr3_9	Request certification data from certification body	Food processor	Certification validation, Data management
FR_99			Food processor send requirements to retailer, or retailer directly contacting food processor	Food processor, distributor	Certification validation, Alerting
FR_100	nr_3	BR_nr3_9, BR_nr3_14	Search and request certification from distributor	Food processor	Data management, Alerting
FR_101	nr_3	BR_nr3_9, BR_nr3_12, BR_nr3_14	CB provide audit reports to food processor	Food processor	Data management, Alerting
FR_102	nr_3		Distributor send requirements to retailer, or retailer directly contacting distributor	Retailer, distributor	Certification validation, Alerting
FR_103	nr_3	BR_nr3_25, BR_nr3_26, BR_nr3_28, BR_nr3_29, BR_nr3_30	Provide impartial certification processes, procedures and practices	Certification Body	Certification validation, Data management
FR_104	nr_3	BR_nr3_25, BR_nr3_28, BR_nr3_29, BR_nr3_30	Provide competent audits by certification scheme owners	Certification Body	Certification validation, Data management
FR_105	nr_3	BR_nr3_25, BR_nr3_26, BR_nr3_28, BR_nr3_29, BR_nr3_30	Support independent decision-making on certification issuing	Certification Body	Certification validation, Data management
FR_106	nr_3	BR_nr3_25, BR_nr3_26	Provide certification data, requirements, standards, to: producer, food processor, distributor	Certification Body	Certification validation, Data management
FR_107	nr_3	BR_nr3_25, BR_nr3_28, BR_nr3_33	Interact with consultants	Certification Body	Certification validation, Auditing
FR_108	nr_3	BR_nr3_25, BR_nr3_26, BR_nr3_27	Interact with companies	Certification Body	Certification validation, Data management

FR_109	nr_3	BR_nr3_34, BR_nr3_36	Search and request certification from distributor	Retailer	Data management, Alerting
FR_110	nr_3	BR_nr3_34, BR_nr3_36	Producer send requirements to retailer, or retailer directly contacting producer	Retailer, producer	Certification validation, Alerting
FR_111	nr_3	BR_nr3_34, BR_nr3_36	Interact with certification body to obtain audit reports, certificates and seals of approval	Retailer, Certification Body	Certification validation, Alerting
FR_112	nr_3	BR_nr3_36	Provide traceability data to consumers	Retailer	Certification validation, Traceability
FR_113	nr_3	BR_nr3_34, BR_nr3_36	Receive regular updates on certifications and product specifications	Retailer	Certification validation, Alerting
FR_114	nr_3	BR_nr3_34, BR_nr3_36	Cooperate with consultants for audits	Retailer	Certification validation, Auditing
FR_115	nr_3	BR_nr3_35	(optional) provide detailed risk analysis for products	Retailer	Risk assessment, Data analysis
FR_116	nr_3	BR_nr3_34, BR_nr3_36	Provide samples to labs for testing	Retailer	Certification validation, Data management
FR_117	nr_4	BR_nr4_1, BR_nr4_2	Make available the auditing reports and/or non-compliances found	Public Authorities	Auditing, Data management
FR_118	nr_4	BR_nr4_1, BR_nr4_2, BR_nr4_3, BR_nr4_4	Integration with national DBs like SIAN (it is a registry), to allow wine cellars to collect data about every wine movement	Public Authorities, Producer	Monitoring, Product details
FR_119	nr_4	BR_nr4_1, BR_nr4_2	Enable auditors to fill in reports in digital form	Inspector/Auditor	Auditing, Data management
FR_120	nr_4	BR_nr4_1, BR_nr4_2, BR_nr4_4	Integrate legal requirements, lab certifications, specific parameters for auditor	Inspector/Auditor	Data management
FR_121	nr_4	BR_nr4_7	Evaluate inspector reports	Certification Body	Data management
FR_122	nr_4	BR_nr4_6	Issue certification	Certification Body	Authentication/Authorization, Certification validation
FR_123	nr_4	BR_nr4_6	Send digital certification to winegrower/winemaker/bottler (producer?)	Certification Body	Certification validation, Alerting
FR_124	nr_4	BR_nr4_7	Enable communication between Certification Body and inspector if doubting information	Certification Body, Inspector	Alerting
FR_125	nr_4	BR_nr4_6	Issue measure of non-compliance/irregularity if Operator (producer?) doesn't meet requirements	Certification Body	Certification validation, Alerting
FR_126	nr_4	BR_nr4_7	Periodically check traceability and status of products by checking inspection reports	Certification Body	Monitoring, Traceability

FR_12_7	nr_4	BR_nr4_5	Verify integrity of digital report	Certification Body	Data management
FR_12_8	nr_4	BR_nr4_8	Notify producers about regulations to be fulfilled	Producer	Alerting, Monitoring
FR_12_9	nr_4	BR_nr4_8	Ensure producer has up-to-date status on buying and selling, farm files	Producer	Alerting, Monitoring
FR_13_0	nr_4	BR_nr4_8	Interaction with auditors for farm inspection	Producer	Auditing, Monitoring
FR_13_1	nr_4	BR_nr4_8	Provide timelines, products or techniques suggestions, certification information	Producer	Certification validation, Data management
FR_13_2	nr_4	BR_nr4_8	Update vineyard info if not in touch with consultant	Producer	Certification validation, Data management
FR_13_3	nr_4	BR_nr4_19, BR_nr4_21	Receive producer's data about harvest period of certified product, certification validity and report	Bottler	Traceability, Certification validation
FR_13_4	nr_4	BR_nr4_19, BR_nr4_20, BR_nr4_21	Reach retailers to communicate the value of the certified product.	Bottler	Traceability, Alerting
FR_13_5	nr_4	BR_nr4_22	Receive certificate from producer	Supplier	Certification validation, Traceability
FR_13_6	nr_4	BR_nr4_22	User certification data for marketing purposes	Supplier	Certification validation, Traceability
FR_13_7	nr_4	BR_nr4_23, BR_nr4_24	Populate official databases with farm data and food health data from operators, as well as their certifications	Public authorities	Data management, Data management
FR_13_8	nr_4	BR_nr4_23, BR_nr4_24	Provide remote control, monitoring and traceability capabilities	Public authorities	Monitoring, Traceability
FR_13_9	nr_5	BR_nr5_2, BR_nr5_3, BR_nr5_5	Collect data, inspections on-site at all actors, issue results and upload to db	Public authorities (NVWA)	Data management, Data analysis
FR_14_0	nr_5	BR_nr5_2, BR_nr5_4, BR_nr5_6	Schedule inspections	Public authorities (NVWA)	Auditing, Monitoring
FR_14_1	nr_5	BR_nr5_2, BR_nr5_4, BR_nr5_6	Notify concerned actors about upcoming inspection/auditing	Public authorities (NVWA)	Alerting, Monitoring
FR_14_2	nr_5	BR_nr5_2, BR_nr5_3, BR_nr5_4	Ability to evaluate inspectors against EU regulations (such as (EG) nr. 178/2002)	Public authorities (NVWA)	Monitoring, Alerting
FR_14_3	nr_5	BR_nr5_2, BR_nr5_3, BR_nr5_4	Get product analysis to verify compliance towards certifications	Public authorities (NVWA)	Certification validation, Monitoring
FR_14_4	nr_5	BR_nr5_2, BR_nr5_3, BR_nr5_4, BR_nr5_6	Communicate complaints/accusations for certified producers or processors	Public authorities (NVWA)	Certification validation, Monitoring

FR_14 5	nr_5	BR_nr5_7, BR_nr5_8, BR_nr5_9, BR_nr5_10	View reports of inspection, audits reports, combined data collected from all actors in the supply chain such as declared volumes/quantities/prices	Producer, Industry	Data management, Monitoring
FR_14 6	nr_5	BR_nr5_7, BR_nr5_9, BR_nr5_10	Fill forms for inspection, report volumes, prices and food safety results to authorities, exchange reports with the certification bodies	Producer, Industry	Data management, Certification validation
FR_14 7	nr_5	BR_nr5_10, BR_nr5_12, BR_nr5_13	Trade certified processed products in the market	Industry	Certification validation, Negotiation support
FR_14 8	nr_5	BR_nr5_10, BR_nr5_12	Accept inspections by food authorities and certification bodies for compliance	Industry	Auditing, Monitoring
FR_14 9	nr_5	BR_nr5_10, BR_nr5_12	Set product specifications for products it sends to the market to retailers	Industry	Data management
FR_15 0	nr_5	BR_nr5_10	Contact consultants for audits, guide implementation of best practices against certification standards	Industry	Certification validation, Data analysis

Table 8: Functional requirements – 2nd iteration

The table above displays the updated functional requirements, when compared to the first iteration of the platform and the corresponding D3.1. As shown, some requirements have been removed. More specifically, the following requirements were removed:

FR_13 - View IoT data (from farm, production) of a specific product: Removed due to not handling IoT data in the project.

FR_14 - Provide IoT data (from farm, production) of a specific product: Removed due to not handling IoT data in the project.

FR_45 - Support negotiation with certification body about the financial offer: Not needed in any of the current workflows.

FR_46 - Receive an alert that my company is uploaded in the GLOBALGAP database: Not needed by the current workflows.

FR_73 - Provide relevant data to the planned comprehensive database (FSM): Due to updates on development, the data management processes have become a bit more complex due to semantic enrichment and the exchange scenarios.

FR_76 - Data exchange between orgs: Data exchange is under development between third-party platforms which are part of the ecosystem of TheFSM. Until the third iteration of the platform, we are exploring the potential of expanding this interactivity for added value.

FR_78 - Provide specific customer requirements, standards, seals of approval to producers: Not needed in any of the current workflows.

FR_79 - Determine which standards and seals of approval should be implemented to meet all customer requirements to raise competitiveness: Not needed in any of the current workflows.

FR_82 - Manage high amounts of data, from heterogeneous sources: During communication with the third party platforms participating in the project, it was determined that sources, while heterogeneous, do not require the storage of high amounts of data per upload, although we still take scaling into consideration during development.

FR_83 - Regular data exchange between producer and food processor: Not needed in any of the current workflows.

FR_84 - Regular data exchange between producer and retailer: Not needed in any of the current workflows.

FR_85 - Presale data of producer and company: Not needed in any of the current workflows.

FR_86 - Search and request certification from distributor: Not needed in any of the current workflows.

FR_87 - Request certification data from certification body: Not needed in any of the current workflows.

FR_95 - Regular data exchange between food processor and food processor: Not needed in any of the current workflows.

FR_96 - Regular data exchange between food processor and retailer: Not needed in any of the current workflows.

FR_97 - Presale data of distributor and company: Not needed in any of the current workflows.

FR_98 - Request certification data from certification body: Not needed in any of the current workflows.

FR_99 - Food processor sends requirements to retailer, or retailer directly contacting food processor: Not needed in any of the current workflows.

FR_100 - Search and request certification from distributor: Not needed in any of the current workflows.

FR_101 - CB provides audit reports to food processors: Not needed in any of the current workflows.

FR_102 - Distributor sends requirements to retailer, or retailer directly contacting distributor: Not needed in any of the current workflows.

FR_107 - Interact with consultants: Not needed in any of the current workflows.

FR_108 - Interact with companies: Will be removed for now, but will be explored during development till the third iteration.

FR_112 - Provide traceability data to consumers: Not needed in any of the current workflows.

FR_114 - Cooperate with consultants for audits: Not needed in any of the current workflows.

FR_119 - Enable auditors to fill in reports in digital form: These do not need to be filled by the platform, but will be provided by the third-party platforms in dataset form during upload.

FR_126 - Periodically check traceability and status of products by checking inspection reports: Not needed in any of the current workflows.

FR_129 - Ensure producer has up to date status on buying and selling farm files: Not needed in any of the current workflows.

FR_130 - Interaction with auditors for farm inspection: Not needed in any of the current workflows.

FR_132 - Update vineyard info if not in touch with consultant: Not needed in any of the current workflows.

FR_134 - Reach retailers to communicate the value of the certified product: Not needed in any of the current workflows.

FR_144 - Communicate complaints - accusations for certified producers or processors: Not needed in any of the current workflows.

FR_146 - Fill forms for inspection, report volumes, prices and food safety results to authorities, exchange reports with the certification bodies: These do not need to be filled by the platform, but will be provided by the third-party platforms in dataset form during upload.

ANNEX IV THEFSM NON-FUNCTIONAL REQUIREMENTS

Reference ID	Title	Category
NFR_1	TheFSM platform should provide personal data pseudonymization & GDPR compliance	Anonymization
NFR_2	Data access control based on attributes/ensure different authorisation access to different datasets	Access control
NFR_3	TheFSM platform should provide data representation & coverage of the food supply chain	Core functionality
NFR_4	FSM should provide monitoring capabilities of own data	Monitoring
NFR_5	FSM should ensure reliable communication with labs using API	Integration
NFR_6	FSM should scale well and handle notifications about actions, events, schedules.	Scalability
NFR_7	FSM should provide feedback utilities between actors.	Notifications
NFR_8	FSM should provide a multi-filtered and advanced search engine. Actors should be able to search products based on specific parameters, which certification body is related to issuing certifications they need, etc.	Search engine
NFR_9	FSM should provide agreement support.	Collaboration
NFR_10	FSM should provide support for risk estimation.	Risk assessment
NFR_11	FSM should have a DSS component.	DSS
NFR_12	FSM should be able to showcase custom views per product (based on prices, compliances, certifications, etc.).	Search engine
NFR_13	FSM should provide advanced profiles for companies and stakeholders.	Data views
NFR_14	Search engine should have past history, monitoring and traceability capabilities.	Search engine, traceability, monitoring

NFR_15	FSM should integrate data from various sources and databases	Integration
NFR_16	FSM should log important operations and transactions	Logging
NFR_17	FSM should provide online forms for actors to fill in, when needed (e.g., auditing reports, lab results).	Digital data input
NFR_18	FSM should periodically check and automatically notify actors in cases of legal compliance issues, or actions involving other actors.	Notifications
NFR_19	TheFSM should be able to verify the identity of the user/subject performing any operation	Verification
NFR_20	TheFSM should be able to trace all user/subject operations	Traceability
NFR_21	TheFSM should support an extended list of analytic algorithms on a mixture of confidential and public data in order to perform big data analytics	Analytics
NFR_22	TheFSM should be able to execute (big) data analytics in a timely and efficient manner	Performance, Analytics
NFR_23	TheFSM should be able to handle and store large datasets	Scalability
NFR_24	TheFSM should enable the interconnection and exchange of information with other platforms or devices with appropriate secure mechanisms	Integration
NFR_25	TheFSM should be able to support the functional and flexible operation in a distributed cloud infrastructure	Integration, Cloud infrastructure
NFR_26	TheFSM should be able to consume and handle different datasets in various formats (e.g. CSV, JSON, XML files)	Integration
NFR_27	TheFSM should be able to handle simultaneous requests on a timely and efficient manner	Scalability
NFR_28	TheFSM should provide the mechanisms to recover after system failure conditions	Recovery
NFR_29	TheFSM should have high availability	Availability
NFR_30	TheFSM platform should offer security and transparency regarding IPR management	Access control
NFR_31	TheFSM platform should offer a trusted way for data exchange agreements between two parties	Access control
NFR_32	TheFSM platform should support data representation using well established standards	Integration
NFR_33	TheFSM platform should avoid data duplication	Scalability
NFR_34	TheFSM platform should have a well-defined scope and structure	Core functionality
NFR_35	TheFSM platform should make its data FAIR	Core functionality

NFR_36	TheFSM platform should be designed in such a way that it can potentially become a part of a larger ecosystem	Core functionality
NFR_37	TheFSM platform should support interoperability for cross-device and cross-industry cases	Core functionality

Table 9: Non-Functional requirements – 2nd iteration

The above table displays the list of non-functional requirements which the second iteration of the platform should meet. When compared to the first, non-functional requirements NFR_33-NFR_37 were added. All of them are general requirements which are derived from design principles of state-of-the-art, robust and modern data markets. Since our goal is to provide a valuable platform for supply chain stakeholders, we should endeavor to meet those requirements especially.

ANNEX V THEFSM TECHNICAL REQUIREMENTS

Category	Sub-category	ID	Title	Related non-functional requirement	Related functional requirement
Data curation	Data collection, integration	TR_1	TheFSM platform shall allow data to be imported from external sources and systems	NFR_26	FR_14,FR_25,FR_26,FR_36,FR_40,FR_43,FR_47,FR_50,FR_64,FR_65,FR_66
	Data collection, integration	TR_2	TheFSM platform shall allow the user to upload and download files	NFR_26	FR_3,FR_4,FR_8,FR_9,FR_14,FR_25,FR_26,FR_36,FR_40,FR_43,FR_47,FR_50,FR_64,FR_65,FR_66
	Data collection, integration	TR_3	TheFSM platform shall allow the data ingestion of stream data	NFR_15, NFR_17, NFR_24, NFR_25, NFR_26	FR_14,FR_25,FR_26,FR_36,FR_40,FR_43,FR_47,FR_50,FR_64,FR_65,FR_66
	Data collection, integration	TR_4	TheFSM platform shall allow the data ingestion of batched data	NFR_15, NFR_17, NFR_24, NFR_25, NFR_26	FR_14,FR_25,FR_26,FR_36,FR_40,FR_43,FR_47,FR_50,FR_64,FR_65,

				24,NFR_25,NFR_26	FR_66
	Data management, integration	TR_5	TheFSM platform should provide data curation services	NFR_15, NFR_17, NFR_24, NFR_25, NFR_26	FR_76, FR_83, FR_84, FR_95, FR_96
	Data enrichment, integration	TR_6	TheFSM platform should provide data enrichment services for data deriving from internal and external data sources using a RESTful API	NFR_15, NFR_17, NFR_24, NFR_25, NFR_26	FR_7, FR_68, FR_69, FR_88, FR_89, FR_99, FR_102, FR_110, FR_123, FR_133, FR_135, FR_149
	Data management, integration	TR_7	TheFSM platform should offer a well-defined API for data export	NFR_15, NFR_17, NFR_24, NFR_25, NFR_26	FR_7, FR_68, FR_69, FR_88, FR_89, FR_99, FR_102, FR_110, FR_123, FR_133, FR_135, FR_149
	Data enrichment, integration	TR_8	TheFSM platform should develop and maintain a semantic model for food safety and certification data enrichment	NFR_15	FR_34, FR_35, FR_37, FR_38, FR_47, FR_73, FR_119
	Data management, integration, cloud infrastructure	TR_9	TheFSM platform should support updating and maintaining uploaded datasets	NFR_3, NFR_15, NFR_17, NFR_25	FR_3, FR_4, FR_8, FR_9, FR_34, FR_35, FR_37, FR_38, FR_47, FR_73, FR_119
	Data management, Integration	TR_10	TheFSM platform should support data representation using well established standards (GS1, EPCIS, WoT)	NFR_26, NFR_32	FR_12, FR_17, FR_20, FR_21, FR_36, FR_39, FR_40, FR_42, FR_52, FR_53, FR_54, FR_65, FR_75, FR_78, FR_79, FR_94, FR_106, FR_112,

					FR_126, FR_133,FR_150
	Scalability, integration	TR_11	TheFSM platform should offer a secure big data infrastructure	NFR_6, NFR_19, NFR_23, NFR_27	FR_82
Security and privacy	Access control, Search engine	TR_12	TheFSM platform should offer access control to data based specific parameters	NFR_8, NFR_14	FR_86,FR_100,FR_109
	Anonymization	TR_13	TheFSM platform should offer anonymization/pseudonymization services	NFR_1	
	Encryption, Access control	TR_14	TheFSM platform should encrypt data files	NFR_1, NFR_31	
	Authorization	TR_15	TheFSM platform should provide a controlled and secure way to decrypt data files	NFR_2	
	Authorization	TR_16	TheFSM platform should provide robust identity management for user authorization	NFR_1, NFR_2	
	Authorization	TR_17	TheFSM platform shall provide a secure and controlled registration process for new users	NFR_1, NFR_2	FR_1,FR_2
Licencing	Access control	TR_18	TheFSM platform should offer an IPR management service to data providers	NFR_30	FR_50,FR_73,FR_137
	Access control	TR_19	TheFSM platform shall store the data sharing contracts in a DLT-based repository for non-repudiation purposes.	NFR_30, NFR_31	FR_12, FR_17, FR_20, FR_21, FR_36, FR_39, FR_40, FR_42, FR_52, FR_53, FR_54, FR_55,FR_56,FR_57,FR_58, FR_94, FR_112, FR_126, FR_133

Traceability	Integration	TR_20	TheFSM platform shall use widely established standards (EPCIS) for traceability data	NFR_32	FR_12, FR_17, FR_20, FR_21, FR_36, FR_39, FR_40, FR_42, FR_52, FR_53, FR_54,FR_55,FR_56,FR_57,FR_58,FR_65,FR_75,FR_78,FR_79,FR_94, FR_106,FR_112, FR_126, FR_133,FR_150
	Traceability, monitoring, notifications	TR_21	TheFSM platform should capture the certification and auditing event in traceability data	NFR_14, NFR_16, NFR_20	FR_6,FR_7,FR_8,FR_12, FR_17, FR_20, FR_21, FR_22,FR_23,FR_24,FR_36, FR_39, FR_40, FR_42, FR_44,FR_45,FR_46,FR_47,FR_48,FR_49,FR_50,FR_52, FR_53, FR_54, FR_55,FR_56,FR_57,FR_58,FR_91,FR_92,FR_94,FR_101,FR_104,FR_107,FR_111, FR_112, FR_114,FR_117,FR_126, FR_130,FR_133,FR_140,FR_148
	Access control, Traceability	TR_22	TheFSM platform should use DLT for trust and transparency in traceability	NFR_30, NFR_31	FR_12, FR_17, FR_20, FR_21, FR_36, FR_39, FR_40, FR_42, FR_52, FR_53, FR_54,FR_55,FR_56,FR_57,FR_58, FR_94, FR_112, FR_126, FR_133
Data processing	Decision Support, Risk assessment	TR_23	TheFSM platform should offer services for risk assessment and prediction	NFR_10, NFR_11, NFR_2	FR_5,FR_18,FR_115

				1	
	Search engine, asset exporation	TR_24	TheFSM platform should offer query services for data asset exploration	NFR_8, NFR_12, NFR_14	FR_77,FR_86,FR_100,FR_109
	Search engine, asset exporation	TR_25	TheFSM platform shall retrieve and show the datasets that are relevant to a dataset that is returned as a query result.	NFR_8, NFR_12, NFR_14	FR_86,FR_100,FR_109
	Decision Support, Risk assessment	TR_26	TheFSM platform shall enable the integration and combined analysis over multiple datasets	NFR_8, NFR_10, NFR_11, NFR_12, NFR_14, NFR_21	FR_5,FR_18,FR_115
	Decision Support, Risk assessment	TR_27	TheFSM platform shall enable the application of predefined data analysis algorithms on datasets	NFR_10, NFR_11, NFR_21	FR_5,FR_18,FR_115
	Decision Support, Risk assessment	TR_28	TheFSM platform shall provide tools and services to apply machine learning algorithms	NFR_10, NFR_11, NFR_21	FR_5,FR_18,FR_115
	Decision Support, Risk assessment	TR_29	TheFSM platform shall provide tools and services to apply deep learning algorithms	NFR_10, NFR_11, NFR_21	FR_5,FR_18,FR_115
	Decision Support, Risk assessment	TR_30	TheFSM platform shall provide tools and services to apply basic analytics and statistics	NFR_10, NFR_11	FR_5,FR_18,FR_115

				,NFR_21	
Added value services	Notifications	TR_31	TheFSM platform should inform users with active contracts on a dataset that the dataset has been updated	NFR_4, NFR_7, NFR_16, NFR_18	FR_10,FR_11,FR_32,FR_80,FR_128,FR_141
	Monitoring	TR_32	TheFSM platform should provide data usage analytics to the users for the datasets they own.	NFR_4, NFR_16	FR_16,FR_27,FR_28,FR_29,FR_30,FR_31,FR_33,FR_67,FR_138
	Access control, authorization, authentication	TR_33	TheFSM platform shall ensure that access control over datasets is applied according to the data provider's policies and the terms of relevant active valid data sharing contracts	NFR_2	FR_94,FR_146,FR_148
Applications	Data views	TR_34	TheFSM platform shall enable the certification data exchange among the parties through intuitive UIs	NFR_9, NFR_13, NFR_31	FR_13,FR_19,FR_21,FR_41,FR_59,FR_60,FR_61,FR_62,FR_63,FR_71,FR_77,FR_81
	Data views	TR_35	TheFSM platform shall provide certification data to food safety stakeholders through intuitive UIs	NFR_9, NFR_13, NFR_31	FR_13,FR_19,FR_21,FR_41,FR_59,FR_60,FR_61,FR_62,FR_63,FR_71,FR_77,FR_81
	API integration, data processing	TR_36	TheFSM platform shall integrate with the application using RESTful APIs exchanging data in json format	NFR_9, NFR_13, NFR_17, NFR_24, NFR_25, NFR_26, NFR_31	FR_34,FR_35,FR_37,FR_38,FR_47,FR_73,FR_119
Licensing	Asset exploration, data views	TR_37	TheFSM platform shall have well-define overall terms, conditions and data categories	NFR_32, NFR_34	FR_12, FR_17, FR_20, FR_21, FR_36, FR_39, FR_40, FR_42, FR_52, FR_53,

				NFR_35 , NFR_36	FR_54,FR_65,FR_75,FR_78,FR_79,FR_94,FR_106,FR_112,FR_126,FR_133,FR_150
Added value services	Access control, authorization, authentication, integration, API integration	TR_38	TheFSM platform shall provide a services registry for the incorporation of third-party services	NFR_36 , NFR_37	FR_34,FR_35,FR_37,FR_38,FR_47,FR_73,FR_119
Data curation	Data views	TR_39	TheFSM platform shall allow data consumers to sample the data they are going to buy to ensure they are satisfied with their purchase	NFR_12	
Data curation	Data enrichment, data processing	TR_40	TheFSM platform shall ensure consistent high data quality	NFR_26 , NFR_29 , NFR_32 , NFR_33 , NFR_35	FR_65
Licensing	Data management	TR_41	TheFSM platform shall allow users to monetize their data	NFR_9, NFR_30 , NFR_31	

Table 10: Technical requirements – 2nd iteration

The above table displays the list of technical requirements which the second iteration of the platform should meet. When compared to the first, technical requirements TR_37-TR_41 were added. Like the additions in non-functional requirements, all of them are general requirements which are derived from design principles of state-of-the-art, robust and modern data markets. They match the respective new non-functional requirements and some pre-existing ones too. It is important to note that we also take into consideration the requirements of the IPR management and the Data Governance into consideration.

ANNEX VI THEFSM TECHNICAL REQUIREMENTS MAPPING TO USER STORIES

Technical Requirements	Title	Related user stories
TR_1	TheFSM platform shall allow data to be imported from external sources and systems	US_4, US_6, US_7, US_10, US_13, US_18, US_21, US_22, US_28
TR_2	TheFSM platform shall allow the user to upload and download files	US_7, US_9, US_30, US_18, US_21, US_28
TR_3	TheFSM platform shall allow the data ingestion of stream data	US_7, US_9, US_10, US_13, US_18, US_21, US_22, US_28
TR_4	TheFSM platform shall allow the data ingestion of batched data	US_4, US_6, US_9, US_10, US_13, US_18, US_21, US_22, US_28
TR_5	TheFSM platform should provide data curation services	US_11, US_13, US_18, US_28
TR_6	TheFSM platform should provide data enrichment services for data deriving from internal and external data sources using a RESTful API	US_40,US_19,US_41,US_20,US_42,US_21,US_43,US_22,US_1,US_44,US_23,US_2,US_45,US_24,US_3,US_46,US_25,US_4,US_47,US_26,US_5,US_48,US_27,US_6,US_49,US_28,US_7,US_50,US_29,US_8,US_51,US_30,US_9,US_31,US_10,US_32,US_11,US_33,US_12,US_34,US_13,US_35,US_14,US_36,US_15,US_37,US_16,US_17,US_39,US_18
TR_7	TheFSM platform should should offer a well-defined API for data export	US_18,US_40,US_19,US_41,US_20,US_42,US_21,US_43,US_22,US_1,US_44,US_23,US_2,US_45,US_24,US_3,US_46,US_25,US_4,US_47,US_26,US_5,US_48,US_27,US_6,US_49,US_28,US_7,US_50,US_29,US_8,US_51,US_30,US_9,US_31,US_10,US_32,US_11,US_33,US_12,US_34,US_13,US_35,US_14,US_36,US_15,US_37,US_16,US_38,US_17,US_39
TR_8	TheFSM platform should develop and maintain a semantic model for food safety and certification data enrichment	US_6, US_7, US_11, US_22
TR_9	TheFSM platform should support updating and maintaining uploaded datasets	US_7, US_9, US_10, US_22, US_26, US_30

TR_10	TheFSM platform should support data representation using well established standards (GS1, EPCIS, WoT)	US_7, US_10, US_13, US_15, US_26, US_29, US_34, US_41, US_43
TR_11	TheFSM platform should offer a secure big data infrastructure	
TR_12	TheFSM platform should offer access control to data based specific parameters	US_7, US_16, US_39
TR_13	TheFSM platform should offer anonymization/pseudonymization services	
TR_14	TheFSM platform should encrypt data files	US_16
TR_15	TheFSM platform should provide a controlled and secure way to decrypt data files	US_16
TR_16	TheFSM platform should provide robust identity management for user authorization	US_5, US_15, US_16
TR_17	TheFSM platform shall provide a secure and controlled registration process for new users	US_16
TR_18	TheFSM platform should offer an IPR management service to data providers	
TR_19	TheFSM platform shall store the data sharing contracts in a DLT-based repository for non-repudiation purposes.	US_13, US_29, US_31, US_41
TR_20	TheFSM platform shall use widely established standards (EPCIS) for traceability data	US_7, US_26, US_27, US_29, US_34, US_41, US_43
TR_21	TheFSM platform should capture the certification and auditing event in traceability data	US_9, US_22, US_25, US_29, US_30, US_34, US_41,
TR_22	TheFSM platform should use DLT for trust and transparency in traceability	US_5, US_7, US_12, US_15, US_23, US_26, US_27, US_28, US_31, US_34, US_35, US_42, US_43, US_44, US_47
TR_23	TheFSM platform should offer services for risk assessment and prediction	US_14, US_24, US_30, US_40, US_48
TR_24	TheFSM platform should offer query services for data asset exploration	US_11, US_13, US_17, US_19, US_20, US_23, US_24, US_25, US_26, US_27, US_28, US_29, US_30, US_32, US_34, US_35, US_36, US_37, US_39, US_40, US_41, US_43, US_44, US_47

TR_25	TheFSM platform shall retrieve and show the datasets that are relevant to a dataset that is returned as a query result.	US_1, US_11, US_13, US_14, US_17, US_19, US_20, US_23, US_25, US_26, US_27, US_28, US_29, US_30, US_32, US_34, US_35, US_36, US_37, US_39, US_40, US_41, US_43, US_44, US_47
TR_26	TheFSM platform shall enable the integration and combined analysis over multiple datasets	US_1, US_2, US_11, US_12, US_19, US_23, US_26, US_27, US_28, US_30, US_36, US_43, US_45
TR_27	TheFSM platform shall enable the application of predefined data analysis algorithms on datasets	US_12, US_14, US_26, US_30, US_33, US_43, US_45
TR_28	TheFSM platform shall provide tools and services to apply machine learning algorithms	US_30, US_31, US_45
TR_29	TheFSM platform shall provide tools and services to apply deep learning algorithms	US_30, US_31, US_45
TR_30	TheFSM platform shall provide tools and services to apply basic analytics and statistics	US_12, US_19, US_26, US_30, US_31, US_33, US_36, US_40
TR_31	TheFSM platform should inform users with active contracts on a dataset that the dataset has been updated	US_3, US_9, US_12, US_14,
TR_32	TheFSM platform should provide data usage analytics to the users for the datasets they own.	US_2
TR_33	TheFSM platform shall ensure that access control over datasets is applied according to the data provider's policies and the terms of relevant active valid data sharing contracts	US_2, US_28, US_30, US_32, US_34, US_47
TR_34	TheFSM platform shall enable the certification data exchange among the parties through intuitive UIs	US_5, US_15, US_17, US_18, US_20, US_21, US_22, US_24, US_25, US_29, US_30, US_32, US_34, US_35, US_39, US_41, US_44, US_47, US_49
TR_35	TheFSM platform shall provide certification data to food safety stakeholders through intuitive UIs	US_15, US_17, US_18, US_20, US_21, US_22, US_24, US_25, US_29, US_32, US_34, US_35, US_39, US_41, US_47, US_49
TR_36	TheFSM platform shall integrate with the application using RESTful APIs exchanging data in json format	US_7

TR_37	TheFSM platform shall have well-define overall terms, conditions and data categories	US_29, US_41
TR_38	TheFSM platform shall provide a services registry for the incorporation of third-party services	US_7
TR_39	TheFSM platform shall allow data consumers to sample the data they are going to buy to ensure they are satisfied with their purchase	US_34, US_47, US_50
TR_40	TheFSM platform shall ensure consistent high data quality	US_5, US_17, US_28, US_32, US_34, US_35, US_39, US_41, US_42, US_46, US_49
TR_41	TheFSM platform shall allow users to monetize their data	US_31, US_38, US_46, US_47, US_50

Table 11: Technical requirements mapping to user stories

Finally, for the sake of completeness, we provide a mapping between technical requirements and user stories, which was not present in the first iteration of D3.1.

ANNEX VII THEFSM COMPONENTS MAPPING TO TECHNICAL REQUIREMENTS

Functional component needed	Technical Requirement Reference ID
OTNode DLT Services and Interfaces	TR_11, TR_17, TR_21, TR_31, TR_32
A2C engine	TR_8, TR_9, TR_12, TR_16, TR_17, TR_21, TR_24, TR_25, TR_26, TR_31, TR_32, TR_34, TR_35
OTNode DLT Services and Interfaces	TR_8, TR_32, TR_34, TR_35
Data Brokerage Engine and Model, Crypto Wallet	TR_8, TR_32, TR_34, TR_35, TR_41
Food Inspector	TR_8, TR_9, TR_21, TR_32, TR_34, TR_35
Secure storage and indexing	TR_1, TR_2, TR_3, TR_4, TR_5, TR_6, TR_7, TR_8, TR_9, TR_10, TR_12, TR_14, TR_21, TR_24, TR_25, TR_26, TR_32, TR_34, TR_35, TR_36
Data Handler	TR_1, TR_2, TR_3, TR_4, TR_5, TR_6, TR_7, TR_8, TR_9, TR_10, TR_12, TR_14, TR_21, TR_24, TR_25, TR_26, TR_32, TR_34, TR_35, TR_36, TR_39

Data Staging	TR_1, TR_2, TR_3, TR_4, TR_5, TR_6, TR_7, TR_8, TR_9, TR_10, TR_12, TR_14, TR_21, TR_24, TR_25, TR_26, TR_32, TR_34, TR_35, TR_36, TR_39, TR_40
Data Encryption Service	TR_1, TR_2, TR_3, TR_4, TR_5, TR_6, TR_7, TR_8, TR_9, TR_10, TR_12, TR_14, TR_21, TR_24, TR_25, TR_26,, TR_34, TR_35, TR_36
Foodakai 2.0	TR_4, TR_14
Agrivi 2.0	TR_4, TR_9, TR_14, TR_21
Semantic Mapper	TR_1, TR_3, TR_4, TR_5, TR_6, TR_8, TR_10, TR_36
Message Brokerage	TR_9, TR_31
AI Models	TR_10, TR_23, TR_26, TR_27, TR_28, TR_29, TR_30
Data Licence and Agreement management	TR_13, TR_34, TR_35, TR_37
Analytics-as-a-service, Data Encryption Service, OTNode DLT Services and Interfaces	TR_21, TR_22, TR_32
Query explorer	TR_12, TR_21, TR_24, TR_25, TR_26
Data sources and application catalogue (data publish)	TR_12, TR_21, TR_24, TR_25, TR_26, TR_40
API Gateway	TR_38

Table 12: Components mapping to technical requirements