Research and Innovation Action
HORIZON-CL6-2022-GOVERNANCE-01
TOPIC: HORIZON-CL6-2022-GOVERNANCE-01-11

AgriDataValue

Smart Farm and Agri-environmental Big Data Value

Deliverable D1.3
AgriDataSpace Technical Specifications & Reference Architecture

Authors	S. Neicu, F.Crihan (The complete list of contributors is included in document's history)
Nature	Report
Dissemination	Public
Version	V1.0
Status	Final
Delivery Date (DoA)	M08
Actual Delivery Date	05/10/2023

Keywords	Technical Specification, Data Model, Reference Architecture, Security
Abstract	This deliverable describes the technical specifications of the AgriDataSpace components, the low-level design of the architectural patterns and the reference architecture. The technical specifications provide detailed information regarding performance, security, and reliability, as well as outline specific implementation constraints associated with the system. The ADV Reference Architecture (RA) depicts the architecture in its evolved state from the beginning of the project. The RA has been influenced by the numerous discussions held during the first period of the project, either in group telcos (e.g., biweekly Architecture telco since M4) and meetings or in peer-to-peer telcos where several aspects of the platform's components and functionality were discussed. In this deliverable we present several views of the architecture, namely: High-level view, Functional view, Process view, Data view and Deployment view.

Funded by the European Union under Grant Agreement no. 101086461

ACKNOWLEDGEMENT

The AgriDataValue project is funded by the European Union under Grant Agreement No. 101086461. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency, while neither the European Union nor the granting authority can be held responsible for any use of this content. No part of this document may be used, reproduced and/or disclosed in any form or by any means without the prior written permission of the AgriDataValue consortium.

	Participant organisation name	Short	Country
01	SYNELIXIS SOLUTIONS S.A.	SYN	EL
02	ATOS IT SOLUTIONS AND SERVICES IBERIA SL	ATOS	ES
03	SIXENSE ENGINEERING	SIXEN	FR
04	NETCOMPANY-INTRASOFT SA	INTRA	LU
05	SIEMENS SRL	SIEM	RO
06	SINERGISE LABORATORIJ ZA GEOGRAFSKEINFORMACIJSKE SISTEME DOO	SINER	SI
07	ALMAVIVA - THE ITALIAN INNOVATION COMPANY SPA	ALMA	IT
08	INTERNATIONAL DATA SPACES EV	IDSA	DE
09	SOFTWARE IMAGINATION & VISION SRL	SIMAVI	RO
10	SINGULARLOGIC S.A.	SLG	EL
11	EIGEN VERMOGEN VAN HET INSTITUUT VOOR LANDBOUW- EN VISSERIJONDERZOEK	EV ILVO	BE
12	ETHNIKO KAI KAPODISTRIAKO PANEPISTIMIO ATHINON	NKUA	EL
13	INAGRO, PROVINCIAAL EXTERNVERZELFSTANDIGD AGENTSCHAP IN PRIVAATRECHTELIJKE VORM VZW	InAgro	BE
14	UNIWERSYTET LODZKI	UL	PL
15	FUNDACION PARA LAS TECNOLOGIAS AUXILIARES DE LA AGRICULTURA	TEC	ES
16	DELPHY BV	Delphy	NL
17	INSTITUTO TECNOLOGICO DE ARAGON	ITAIN	ES
18	ZEMNIEKU SAEIMA	ZSA	LV
19	SOCIEDAD ARAGONESA DE GESTION AGROAMBIENTAL SL	SARGA	ES
20	AGROTIKOS KTINOTROFIKOS SYNETAIRISMOS KATOUNAS TO VIOLOGIKO AGROKTIMA	TBA	EL
21	SOCIETA ITALIANA DI VITICOLTURA ED ENOLOGIA	SIVE	IT
22	NILEAS-SYNETAIRISMOS PISTOPOIIMENON AGROTIKON PROIONTON DIMOU NESTOROS MESSINIAS	NILEAS	EL
23	CONSEIL DES VINS DE SAINT-EMILION	CVSE	FR
24	ASOCIATIA OPERATORILOR DIN AGRICULTURA ECOLOGICA BIO ROMANIA	BIORO	RO
25	RI.NOVA SOCIETA COOPERATIVA	RI.NO	IT
26	AGRO DIGITAL SOLUTIONS	AgroDS	LT
27	NATIONAL PAYING AGENCY	NPA	LT
28	AGENZIA PROVINCIALE PER I PAGAMENTIDELLA PROVINCIA AUTONOMA DI TRENTO	APPAG	IT
29	AGENTIA DE PLATI SI INTERVENTIE PENTRU AGRICULTURA	APIA	RO
30	QUEEN MARY UNIVERSITY OF LONDON	QMUL	UK

DISCLAIMER

Document History

Version	Date	Contributor(s)	Description
v0.1	20/08/2023	Y. Oikonomidis, I. Chrysakis (INTRA)	Added content for sections 5, 6, 7.
v0.1.1	21/08/2023	S.Neicu(SIMAVI)	Added content for sections 1, 3, 9.
v0.1.2	01/09/2023	A. Turkmayali (IDSA)	Updated chapter 4
v0.1.3	05/09/2023	Y. Oikonomidis, I. Chrysakis (INTRA), I. Ilie (SIEM), R. Lazcano, P. Ramirez (ATOS), A. Retico (ALMA), S. Bourou, K. Railis (SYN), I. Sotiropoulos (SLG)	Added Logical view diagrams
v0.2	12/09/2023	S.Neicu(SIMAVI)	Incorporate partners updates
v0.2.1	14/09/2023	R. Lazcano (ATOS), P. Ramirez (ATOS)	First revision
v0.2.2	28/09/2023	M. Perdikeas, K. Railis (SYN), I. Sotiropoulos, S. Rizou (SLG), A. Retico, S. Sestili (ALMA), I. Oikonomidis, I. Chrysakis (INTRA), A. Turkmayali (IDSA)	Added additional content to section 5, addressing internal reviewers comments.
v0.3	28/09/2023	S.Neicu (SIMAVI)	Incorporate partners updates
v0.4	30/09/2023	I. Ilie (SIEM), I. Oikonomidis, I. Chrysakis (INTRA), M. Perdikeas, K. Railis (SYN)	Added content to sections 2 and 5. Merged input. New document version
v0.4.1	04/10/2023	I. Oikonomidis, I. Chrysakis (INTRA), A. Skias, M. Perdikeas, K. Railis (SYN)	Added final content in several sections (1, 2, 8).
V0.1	05/10/2023	Th. Zahariadis (SYN)	

Document Reviewers

Date	Reviewer's name	Affiliation
14/09/2023	R. Lazcano	ATOS
14/09/2023	P. Ramirez	ATOS
05/10/2023	T. Zahariadis	SYN

Table of Contents

Table of Figures

Table of Tables

Definitions, Acronyms and Abbreviations

ADV	AgriDataValue
AI	Artificial Intelligence
AIM	Agriculture Information Model
API	Application Programmable Interface
CAS	Central Authentication Service
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
DLT	Distributed Ledger Technology
EO	Earth Observation
FL	Federated Learning
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
HTTPS	HyperText Transfer Protocol Secure
IDS	International Data Spaces
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organisation for Standardisation
KP-ABE	Key-Policy Attribute-Based Encryption
ML	Machine Learning
OWL	Ontology Web Language
RA	Reference Architecture
RDF	Resource Description Framework
REST	Representational State Transfer
TCP	Transmission Control Protocol
TL	Task Leader
TM	Technical Manager
UML	Unified Modelling Language
WP	Work Package
WPL	Work Package Leader
XAI	Human Explainable Artificial Intelligence

Executive Summary

The aim of this deliverable is to provide AgriDataValue technical specifications and reference architecture. This document initially includes the details that define the system requirements and technical specifications of the solution provided by the ADV (AgriDataValue) platform of platforms. The technical requirements are mapped to the user requirements, as defined in deliverable D1.1, to provide an overview of the coverage of the platform's functionality with respect to the user needs.

The description of the approach towards the definition of the Data model that will be used in the context of the AgriDataValue platform is also presented in this document. The data model is expected to play an integral role in the interoperability aspects of the platform as it is envisioned to combine well established information models (i.e., the Agriculture Information Model (AIM) and the IDS data models)

It then discusses the methodology employed to design the Reference Architecture, it describes in detail the actual AgriDataValue Reference Architecture through several viewpoints (i.e., high-level, functional, process, data, and deployment views), and it gives an overview of the main interactions among the core building blocks of the Reference Architecture. Finally, the document concludes with an elaboration on the GDPR concerns and guidelines that will need to be enforced to protect the data and privacy of the various AgriDataValue stakeholders.

This document is based on the user requirements identified in Task 1.1, the key technology areas identified in Task 1.2, the outcome of the work performed in T1.3 on semantic interoperability, the work of T1.4 on technical requirements and specifications, and the work on the definition of the reference architecture of the AgriDataValue platform in T1.5. Of course, all component owners in the consortium were also involved throughout the elicitation of this report providing the latest input on their components.

1 Introduction

1.1 Scope and purpose

This deliverable presents the first version of Deliverable D1.3 - AgriDataSpace Technical Specifications and Reference Architecture that will be used to guide the development of technologies in the technical work packages (WP2, WP3, and WP4). The specifications in this document are based on a detailed analysis, as well as the initial technical requirements extracted from the technical work packages and, of course, from the ADV vision and targeted objectives.

The document serves as a unique guide for building the solution for the project's technical partners and details concretely the technical and functional way in which the functionalities described in D1.1 – Definition and analysis of use cases and system requirements will be implemented.

1.2 Document overview

This document includes the following sections:

Section 2 discusses the Technical Requirements and Specifications, presenting the methodology followed for these, including a mapping among technical and user requirements, while it provides an overview of the technical specifications in place to guide the development of the ADV components.

Section 3 presents the approach followed by ADV towards the definition of the data model to be used throughout the ADV platform's components. It describes the initial steps required and the key aspects that need to be addressed with a view to the interoperability of the platform.

Section 4 presents the methodology used during the definition of the Reference Architecture in ADV.

Section 5 describe the ADV Reference Architecture through several viewpoints.

Section 6 presents a view of the main interactions and dependencies between the software components of the architecture and a high-level description of the interfaces that need to be in place between these components.

Section 7 presents details on general Security and GDPR aspects related with the architecture.

Finally, *Section 8* concludes the document and *Section 9* provides the respective references used.

The architecture presented in this deliverable will be complemented and completed by the following deliverables as they become available, and which detail the updated reports regarding the various AgriDataValue tools and components: D1.4, D2.1, D2.2, and D2.3.

2 Technical Requirements and Specifications

This chapter presents the system requirements considered by the ADV project to cover the overall technical capabilities of the envisaged solution, describing the requirements related to the availability, scalability and performance of the entire system, and drawing a set of constraints and initial technical specifications for the target environment to meet these requirements.

These system requirements form the baseline for designing the overall system architecture as a highly available, scalable, and extensible solution.

2.1 Methodology description

The common classification of requirements divides them into functional and non-functional requirements. Functional requirements capture the fundamental properties of the system, describing its processes, actions and technical properties. Non-functional requirements capture the properties that system functions must have, such as performance, usability, GDPR compliance, and security. Unlike functional requirements, non-functional requirements focus on the quality characteristics of the system.

The proposed methodology for prioritizing and refining the functional and non-functional requirements of the platform includes the following stages:

- Defining a form for collecting information about ADV tools/modules. To begin with, a detailed description of the instruments/sensors to be used is required, detailing the key technological areas to which it will contribute. Based on the information collected through the form, the necessary basic functionalities will be identified.
- Afterwards, in order to identify the functionalities and additional aspects of the system that must be satisfied by the technical solution, the use cases defined in Task 1.1: Use case refinement & end-user requirements will be examined.
- Finally, to conclude the definition of technical requirements with the collaboration of all partners, a series of brainstorming sessions will create an ADV technical requirements template through which each partner will identify and report all functional and non-functional requirements. The prioritization of technical requirements will follow the MoSCoW [1] prioritization approach, to define the criticality of each requirement:
  - Must have: A label used for the most critical requirements which, if not delivered, could jeopardise the success of the whole project.
  - Should have: Denotes requirements that are important but not critical for the success of the implementation. Such requirements can be less time-critical than “Must have” requirements or there might be alternative ways to satisfy them.

[1] https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html

- Could have: This category includes requirements that are desirable but not necessary. They usually include aspects relating to user experience or customer satisfaction which can be implemented if time and resources permit it.
- Won't have: Requirements in this category are the least critical and shall not be included or shall be reconsidered during later stages of the implementation.

2.2 Technical requirements

The Technical (System) Requirements of the ADV platform were developed using the aforementioned methodology based on input from the stakeholders (as provided with the User Requirements included in deliverable D1.1). The table below lists each technical requirement's basic details (ID, Title, Type, Priority, and Related Component), along with any associated User requirements. The table includes additional entries compared to the list of technical requirements reported in D1.1, while "N/A" means that there is no specific related component (the components' acronyms are defined in section 5) for that technical requirement. At the time this deliverable was submitted, 32 of 32 user requirements (or 100%) were covered by at least one technical requirement. The final table column will have data entered into it during the validation process, and it will be reported in WP2 deliverables D2.1, D2.2, and D2.3 where the integrated platform's versions will be reported. The technical requirements list is expected to be further updated in the next few months while the development of the platform's components and the interaction with the end users will be quite intensive; any updates will be reported in the upcoming WP2 deliverable D2.1. This is the initial identified system requirements (86) for the ADV platform:

Table 1: Mapping among Technical requirements and User requirements

ID	Type	Title	Related component	Priority	Related User Requirement	Verification Method (Description/Demonstration of functionality, Achieved/Not Achieved, N/A)
Weather/Micro-clima related system requirements						
REQ.FN.01.01	Functional	Weather monitoring	IOTD	M	REQ.US.01.01, REQ.US.01.06, REQ.US.01.07	
REQ.FN.01.02	Functional	Weather Parameters	IOTD	M	REQ.US.01.01	
REQ.FN.01.03	Functional	Additional Parameters	IOTD	C	REQ.US.01.03	
REQ.FN.01.04	Functional	Calculated Parameters	IOTD	C	REQ.US.01.01, REQ.US.01.02, REQ.US.01.03, REQ.US.01.04, REQ.US.01.05, REQ.US.01.06, REQ.US.01.07, REQ.US.01.10	

ID	Type	Title	Related component	Priority	Related User Requirement	Verification Method (Description/Demonstration of functionality, Achieved/Not Achieved, N/A)
REQ.FN.01.05	Functional	Weather impact assessment	IOTD, FDML, XAI, DKM	M	REQ.US.01.01, REQ.US.01.02, REQ.US.01.03, REQ.US.01.04, REQ.US.01.05, REQ.US.01.08, REQ.US.01.09, REQ.US.01.10, REQ.US.01.11, REQ.US.01.12, REQ.US.01.13, REQ.US.01.16, REQ.US.01.17	
Soil related system requirements						
REQ.FN.02.01	Functional	Location	DRD, EOD	S	REQ.US.01.01 - REQ.US.01.17	
REQ.FN.02.02	Functional	Soil parameters	IOTD	S	REQ.US.01.01 - REQ.US.01.17	
REQ.FN.02.03	Functional	Data Management	STORE, SECURESTORE	S	REQ.US.01.01 - REQ.US.01.17	
REQ.FN.02.04	Functional	Reporting and Visualization	N/A	S	REQ.US.01.01 - REQ.US.01.17	
REQ.FN.02.05	Functional	Decision Support	FDML, DKM	S	REQ.US.01.01 - REQ.US.01.17	
REQ.FN.02.06	Functional	Support for advice	XAI	C	REQ.US.01.01 - REQ.US.01.17	
Greenhouse Air Quality related system requirements						
REQ.FN.03.01	Functional	Air Quality Parameters	IOTD, STORE, SECURESTORE	M	REQ.US.01.01 - REQ.US.01.13	
REQ.FN.03.02	Functional	Additional parameters of interest to be measured inside the greenhouse	IOTD, STORE, SECURESTORE	C	REQ.US.01.01 - REQ.US.01.13	
Farm Air Quality related system requirements						
REQ.FN.04.01	Functional	Air Quality Parameters	IOTD, STORE, SECURESTORE	M	REQ.US.01.01 - REQ.US.01.13	
REQ.FN.04.02	Functional	Additional Parameters	IOTD, STORE, SECURESTORE	C		
Livestock wellbeing related system requirements						

ID	Type	Title	Related component	Priority	Related User Requirement	Verification Method (Description/Demonstration of functionality, Achieved/Not Achieved, N/A)
REQ.FN.05.01	Functional	Air Quality Parameters	IOTD, STORE, SECURESTORE	M	REQ.US.02	
REQ.FN.05.02	Functional	Feed Quality	IOTD, STORE, SECURESTORE	M	REQ.US.02	
REQ.FN.05.03	Functional	Feed intake	IOTD, STORE, SECURESTORE	M	REQ.US.02	
REQ.FN.05.04	Functional	Bedding material	IOTD, STORE, SECURESTORE	M	REQ.US.02	
REQ.FN.05.05	Functional	Milk production parameters	IOTD, STORE, SECURESTORE	M	REQ.US.02	
REQ.FN.05.06	Functional	Weight	IOTD, STORE, SECURESTORE	S	REQ.US.02	
REQ.FN.05.07	Functional	Moving behaviour	IOTD, STORE, SECURESTORE	S	REQ.US.02	
REQ.FN.05.08	Functional	Camera's	IOTD, STORE, SECURESTORE	S	REQ.US.02	
REQ.FN.05.09	Functional	RFID tag	IOTD, STORE, SECURESTORE	S	REQ.US.02	
Terrestrial Geotagged-Photos' Capturing system requirements						
REQ.FN.06.01	Functional	Image quality	DRD, EOD	S	REQ.US.01	
REQ.FN.06.02	Functional	Image pre-processing	IOTD, STORE, SECURESTORE	C	REQ.US.01	
REQ.FN.06.03	Functional	Image analytics	IOTD, STORE, SECURESTORE, FDML, DKM	M	REQ.US.01	
REQ.FN.06.04	Functional	Knowledge extraction	IOTD, STORE, SECURESTORE, FDML, DKM	M	REQ.US.01	
CAP related actions system requirements						
REQ.FN.07.01	Functional	CAP supervisory services	IOTD, DRD, EOD	C	REQ.US.01	
REQ.FN.07.02	Functional	Weather and livestock data	IOTD	C	REQ.US.02	
REQ.FN.07.03	Functional	Economic risk assessment	N/A	C	ALL	
REQ.FN.07.04	Functional	Comparative evaluation and monitoring of	EOD, DRD	C	REQ.US.01	

ID	Type	Title	Related component	Priority	Related User Requirement	Verification Method (Description/Demonstration of functionality, Achieved/Not Achieved, N/A)
		ecological schemes				
REQ.FN.07.05	Functional	Food security in the face of climate change and biodiversity loss	DKM, FDML, XAI	C	ALL	
REQ.FN.07.06	Functional	Global transition towards competitive sustainability from farm to fork	CHAINTRACK	C	ALL	
Satellite Earth Observation Capturing requirements						
REQ.FN.08.01	Functional	EO imagery catalogue	EOD	S	REQ.US.01	
REQ.FN.08.02	Functional	EO imagery access	EOD	S	REQ.US.01	
REQ.FN.08.03	Functional	Ingestion of and access to other datasets	EOD	S	REQ.US.01	
REQ.FN.08.04	Functional	EO data processing	EOD	S	REQ.US.01	
REQ.FN.08.05	Functional	EO Large Scale (raster) processing	EOD	S	REQ.US.01	
REQ.FN.08.06	Functional	EO Large Scale (object-based) processing	EOD	S	REQ.US.01	
Data Sovereignty related requirements						
REQ.FN.09.01	Functional	Data Federation	STORE, SECURE	S	ALL	
REQ.FN.09.02	Functional	Data Openness	IDS	S	ALL	
REQ.FN.09.03	Functional	Blockchain & NFTs	SECURESTORE	S	ALL	
REQ.FN.09.04	Functional	Usage Policies & enforcement	IDS	S	ALL	
REQ.FN.09.05	Functional	Data discoverability & observability	IDS	S	ALL	

ID	Type	Title	Related component	Priority	Related User Requirement	Verification Method (Description/Demonstration of functionality, Achieved/Not Achieved, N/A)
Data Interoperability related requirements						
REQ.FN.10.01	Functional	Data Standardization	STORE, SECURESTORE, IDS	S	ALL	
REQ.FN.10.02	Functional	Data Mapping and Transformation	IOTD, EOD, DRD, IDS	S	ALL	
REQ.FN.10.03	Functional	Data Quality Assurance	IDS, SECURESTORE	S	ALL	
REQ.FN.10.04	Functional	Data Governance and Metadata Management	IDS	S	ALL	
REQ.FN.10.05	Functional	Data Exchange Protocols and APIs	IDS	S	REQ.US.01.01 - REQ.US.01.13, REQ.US.02	
Federated ML related requirements						
REQ.FN.11.01	Functional	Agent identification	FDML, DKM, ACS	S		
REQ.FN.11.02	Functional	Global parameters communication	DKM, FDML	S		
REQ.FN.11.03	Functional	Local training	FDML	S	REQ.US.01.01 - REQ.US.01.13	
REQ.FN.11.04	Functional	Aggregated model	DKM, FDML	M		
REQ.FN.11.05	Functional	Anomaly detection	DATAGEN, DKM	M		
REQ.FN.11.06	Functional	Data preprocessing	FDML	M		
REQ.FN.11.07	Functional	Synthetic Data Generation for training	FDML, DKM, XAI	S		
System level Requirements						
REQ.FN.12.01	Functional	Authentication/ Authorization	ACS	M	REQ.US.01.01 - REQ.US.01.13, REQ.US.02	
REQ.FN.12.02	Functional	IoT data collection	IOTD	S	REQ.US.01.01 - REQ.US.01.13, REQ.US.02	

ID	Type	Title	Related component	Priority	Related User Requirement	Verification Method (Description/Demonstration of functionality, Achieved/Not Achieved, N/A)
REQ.FN.12.03	Functional	IoT data transmission	IOTD	S	REQ.US.01.01 - REQ.US.01.13, REQ.US.02	
REQ.FN.12.04	Functional	IoT sensors/devices management	IOTD	C	REQ.US.01.01 - REQ.US.01.13	
REQ.FN.12.05	Functional	IoT device connectivity	IOTD	M	REQ.US.01.01 - REQ.US.01.13, REQ.US.02	
REQ.FN.12.06	Functional	Historical data processing	STORE, SECURESTORE, IOTD, DKM	S	REQ.US.01.01 - REQ.US.01.13, REQ.US.02	
REQ.FN.12.07	Functional	Data access	IOTD, STORE, SECURESTORE	S	ALL	
REQ.FN.12.08	Functional	IoT data storage	STORE, SECURESTORE	M	ALL	
REQ.FN.12.09	Functional	Processed data storage	STORE, SECURESTORE	S	ALL	
REQ.FN.12.10	Functional	Satellite image processing	EOD	S	REQ.US.01.01 - REQ.US.01.13	
REQ.FN.12.11	Functional	Satellite image exposure API	EOD	S	REQ.US.01.01 - REQ.US.01.13	
REQ.FN.12.12	Functional	Standard APIs	IDS	S	REQ.US.01.01 - REQ.US.01.13, REQ.US.02	
REQ.FN.12.13	Functional	Receiving control commands	IOTD	C	ALL	
REQ.FN.12.14	Functional	Data reports	N/A	C	ALL	
REQ.FN.12.15	Functional	Data geo-visualization	N/A	C	ALL	
REQ.FN.12.16	Functional	Monitoring	N/A	S	ALL	
UC related requirements						
REQ.FN.13.01	Functional	Field status	N/A	C		
REQ.FN.13.02	Functional	Pest infestation identification	DKM, FDML	S	REQ.US.01.13	
REQ.FN.13.03	Functional	Disease outbreaks module	DKM, FDML	S	REQ.US.01.13	

ID	Type	Title	Related component	Priority	Related User Requirement	Verification Method (Description/Demonstration of functionality, Achieved/Not Achieved, N/A)
REQ.FN.13.04	Functional	Disease outbreaks alert	N/A	S	REQ.US.01.13	
Non - Functional Requirements						
REQ.NFN.01	Non-functional	Low latency	STORE, SECURESTORE	S	ALL	
REQ.NFN.02	Non-functional	Availability	N/A	S	ALL	
REQ.NFN.03	Non-functional	Scalability	N/A	S	ALL	
REQ.NFN.04	Non-functional	Usability	N/A	S	ALL	
REQ.NFN.05	Non-functional	Security & Privacy	ACS, SECURESTORE, IDS	S	ALL	
REQ.NFN.06	Non-functional	Reliability	N/A	S	ALL	
REQ.NFN.07	Non-functional	Power efficient & Hybrid electrically powered devices	N/A	S	REQ.US.01.01 - REQ.US.01.12, REQ.US.02.01, REQ.US.02.08, REQ.US.02.11, REQ.US.02.12	
REQ.NFN.08	Non-functional	Accuracy	N/A	S	REQ.US.01.01 - REQ.US.01.12, REQ.US.02.01, REQ.US.02.08, REQ.US.02.11, REQ.US.02.12	
REQ.NFN.09	Non-functional	Rapid testing	N/A	S	REQ.US.01.01 - REQ.US.01.12, REQ.US.02.01, REQ.US.02.08, REQ.US.02.11, REQ.US.02.12	
REQ.NFN.10	Non-functional	Durability and ruggedness	N/A	S	REQ.US.01.01 - REQ.US.01.12, REQ.US.02.01, REQ.US.02.08, REQ.US.02.11, REQ.US.02.12	

2.3 Technical specifications

This subsection provides the Technical specifications and guidelines about security, transparency, and performance and describes specific implementation constraints related to the system. This subsection, in combination with sections 3, 5, 6 and 7 form the set of technical specifications that the ADV platform and its components need to implement and be compliant with.

2.3.1 General technical specifications

In this section, we describe some general specifications that need to be met by the ADV platform's components as well as by the ADV platform as a whole:

- The equipment and software products that are part of the solution will be designed and built using up-to-date concepts and technologies, in accordance with the current trends on the market.
- The component equipment of the proposed technical solution will be modular, scalable and will allow upgrade operations, if necessary.
- The proposed solution will allow the online execution of the administration procedures of the component systems, without the users' activity being affected.

2.3.2 Security of the system

This section presents the security and privacy specification guidelines for the system and its users. ADV's approach on handling security is to review risks and vulnerabilities at every stage of the development life cycle, paying particular attention to the design of a solid authentication and authorization strategy, while taking into account that the majority of application-level attacks rely on maliciously formed input data and poor application input validation.

In the following we will present a set of design guidelines for the security requirements that will ensure the development of a high security system for the ADV project.

Authentication is the process of determining the identity of users or, in the context of application interoperability, the identity of the calling applications or processes. Poor design and implementation of authentication can lead to identity spoofing, password cracking, privilege escalation or unauthorized access to sensitive resources. To design secure authentication features for a web application, it is recommended to divide the application into anonymous, identified, and authenticated zones. The use of strong passwords and password expiration periods and account deactivation are also required.

To design a strong authentication for our system, there are some aspects to consider:

- It is necessary to identify where authentication is required in the application.
- It is necessary to validate who is calling (authenticate).

The ADV platform integrates heterogeneous technologies and components, so there is a need to establish an authentication method suitable for machine-to-machine communication to secure the communication between various components. Machine-to-machine authentication works on a similar principle as requiring users to enter their usernames and passwords in order to access a system. However, machine-to-machine authentication mandates that programs obtain an access token from an authorization system before they can access server data, in place of these user credentials.

There are typically three steps in the procedure.

- The client, which could be an application, a process, or any other system, contacts the authorization server with a request. The audience, client secret, and client ID are all included in the request.
- Following the request's validation, the authorization server replies with an access token, a seemingly random string of characters that denotes the client's permission to access the requested data.
- The client uses the access token to ask for access to particular server-stored data.

This machine-to-machine authentication solution will use Open Authorization 2.0 (OAuth) protocol as a way for applications to access different components of the system. OAuth 2.0 is the current standard protocol for online authorization.

Users usually log in with usernames and passwords. The external application should have authentication credentials as well (it is necessary to identify the user or the application in the subsequent requests). This can be done by using some form of token or cookie authentication.

Authorization determines which resources the authenticated identity has access to and what it can do. Inadequate authorization can often lead to data manipulation and the execution of unauthorized operations, as well as to information disclosure by allowing access to confidential or restricted data. In order to prevent the situations detailed above, it is recommended to use the least privileged accounts, where each system component or process should have the minimum rights necessary to perform its duties. This can reduce the "attack surface" of your computer by removing unnecessary privileges that can lead to network exploits and computer compromise. An important aspect to consider is the granularity of authorization, where security needs may require more detail regarding the authorization context. Restricting user access to system-wide resources by disabling unnecessary system features, services, and ports is a good principle for security design.

In addition to the above, the communication channel must be secured, and the content of the authentication cookies must be encrypted. Finally, all requests received from system components will be logged.

2.3.3 Transparency of the system

This section focuses on the ADV's independence from physical location. That can be interpreted in that the ADV platform will be a "location independent" platform, so that a custom, end-user application should be able to access the cloud service ubiquitously and responsively regardless of its location.

Also, end users do not need to know or care about the physical location (of course, with respect to the GDPR regulations) of the data that flow in the platform's components; data is retrieved without any specific reference to the physical sites.

2.3.4 Performance of the system

This section presents some basic principles to be followed by the ADV platform and its components with regards to user concurrency and application response times.

A system is considered non-functional if it does not function at, or beyond, acceptable performance levels. The system will not even be used at all if its performance is truly unacceptable for a significant number of users. The identification and quantification of performance requirements is part of the overall capacity planning process. The scenarios for which the performance is characterized must be identified before actually determining the desired performance characteristics of the system. Performance objectives must be specified clearly and unambiguously.

To assess the performance of a system the following aspects must be clearly specified:

- Response Time
- Workload

Users' perception of the overall experience of the system is influenced by its response time. Determining what system users consider "acceptable" in terms of performance is difficult enough. In most cases, the system response times are clearly identified as part of a generic use case.

Table 2: Process Types Classification for Generic Use Cases

Technical Specification No / Name	***Process Type***
TS1 - Ingest historic/operational batch data	*long-running*
TS2 - Enrich collected data	*long-running*
TS3 - Analyse and visualize collected/enriched data	*high-impact*
TS4 - Configure and receive alerts	*high-impact*

Existing processes or planned use cases (listed in Table 2) should be the starting point for defining the workload. The workload must be specified to be supported by the planned hardware capacity without the need for upgrades, and thus will define the work that the system must support. The performance of the system is influenced by the requirements derived from the workflows and transactions initiated by the user. These requirements must be considered by all system users as well as automated processes. In the table below, for a set of relevant usage scenarios, a scale is proposed to quantify the daily workloads and the corresponding response times.

Table 3: Response Time Quantification for Specific Usage Scenarios

Usage Scenario	***Daily Workload***	***Response Time***
User authentication	*1,000*	*2s*
Scheduled nightly backup	*1*	-

It is necessary to specify that the response time in Table 3 quantifies the average time for displaying a page/screen following the user's action. If more than one action is required on a page, the estimated time is not cumulative for all actions, but only counts the response time between actions. Also, time to fill in form fields is not considered. The performance of the system also depends on how the workload is distributed over the operating hours; therefore, it is important that the system workload profile is defined. Another important factor related to the workload is the determination of the "peak" workload, which will define the minimum hardware configuration of the system to handle such a workload. All of the above system requirements are interrelated. For example, scalability can be simply specified as the increase in system workload that the system should be able to process. Scalability requirements are often determined by the lifetime and maturity of the system.

2.3.5 Resource constraints related to the infrastructure

According to the component owners' input, the infrastructure which will host the various components of the ADV platform should be capable of providing high-end performance in terms of CPU, plenty of RAM and Storage as well as to provide GPU for the FL and ML related processing. The above specifications were captured using the following table where H/W requirements from indicative components are provided:

Table 4: H/W requirements - infrastructure

Component / Submodule	Partner	RAM	CPU cores	Storage needs	GPU required	GPU type
SECURESTORE	SIEMENS	16GB	8	2TB until the end of the project	No	
Blockchain Traceability Solution	ALMA	8 GB	2x Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz (64 bits)	30 GB (SSD)	No	N/A
XAI Conceptual Framework	ALMA	256 GB	16x Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz (64 bits)	1 TB	Yes	1xTesla K80, compute 3.7, having 2496 CUDA cores, 12GB GDDR5 VRAM
DKM	ATOS	128 GB	128	512 GB	Recommended	N/A
FDML	ATOS	256 GB	128	1 TB	Yes	N/A

3 AgriDataValue Data Model and Semantic Interoperability

3.1 Data Model Overview

In the AgriDataValue project, data models and semantic interoperability play a crucial role in enabling efficient data exchange and communication between different systems and stakeholders in the agricultural sector. Data models provide a structured and standardized way to represent and organize agricultural data, making it easier to understand, share, and use. On the other hand, semantic interoperability ensures that the meaning of the data is preserved and understood consistently across different systems, platforms, and users. This is particularly important in the context of the AgriDataValue project, where diverse data sources and systems are involved.

The combined model is expected to serve as a common language for the AgriDataValue project, enabling different systems and devices to exchange data with unambiguous meaning. This will ensure that data can be integrated and used in a consistent and reliable way, regardless of the source of the data. It is also expected to provide a framework for semantic interoperability, ensuring that the meaning of the data is preserved when it is exchanged between different systems. This will ensure that the data can be understood and used correctly, regardless of the system or device that is processing the data.

The combined model is also expected to be extensible, allowing for the addition of new concepts and elements as necessary. This will ensure that the model can evolve and adapt to the changing needs of the AgriDataValue project.

3.2 Semantic Interoperability in AgriDataValue

In AgriDataValue, semantic interoperability plays a crucial role in enabling seamless communication and data exchange among various agricultural systems and stakeholders. As such, it is considered a very important technical specification for the ADV platform. By adhering to the IDS (International Data Spaces) standard and the GAIA-X Trust Framework[2], AgriDataValue ensures compatibility and trustworthiness in semantic interoperability.

Semantic interoperability within AgriDataValue will be achieved by leveraging the guidelines and specifications provided by the International Data Spaces (IDS) standard, the GAIA-X Trust Framework, and the Agriculture Information Model (AIM) from the DEMETER project[3]. This combination of frameworks establishes a technical foundation for seamless and trustworthy data exchange in the agriculture domain.

The IDS Information Model forms the core of AgriDataValue 's semantic interoperability approach, providing a common framework and data model for representing agricultural data. It defines shared ontologies, metadata schemas, and concepts on a domain-agnostic level that capture relevant information in a standardized manner.

[2] https://gaia-x.gitlab.io/policy-rules-committee/trust-framework/gaia-x_trust_framework/
[3] https://h2020-demeter.eu/technical-components/#aim

By aligning with the IDS Information Model[4], AgriDataValue enables meaningful integration and analysis of data from diverse agricultural sources.

In parallel, AgriDataValue embraces the GAIA-X Trust Framework to ensure data sovereignty, security, and privacy in the agriculture domain. It adheres to the technical and organizational guidelines provided by the framework, implementing data anonymization techniques and access control mechanisms. These measures protect sensitive agricultural data and comply with data protection regulations, fostering trust among data providers and consumers within the AgriDataValue ecosystem.

To further enhance semantic interoperability (on a sector-specific level), AgriDataValue explores the integration of the Agriculture Information Model (AIM)[5] from the DEMETER project. The AIM focuses specifically on agricultural data and provides a standardized representation across the agriculture value chain. By combining the IDS Information Model and the AIM, AgriDataValue aims to achieve a comprehensive and harmonized representation of agricultural data, as well as align with the standards of the initiatives mentioned.

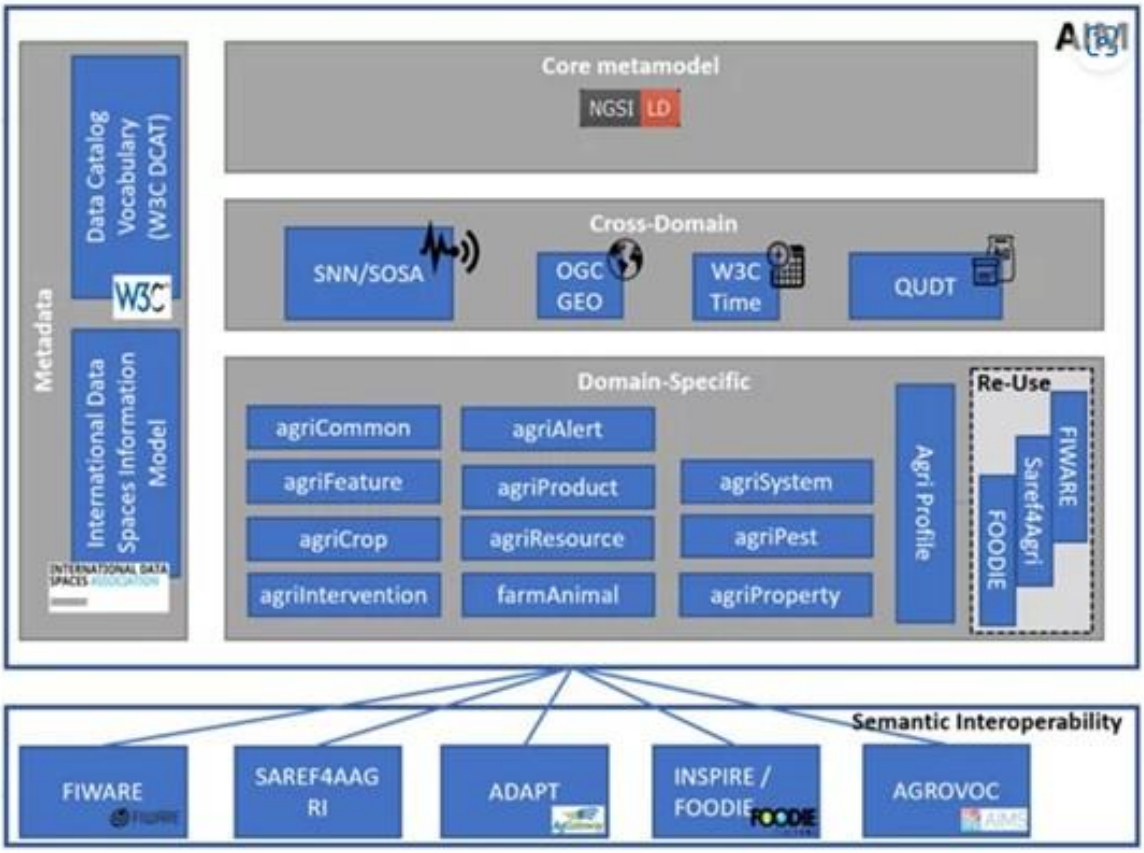

Figure 1: Agriculture Information System (AIM)

[4] https://international-data-spaces-association.github.io/InformationModel/docs/index.html
[5] https://github.com/rapw3k/DEMETER

Semantic interoperability within the International Data Spaces (IDS) framework is facilitated by the IDS Reference Architecture Model (IDS-RAM) and the utilization of vocabularies. The IDS-RAM provides a comprehensive view of the structure and concepts within a Data Space, employing a layered approach to describe various concepts.

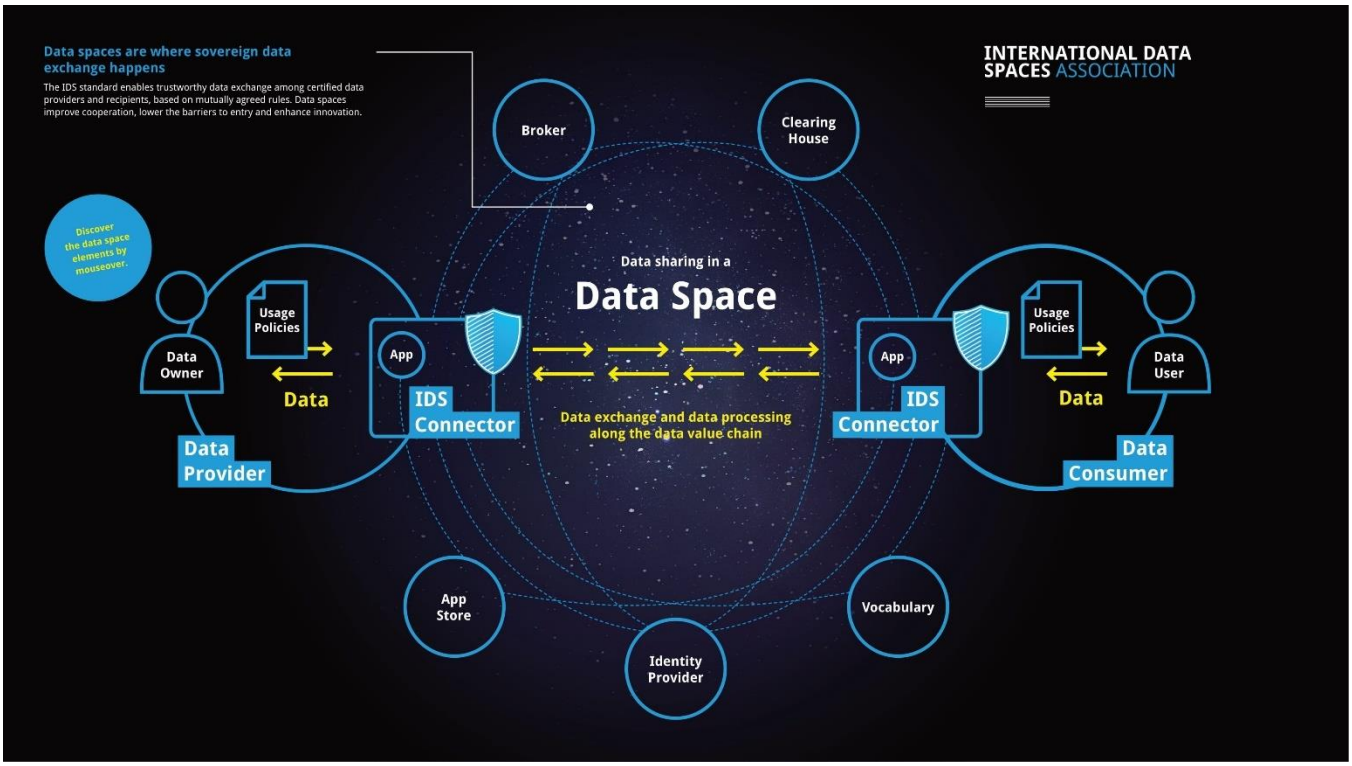

Figure 2: Infographic showing roles in an IDS-compliant dataspace

Vocabularies, such as ontologies and metadata elements, annotate and describe data assets within IDS, including the Information Model and domain-specific vocabularies. The Vocabulary Intermediary manages and provides these vocabularies, which are owned by standardization organizations. Multiple vocabularies can describe the same context, promoting standardization and flexibility.

Application Profiles adapt the Information Model for specific IDS ecosystems, while independent domain-specific vocabularies describe resource content and concepts. The Vocabulary Hub supports vocabulary management throughout the lifecycle, allowing data providers to link semantic descriptions to their data offerings. The IDS Metadata Broker references vocabularies and Vocabulary Hubs during the Runtime Phase.

Data consumers verify vocabulary compatibility when querying an IDS Metadata Broker or a Data Providers Connector. If incompatible, they may request data transformation, implement necessary interfaces, or choose a different provider. Validation of provided schemas is recommended before initiating contract negotiation.

For more details, refer to section 3.4 of the IDS-RAM[6].

[6] https://docs.internationaldataspaces.org/ids-ram-4/

Through the integration of these models, AgriDataValue seeks to bridge the gap between cross-industry and domain-specific data elements. It identifies common concepts, establishes mappings, and harmonizes representations to enable seamless data exchange and interoperability between systems adhering to either model. This integrated approach provides a unified view of agricultural data, encompassing both general industry-wide elements and domain-specific agricultural aspects.

In the table below, we have listed the semantic interoperability mechanisms that could be used in the AgriDataValue project, and the consortium already decided to start with DEMETER AIM (that already covers most of the models listed below) and then will integrate other mechanisms such as IDS Information Model (which is a domain-agnostic information model) to ensure cross-domain interoperability and standardization.

Table 5: Semantic Interoperability Mechanisms for AgriDataValue

Name	Purpose	Type
DEMETER Agriculture Information Model (AIM)	To provide a common framework for the representation of agricultural data, enabling the integration of data from different sources and facilitating the interoperability of different agricultural systems.	Data Model Framework
FIWARE AgriFood Data Model	To provide a standardized way of representing agricultural data, facilitating the integration of data from different sources and enabling the development of innovative agricultural applications. (DEMETER AIM has semantic mapping to this technology)	Data Model Framework
ADAPT	To provide a common framework for the representation of agricultural data, enabling the integration of data from different sources and facilitating the interoperability of different agricultural systems. (DEMETER AIM has semantic mapping to this technology)	Data Model Framework
SAREF4AGRI	To provide a common semantic model for the representation of agricultural data, facilitating the integration of data from different sources and enabling the interoperability of different agricultural systems. (DEMETER AIM has semantic mapping to this technology)	Ontology
IDS Information Model	To provide a common framework for the representation of data, enabling the integration of data from different sources and facilitating the interoperability of different systems.	Data Model Framework
AGROVOC	To ensure consistent terminology and facilitate data retrieval and organization across different agricultural datasets. (DEMETER AIM has semantic mapping to this technology)	Controlled Vocabulary

INSPIRE/FOODIE	To ensure semantic interoperability by providing a common framework for the exchange of information related to organic food products. This will facilitate the integration of data from different sources and enable the efficient management of agricultural operations. (DEMETER AIM has semantic mapping to this technology)	Data Model Framework

Table 6: Semantic Interoperability Mechanisms for AgriDataValue

- As mentioned on the table, AIM has semantic mappings to several well-known ontologies and systems, including FIWARE AgriFood, SAREF4AGRI, ADAPT, INSPIRE/FOODIE, and AGROVOC. This means that the concepts and terms in these systems are aligned with the concepts and terms in the AIM, enabling interoperability between these systems.

By integrating these systems into the combined model, the AgriDataValue project can leverage a wide range of data and capabilities, enhancing the value and utility of the combined model.

3.3 Integration of DEMETER AIM and IDS Information Model

We will take a 5-step approach for integrating the chosen semantic interoperability mechanisms:

1) Understanding the Models: The first step is to thoroughly understand both models. This involves studying the classes, properties, and relationships of each model, and understanding the semantics and context of each element. This step might involve consulting with domain experts or referring to documentation and use cases.

2) Identifying Common Elements: The next step is to identify common elements between the models. This could be classes that represent the same concept, properties that have the same meaning, or relationships that have the same semantics. These common elements will form the basis of the integration.

3) Mapping Elements: After identifying the common elements, the next step is to map the elements from one model to the other. This involves creating a mapping table that shows how each class, property, or relationship in the AIM corresponds to an element in the IDS Information Model. This mapping table will guide the integration process.

4) Implementing the Integration: The final step is to implement the integration based on the mapping table. This will involve creating a new model that combines elements from both models, or alternatively using a translation layer that converts data from one model to the other. This step will likely involve software development and testing to ensure the integration works according to the expectations.

3.3.1 Understanding the Models

IDS Information Model

IDS Information Model serves as the universal language (lingua franca) of the International Data Spaces, establishing a central consensus that ensures compatibility and interoperability among its participants and components. Its primary objective is to facilitate the description, publication, and discovery of data products and

data processing software within the International Data Spaces. Through structured, semantic annotation, it ensures that the most relevant and suitable assets are provided for a client's task. Once these assets are identified, the Information Model enables their automated consumption through service interface and protocol binding definitions. In addition to these core assets, the Information Model also describes key properties of International Data Spaces entities, participants, infrastructure components, and processes.

The Information Model's main goal is to describe, publish, and identify data products (Data Assets) and reusable data processing software (Data Apps) within the Industrial Data Space. These Data Assets and Data Apps are the core resources of the International Data Spaces and are collectively referred to as resources. Through structured semantic annotation, the model ensures that only relevant and suitable resources are provided to meet the requirements of the Data Consumer. Once these resources are identified, they can be exchanged and consumed in an automated manner through semantically defined service interfaces and protocol bindings. Beyond these core commodities, the Information Model also outlines essential properties of Industrial Data Space entities, participants, infrastructure components, and processes.

The IDS Information Model is defined using RDF (Resource Description Framework) and can be serialized in various formats such as JSON-LD and Turtle.

Agriculture Information Model

The Agriculture Information Model (AIM) is a common vocabulary developed under the DEMETER project, providing the foundation for semantic interoperability across smart farming solutions. It is a domain-specific model designed to capture and represent information in the agriculture sector. The AIM is publicly accessible and is distributed under the Creative Commons Attribution 4.0 License.

The AIM is structured into several specific profiles or modules, each addressing a particular aspect of the agriculture domain. These include modules for properties, products, interventions, pests, common elements, systems, crops, farm animals, alerts, and features. This modular structure allows for a comprehensive and detailed representation of various aspects of agriculture, from crop management to animal husbandry and alert systems.

The AIM is represented in the OWL (Ontology Web Language) format, which is a standard Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language, which means that the concepts defined in it can be processed by computers. In addition to OWL, AIM also supports the RDF (Resource Description Framework) format. RDF is a standard model for data interchange on the Web, which provides interoperability between applications that exchange machine-understandable information. The use of these standard formats ensures that the AIM can be easily integrated with other data models and systems, further enhancing its utility in the agriculture sector.

The IDS Information Model and AIM complement each other in several ways:

- **Generic vs. Domain-Specific**: The IDS Information Model provides a generic framework that can be applied across various domains. It includes concepts like data assets, participants, contracts, and messages. On the other hand, AIM is specifically designed for the agricultural sector, with classes like AgriParcel, Crop, Alert, and Intervention. This combination allows AgriDataValue to leverage the broad applicability of IDS while also addressing the specific needs of the agricultural sector with AIM.

- **Data Sovereignty and Security**: IDS excels in ensuring data sovereignty and secure data exchange, which are crucial for protecting sensitive agricultural data. AIM, while not explicitly designed for data security, benefits from the secure framework provided by IDS.
- **Interoperability**: Both models promote interoperability, which is key for Task 1.3. IDS does this through a standardized framework that can be extended with domain-specific models like AIM. AIM, with its specific agricultural classes, can easily be mapped to the IDS classes, enhancing interoperability within the agricultural domain.

In the context of AgriDataValue, the integration of IDS and AIM can fulfil the requirements of Task 1.3 in the following ways:

- **Data Modelling**: The combined model provides a comprehensive framework for representing agricultural data. The domain-specific classes of AIM can be used to model agricultural data, while the generic classes of IDS can be used to model other necessary data elements.
- **Data Exchange**: The secure data exchange mechanisms of IDS can be applied to the agricultural data represented by AIM, ensuring secure and sovereign data exchange within the AgriDataValue project.
- **Interoperability**: The mapping between IDS and AIM enhances interoperability, as data represented in one model can be understood in the context of the other. This is particularly important for Task 1.3, which requires the development of interoperable data models.

3.3.2 Identifying Common Elements

The International Data Spaces (IDS) Information Model and the Agriculture Information Model (AIM) can be considered as two complementary models, each providing unique and essential elements for the AgriDataValue project. While the IDS Information Model provides a generic framework for data sovereignty and secure data exchange, the AIM offers domain-specific models tailored for the agricultural sector. The integration of these two models can significantly enhance the data management capabilities of the AgriDataValue project, particularly in Task 1.3, which focuses on data modeling and interoperability.

IDS Information Model Class	DEMETER-AIM Class	Common Properties	Common/Similar Definitions
Connector	System	id, description	a) IDS: A system of endpoints enabling different types of communication. b) AIM: A system is a set of correlated network elements. c) Common: A system or connector is a set of related elements or endpoints enabling communication.
Participant	Profile	id, name	a) IDS: A participant in a business process or use case. b) AIM: A profile is a set of correlated properties of an entity. c) Common: A participant or profile is a set of properties or characteristics related to an entity or process.
Resource	AgriParcelRecord	id, description	a) IDS: A resource is an identifiable component of a system. b) AIM: An AgriParcelRecord is a record of a specific agricultural parcel. c) Common: A resource or AgriParcelRecord is an

			identifiable component or record within a system or agricultural context.
Asset	CropType, AgriPest, Animal	id, name	a) IDS: An asset is a valuable item, entity, or property. b) AIM: CropType, AgriPest, and Animal are specific types of assets in an agricultural context. c) Common: An asset is a valuable item, entity, or property, which can be specific types such as CropType, AgriPest, or Animal in an agricultural context.
Message	Alert	id, issued	a) IDS: A message is a discrete unit of communication intended by the source for consumption by some recipient. b) AIM: An alert is a specific type of message indicating a potential issue or condition. c) Common: A message or alert is a unit of communication, potentially indicating a specific issue or condition.

Table 7: Common Elements of AIM and IDS Information Model

This table provides a starting point for integrating the two models. By identifying corresponding classes and their common properties, we can create a unified model that leverages the strengths of both IDS and AIM.

The IDS Information Model is quite extensive and includes a wide range of classes and properties. These classes from the IDS Information Model could be mapped to the Agriculture Information Model (AIM) as seen in the table below.

Table 8: An initial mapping of IDS InformationModel to AIM

Agriculture Information Model	IDS Information Model	Description
ActiveIngredients	Artifact	Both represent specific components or elements within a larger context (product in AIM, data in IDS).
ActivityComplex	Operation Result Message	Both relate to the results or outcomes of certain operations or activities.
AgriculturalContract	Contract	Both represent agreements or contracts within their respective domains.
AgriParcelRecord	Resource	Both represent a specific entity or item of interest within their respective domains.
AgriPest	Alert	Both represent notifications or alerts about certain conditions or situations.
Alert	Notification Message	Both represent messages or alerts that are sent out in response to certain conditions or situations.
CampaignType	Message	Both represent a type of communication or message within their respective domains.
Class	Entity	Both represent a general category or type of item within their respective domains.

Concept	DescribedSemantically	Both represent an idea or concept within their respective domains.
CropType	Resource	Both represent a specific entity or item of interest within their respective domains.
DoseUnit	Measurement	Both represent a unit of measurement within their respective domains.
Entity	Entity	Both represent a specific entity or item of interest within their respective domains.
ID	Identifier	Both represent a unique identifier for an item or entity within their respective domains.
ProductionType	Resource	Both represent a specific entity or item of interest within their respective domains.
Property	Property	Both represent a characteristic or attribute of an item or entity within their respective domains.
PropertyType	Property	Both represent a characteristic or attribute of an item or entity within their respective domains.
Scheme	Protocol Specification	Both represent a plan or protocol for how something should be done within their respective domains.
taxonomic_rank	Classification	Both represent a method of classifying or categorizing items within their respective domains.

The properties of these classes would also need to be mapped in a similar manner. For example, the resourceEndpoint property in the IDS Information Model might correspond to the agriResourceLocation property in the AIM.

3.3.3 Mapping Elements

Table 9: Preliminary Mapping of Similar Elements

IDS Information Model Class	Agriculture Information Model Class	Description
Agent	Farmer, Agricultural Technician	The Agent class in IDS can be used to represent different roles in the agriculture domain such as Farmer or Agricultural Technician.
Artifact	Crop Yield Report, Soil Analysis Report	The Artifact class in IDS can represent various reports or documents in the agriculture domain.
Event	Harvesting, Planting	The Event class in IDS can represent various activities or events in the agriculture domain.
Process	Irrigation, Fertilization	The Process class in IDS can represent various processes in the agriculture domain such as Irrigation or Fertilization.

Resource	Crop, Livestock	The Resource class in IDS can represent various resources in the agriculture domain such as Crop or Livestock.
Policy	Crop Rotation Policy, Organic Farming Policy	The Policy class in IDS can represent various policies in the agriculture domain.
Data	Weather Data, Soil Data	The Data class in IDS can represent various types of data in the agriculture domain.
Representation	ActiveIngredients, CropType	The Representation class in IDS can represent various types of data in the agriculture domain.
Contract	CampaignType, ProductionType	The Contract class in IDS can represent CampaignType and ProductionType from agriculture domain.
Participant	Entity, Scheme	The Participant class in IDS can represent Entity and Scheme from AIM.

This table has been created as an initial step towards achieving semantic interoperability between the Agriculture Information Model (AIM) and the International Data Spaces (IDS) Information Model. The IDS Information Model is domain-agnostic, meaning it is designed to be applicable across various sectors. On the other hand, the AIM is sector-specific, focusing on the unique needs and characteristics of the agriculture sector. By mapping the classes from the AIM to the IDS Information Model, we will align the specific agricultural concepts with the broader, domain-independent structure of the IDS model. This alignment is essential for enabling effective communication and data exchange between different systems and sectors.

Furthermore, this mapping table contributes significantly to the process of mapping elements between different data models. It provides a clear and concise overview of how each class in the AIM corresponds to an element in the IDS Information Model. This overview will serve as a starting point during the integration process, helping to identify which elements need to be transformed or converted to ensure compatibility.

4 Architecture Design Methodology

In ADV, services and components are represented using an ICT architecture that is cross-domain and is based on the ISO/IEC/IEEE 42010[7] International Standard. The ISO and IEEE collaborated to revise the earlier IEEE Standard 1471-2000, resulting in the publication of this Standard, "Systems and Software Engineering - Architecture Description" in 2011. A conceptual model of architecture description and recommended practices for the same were offered in the first edition. A requirement for architecture frameworks and architecture description languages is added to the current edition, which improves the original.

The aforementioned ISO Standard is built on a meta model of the terms and ideas related to an architecture description that is expressed using UML class diagrams to depict classes of things and their relationships. It essentially covers the results of an architecting process without offering any guidance on how to build one.

The norms, rules, and practices for the description of architectures that have been established within a particular area of an application and community of stakeholders are referred to in this Standard as an architecture framework. To define a shared set of architectural principles within a community for the purposes of comprehension, commonality, synergy, and interoperability is the architecture framework's core objective. The basic requirements for any framework that are described in terms of the conceptual model of architecture are likewise defined by this Standard:

1. Information identifying the architecture framework.
2. The identification of one or more stakeholders.
3. The identification of one or more stakeholders' concerns.
4. One or more architecture viewpoints that frame those concerns.
5. Any correspondence rules.

Discussions regarding what is doable, difficult, expensive, etc. should be facilitated early on in the overall development process via a good architecting process. In actuality, it works best when done concurrently with requirements definition, systems analysis, and a set of requirements, producing both a set of needs and an architecture that satisfies those requirements. The system/software requirements (perhaps in draft form), the operational principles for the system/software, and the prospective stakeholders list will be the inputs to this architectural process. An architecture description that complies with the Standard, the findings from examining it and the architecture it describes, as well as potentially updated requirements that take alternative architectural choices into account, are the outputs.

ADV follows the four main steps in this architecting process:

1. **Stakeholder/Concern identification**. First, a list of all prospective stakeholders is provided, together with a description of their issues with the system's design and architecture. It will be improved and given to the stakeholders again for approval. It is preferable to identify requirements that must be addressed as part of this process in order to meet the needs of the stakeholders.

[7] ISO/IEC/IEEE 42010: http://www.iso-architecture.org/42010/index.html

2. **Viewpoint development**. The next step is to determine the best way to respond to the issues raised by the stakeholders. This has two components: what the solutions are (the views) and how they can be obtained (the viewpoints). The Standard explicitly distinguishes between the two, allowing for the creation of reusable views. Capturing architectural decisions should be made simpler by a clearly defined set of viewpoints that have been reviewed by stakeholders and developers.
3. **View development**. Depending on what each viewpoint demands, this. It is crucial to document the key decisions' justifications while the perspective is being produced and to incorporate them in the architecture description. Although we do not explicitly distinguish between views and viewpoints in this work, we will keep this methodology in mind as we explain the many aspects of the architecture.
4. **View integration and evaluation**. Each viewpoint should be reevaluated as part of the architecture's integration, including any rules governing cross-view consistency. Each view's implementation of the viewpoint and the extent to which its contents cover the system as a whole from that view's point of view must be independently verified. The architecture being described should be evaluated in some way in relation to the worries of the stakeholders and other relevant factors. The internal review procedure used by ADV to address this will correct any discrepancies in the initial viewpoint descriptions.

Applying ISO/IEC/IEEE 42010:2011 approach for the architecture description of systems as ADV itself entails defining the architectural viewpoints that answer stakeholders' concerns, drafting their requirements, and developing uniform architectural views using architectural models. The viewpoints used to best extract ADV's correspondent architectural views are briefly described below:

1. **Context viewpoint** (which can be presented via a **process viewpoint** as well as a **high-level viewpoint**). To describe interactions, relationships and dependencies between the system and its environment which will interact with the system itself, other systems, users, and developers.
2. **Information viewpoint** (usually presented via a **data viewpoint**). To describe data models, data flows, and how this data is manipulated and stored. The main goal of this is to provide a common communication language between stakeholders, domain and data experts.
3. **Functional viewpoint**. To describe the main functional elements of the architecture, interfaces and interactions.
4. **Deployment viewpoint**. To describe how and where the system is deployed, considering hardware and physical dependencies. It provides consistent mapping across the existing and emerging technologies and the functional components specified in the *Function View*.

It is vital to have a list of both required and optional questions to be answered in order to record the architecture opinions. The following are some of the issues or subjects that require discussion:

1. Viewpoint name.
2. Viewpoint overview: a brief overview of this view and the information it presents as well as its key features or a high-level view of its operations.
3. Typical stakeholders: a list of the stakeholders expected to be users of views using this viewpoint.
4. Model kinds or diagrams: Describe each model type that the viewpoint specifies; alternatively, provide a diagram outlining the components rather than a model. The International Standard does not prescribe a particular style for documenting these; instead, it allows for a variety of approaches, including the use of a UML diagram, a meta model, a model template, a language specification, or a combination of these.

Depending on the perspective, the following may be added to the previously mentioned necessary information

that must be provided:

1. Concerns and "anti-concerns": a list of the architecture-related concerns to be framed by this viewpoint that help to decide whether this viewpoint will be useful for a particular system of interest.
2. Correspondence rules: rules defined by this viewpoint or by its model kinds.
3. Operations on views: methods to be applied to views or to their model kinds.
4. Examples for the reader.
5. Notes: Any additional relevant information.
6. Sources: Identify the sources for this viewpoint (if any) including e.g., references

The creation and application of the ISO Standard have taught us some very significant lessons. They can be grouped together as falling under the following categories:

1. **Ontology-based**. On top of an explicit conceptual model or ontology, the Standard is constructed. An architecture framework must be useable in order to be effective, which means it must be intelligible and presented in a way that can be used. A precise conceptual base and terminology expressing it are the first component of understandability.

2. **Interest-driven**. Complex entities have a multitude of interested stakeholders, each one with specific interests or concerns that, once identified, can serve as a key index to a successful architecture description.
3. **Open and extensible**. Architecture frameworks should be designed to be open and extensible as a system. One important lesson learned from architecture framework development is that defining the ontology of a given domain of interest will never be finished.
4. **Framework as a foundation**. A robust architecture framework can serve as a foundation for all aspects of architecting beyond its central role in architecture description. This means that a framework has implications for methods, processes, and tooling.
5. **Governance**. Robust conformance is a key means to achieving the usability and interoperability of architecture descriptions and should support extension, reusable model kinds and methods, as well as end-product architecture descriptions. Currently, conformance is defined in terms of meta model consistency.

Following the methodology described above, we present the different viewpoints of the ADV Reference Architecture in sections 5 and 6 below. The stakeholders list along with their needs (expressed as user requirements) are already identified in deliverable D1.1 (section 2.1 and section 7.1). The user requirements are translated to technical specifications of the system and mapped to technical requirements in section 2 below. Finally, in addition to the methodology described above, it is necessary to mention the influence of the DEMETER[8] H2020 project to the AgriDataValue project and the architecture approach followed.

[8] https://h2020-demeter.eu/

5 AgriDataValue Reference Architecture

This section presents the ADV Reference Architecture (RA) as it has evolved from the beginning of the project. It has also been influenced by the numerous discussions held during the first period of the project, either in group telcos (e.g., biweekly Architecture telco since M4) and meetings or in peer-to-peer telcos where several aspects of the platform's components and functionality were discussed.

To that purpose, this section explores the relationships between the key ideas in the ADV RA from a variety of angles. First, we present a broad overview of the entire architecture. Subsection 5.2 which describes the primary elements of the ADV platform, then presents the functional viewpoint. The next deliverables of WP2 (such D2.1 and M12) will include comprehensive descriptions for each of these components. The process view of the architecture is then presented in subsection 5.3 , where it is explained how the components interact to deliver their fundamental functionalities. These are further detailed by also providing the sequence diagram for a hypothetical but representative scenario, followed by the ADV platform instances data exchange. The core data operations, including storage architecture, data retrieval, processing, and security management of the data transferred among ADV components, are highlighted in subsection 0 , which also provides more detail on the data perspective. Subsection 5.5 introduces the architecture's deployment viewpoint next. This relates to runtime operations and displays the connections between software components during program deployment as well as the topology of those components on the physical layer.

For all the views described in the subsections below, the following stakeholder categories will be considered (as described in section 2.1 of deliverable D1.1):

1. Farming companies, cooperatives, and individual farmers.
2. Farming (applied) and climate monitoring research institutes, including universities and scientists.
3. Specialized service and technology providers offering added valued services based on Agri-data and AgriDataValue technology.
4. CAP paying authorities.
5. EU policy makers.

5.1 High-level view

The ADV platform integrates heterogeneous technologies and components, while supporting fluid data exchange among them and addressing scalability. In this way, it offers a way for the integration of additional components, which could employ several different sensing and data collection and processing technologies.

The proposed method separates the visualization (end-user apps) layer from the processing (ADV backend services) layer, allowing additional data collection and/or backed components to be added in the processing and data collection layers (potentially collecting and processing data from new sources). This allows backend components to continue operating independently. In terms of usability, market acceptance, and sustainability, this is more feasible and realistic. Another objective is to make it easier for data from various sources and in various formats to be exchanged and interoperable, which is necessary to develop advanced applications. The following primary goals must be attained by the suggested remedy in order to realize this strategy:

- Supporting interoperability between various data formats that would enable the integration of existing

components, devices, and sensors.

- Extensive use of virtualization containers for services should be made to ensure rapid deployment, portability and scaling once required.

Figure 3 below depicts the High-Level view of the ADV RA. It consists of the data collection toolboxes at the bottom, which take care of the communication of the devices used for data collection from the data sources, the components and modules that form the ADV integrated platform, and a layer (IDS layer) offering the interoperability with external entities, end-user apps, etc. As depicted in the figure, the ADV integrated platform components are located either at the cloud (e.g., ADV cloud) or at the edge (e.g., node at the pilot premises). Finally, the need for utilizing an adaptor (i.e., ADV adaptor) to convert external data models to the ADV data model and vice versa is depicted using yellow triangles.

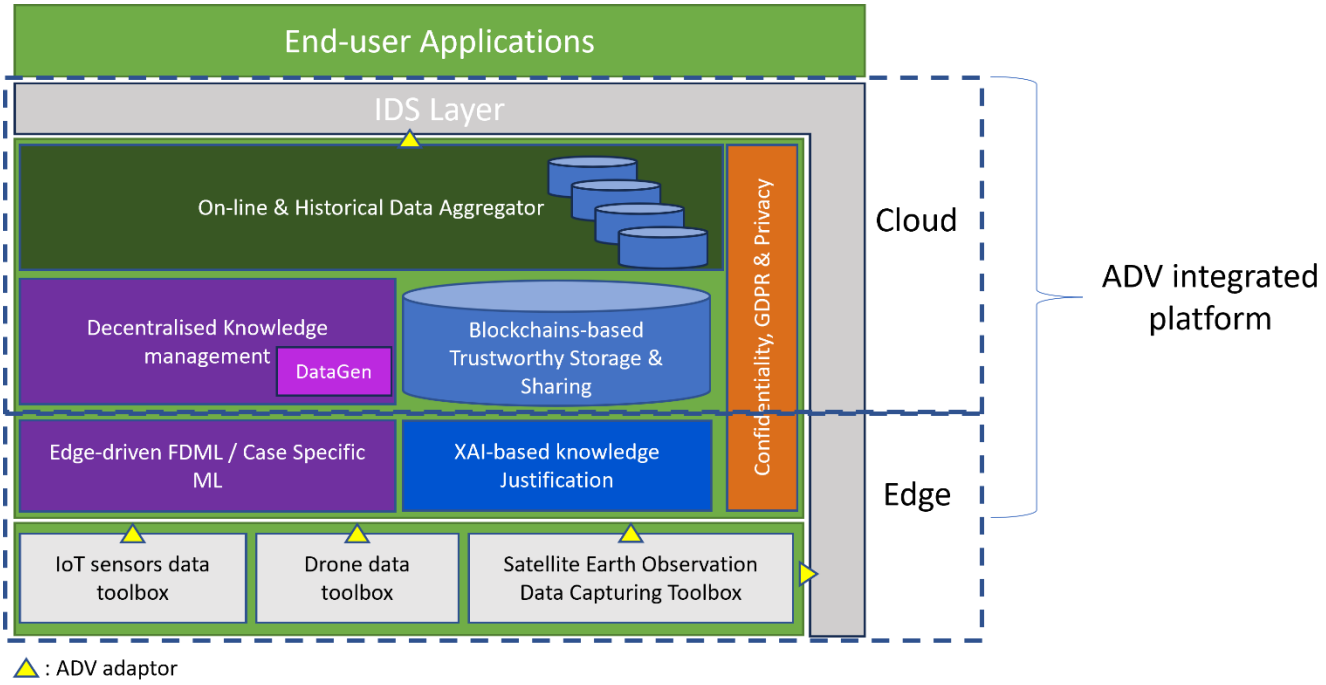

Figure 3: High-level reference architecture overview

5.2 Functional view

The Functional view of a system defines the architectural elements that deliver the system's functionality. The view describes the functional architecture of the system, outlining the major components, their roles, the interfaces they expose, and the relationships they have with one another. Together, this shows how the system will carry out the tasks that are required of it. Most architecture designs are built around the functional view, which also serves as the basis for defining the other architectural viewpoints. The information exchange among the platform components will use the ADV data model (depicted as dark green arrows in figures, JSON-based, described in section 3 and to be further detailed in upcoming deliverables or, if this is not possible, component-tailored data models (depicted as black arrows in figures); the colouring of the arrows captures the current status of the data model to be used, which can change in the course of the project and will be updated in the next version of the deliverable. Figure 4 depicts the functional view of the architecture.

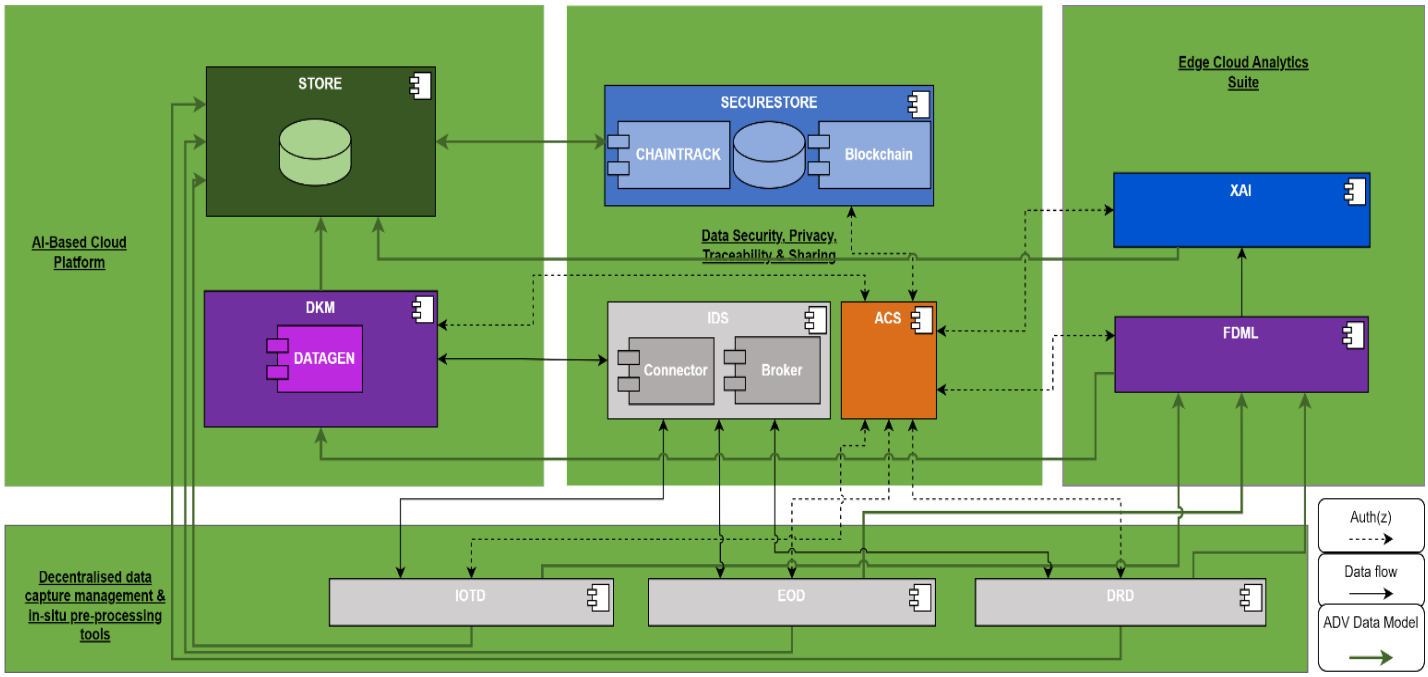

Figure 4: Functional view

In order to implement the high-level view of the architecture and to implement its objectives, ADV needs to provide several facilities/modules that interact with each other, with the various stakeholders as well as with existing devices and data sources. The main functional blocks of the ADV platform that constitute the functional view of its architecture are the following: a) Decentralised data capture management & in-situ pre-processing tools, b) Edge Cloud Analytics Suite, c) Data Security, Privacy, Traceability & Sharing, and d) AI-based Cloud platform. Each of these logical blocks consists of several components, which combined offer the expected functionality. Before we provide an overview of these functional components (detailed descriptions of these components will be included in the next WP2 deliverables, i.e., D2.1, D2.2, and D2.3), we need to link them to the high-level view described in the previous section. According to this view, the ADV integrated platform includes the Edge Cloud Analytics Suite, the Data Security, Privacy, Traceability & Sharing, and the AI-based Cloud platform blocks. The components of these blocks retrieve input data from various data resources (IoT sensor data, drone data, Earth observation data) through the components of the Decentralised data capture management & in-situ processing tools block. The data resources are ADV agnostic in the sense that they do not need to comply with the ADV data model in general. However, to be integrated into the ADV platform, each toolbox providing access to these data resources needs to be paired with an ADV data model adapter, which translates their data format to/from the ADV data model.

To provide a general overview of the components which the ADV platform blocks they consist of, the list of these components along with a short description of the functionality of each of these components (details will be included in D2.1) is provided in the sections below:

5.2.1 Decentralised data capture management & in-situ pre-processing tools

5.2.1.1 IoT sensors data toolbox (IOTD)

This component will provide access to IoT sensor data from sensors deployed on the field. This component will provide access to IoT sensor data from sensors deployed on the field. The technologies that will be used are

Django[9], Python[10], Celery[11], MinIO[12], GDAL[13]. The IOTD component will provide access to different types of sensors to remotely monitor in real-time climate/weather, radiation, soil and leaf conditions, ranging from air temperature and barometric pressure to soil moisture/ salinity, irrigation pipes network pressure/flow and drilling pressure/ flow/ operational efficiency, while additional meta-sensing indices (e.g. evapotranspiration, thermo-hours, dew point) and data analytics are calculated in-situ using edge cloud computational resources. Figure 5 below shows how data from these various sensors installed on-site is ingested into the toolbox and then, after the adaptation to the ADV data model, it becomes available to the ADV components through an API. The technologies that will be used are Django[14], Python[15], Celery[16], MinIO[17], GDAL[18].

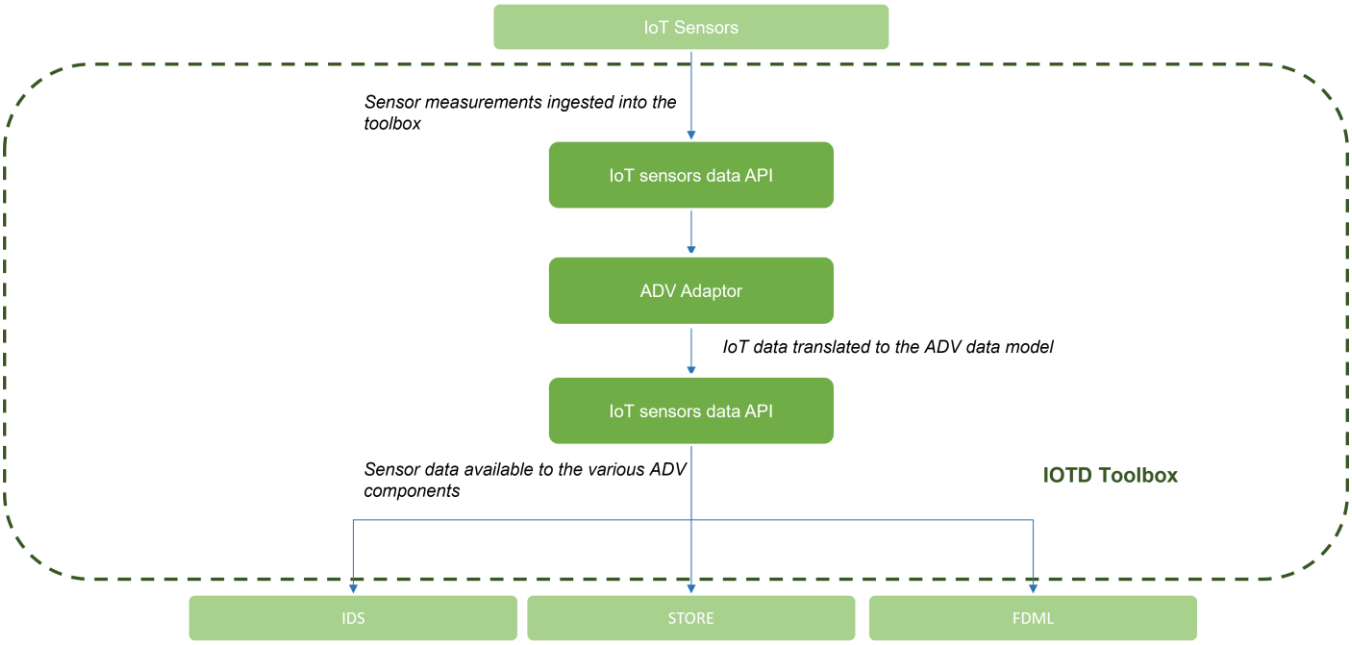

Figure 5: Logical view diagram – IOTD

5.2.1.2 Drone data toolbox (DRD)

This component will provide access to Drone data from drones deployed on the field. The technologies that will be used are Django, Python, Celery, MinIO, GDAL.

[9] https://www.djangoproject.com/
[10] https://www.python.org/
[11] https://docs.celeryq.dev/en/stable/getting-started/introduction.html
[12] https://min.io/
[13] https://gdal.org/
[14] https://www.djangoproject.com/
[15] https://www.python.org/
[16] https://docs.celeryq.dev/en/stable/getting-started/introduction.html
[17] https://min.io/
[18] https://gdal.org/

5.2.1.3 Satellite Earth Observation Data Capturing toolbox (EOD)

The component will provide access to Earth Observation data (Sentinel-2, Sentinel-1, Sentinel-5, Landsat etc.), aerial raster imagery created during the project and additional Copernicus (and other) EO datasets. The technology that will be used is the Sentinel Hub.

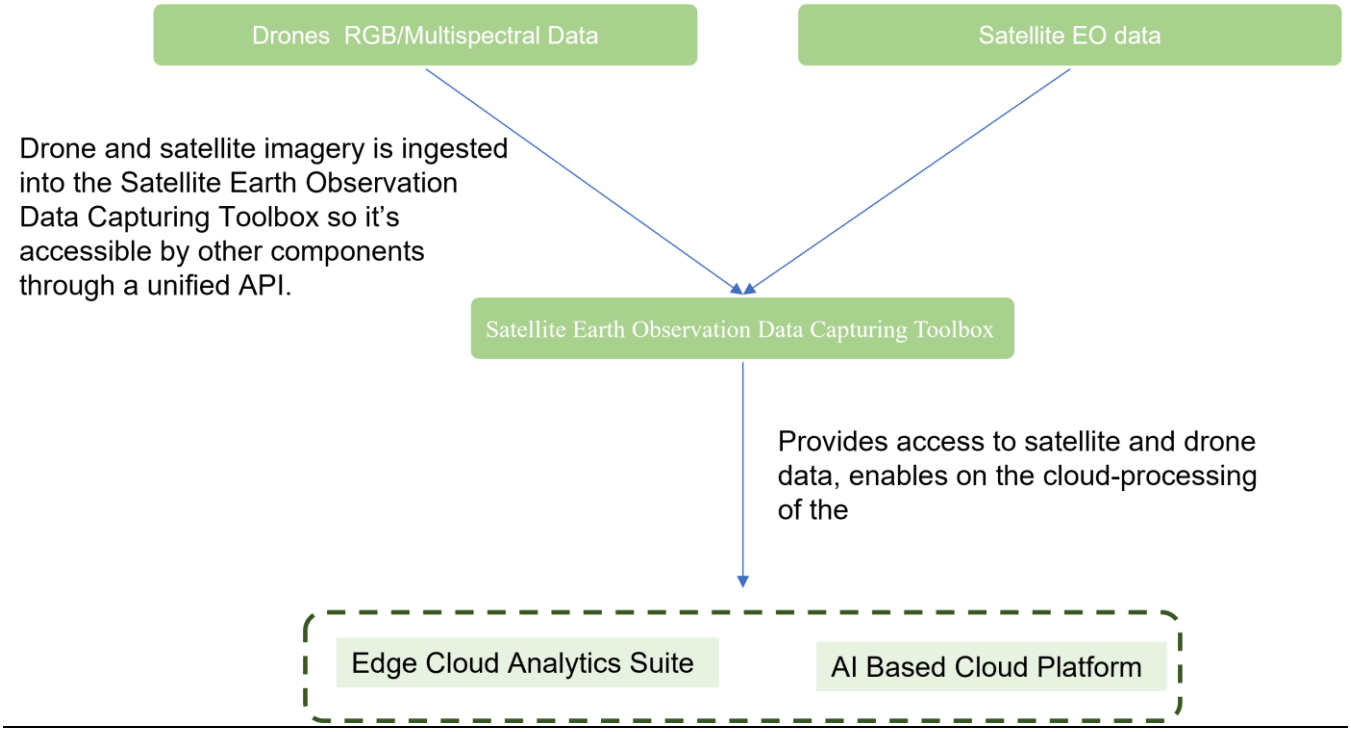

Figure 6: Logical view diagram - EOD/DRD

5.2.2 Edge Cloud Analytics Suite

5.2.2.1 Federated Machine Learning (FDML)

This module serves as a framework for implementing and deploying an agri-environment Decision Support System (DSS) using Federated Deep Learning and privacy-preserving techniques like PATE. By utilizing this module, it becomes possible to train multiple AI models in a federated manner, ensuring the preservation of data privacy and conducting inferences directly on the device. It is important to remark that the FDML component only encompasses the clients of the Federated Learning process, hence this component is completely located on the edge. An additional functionality of this component is the pre-processing step required for accommodating the uses cases' data to the Machine Learning and Deep Learning models that are going to be trained. The technology that will be used is ATOS' own Federated Learning framework. This component is shown in Figure *7* along with DKM.

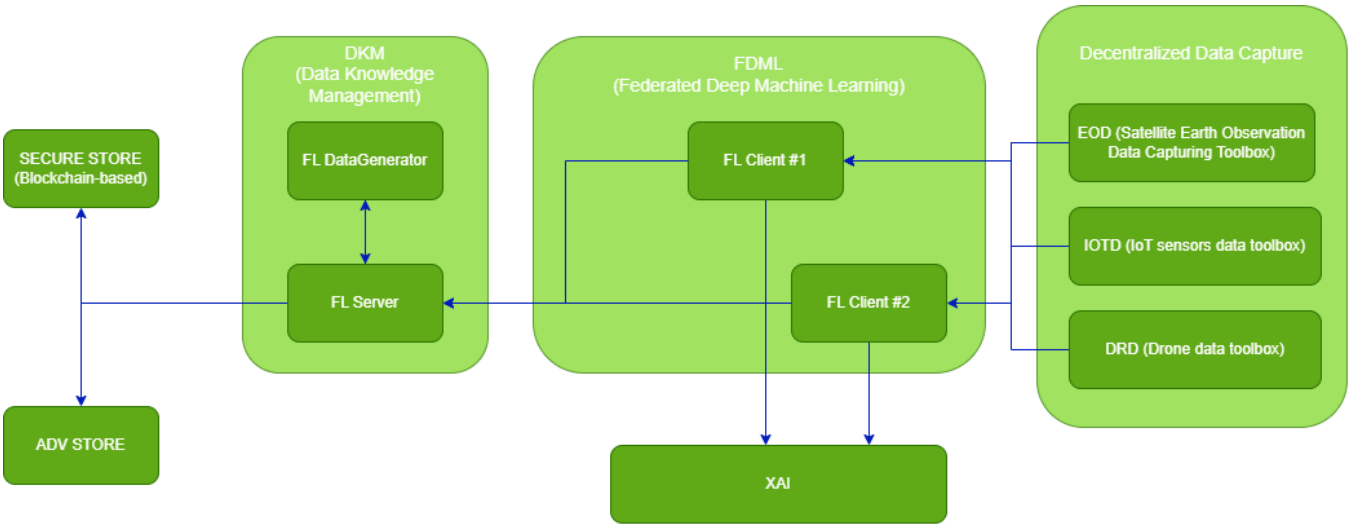

Figure 7: Logical view diagram - FDML & DKM

5.2.2.2 Human Explainable Conceptual Framework (XAI)

This component will enable transparent and interpretable AI to enhance the trust of end-users and advance the adoption of AI in the Agri-environment domain. The technology that will be used is Python.

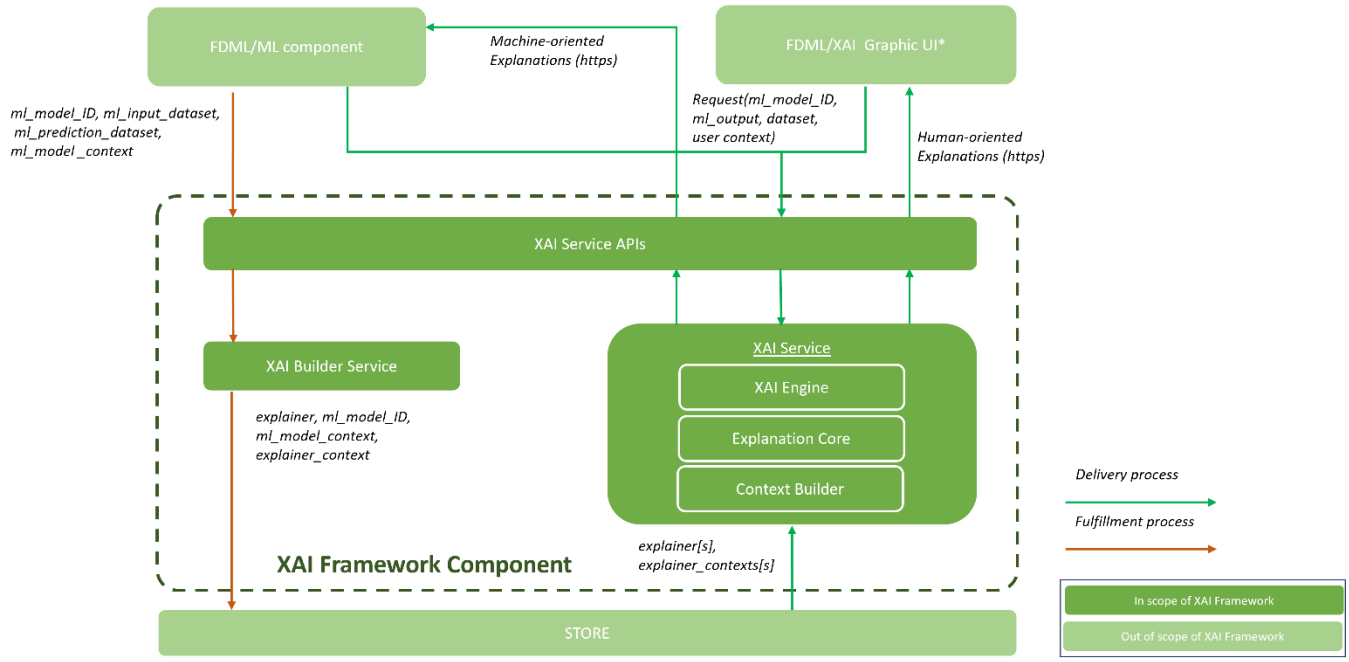

Figure 8: Logical view diagram – XAI

The picture displays at high level the logical architecture of the XAI framework solution, outlining the inner components used within

a) the delivery of explanation content to the end user (delivery process)

b) The production of the information content needed to enable the delivery (fulfilment process)

Both processes are activated by calls to a Service API layer that exposes the services to interfacing components

Fulfilment process:

- Each time a new ML model is trained, the FDML component sends the information needed to create a new explainer (i.e., ml*_model_ID, ml_input_dataset, ml_prediction_dataset, ml_model_context*).
- This information is processed by the XAI Builder Service to produce the corresponding explainer (trained XAI model). At the same time an explainer context is produced, summarizing the explainer characteristics.
- The explainer is then stored in the STORE associated to the ml_model ID.

Delivery process:

- Requests to the XAI service may come both from human users through UI (e.g., end-users, developers) and from programmatic access. The request shall contain reference to the used ml_model (ID) and the output, data sample and user context.
- The XAI Service retrieves from the STORE information about the ml_model context from the store and the matching explainer.
- Based on the explainer an explanation is built and returned

The underlying assumption of this solution is that the XAI framework component and the FDML Component maintain a shared taxonomy and inventory of ML Models- The ML model is produced in the FDML context, but needs to be known and accessible to the XAI framework through the STORE layer (e.g., in H5 and PKL format, as applicable)

5.2.3 Data Security, Privacy, Traceability & Sharing

5.2.3.1 Trustworthy Data and ML models storage and sharing (SECURESTORE)

The SECURESTORE component will handle increasing volumes of data from IoT sensors, drones, and EO hubs. It will support the integration of heterogeneous data sources, including different data formats, protocols, and standards. It will have mechanisms in place to ensure storage and sharing security, as well asl to transform, translate, and harmonize data into a common format for storage and analysis. In addition, data backup and recovery procedures will be implemented. The secure storage is achieved through the combined action of two sub-components: an object store for the actual conservation of the data and a blockchain network to certify in a non-disputable and immutable way the relevant steps of the data lifecycle. SECURESTORE will also allow storage and sharing of trained ML models and metadata coming from the FDML and XAI components. Moreover, this component will permit the storage of smart contracts using blockchain mechanisms. The technologies that will be used for the blockchain are Ethereum and Hyperledger Besu, while for the Storage, MinIO and Python will be used.

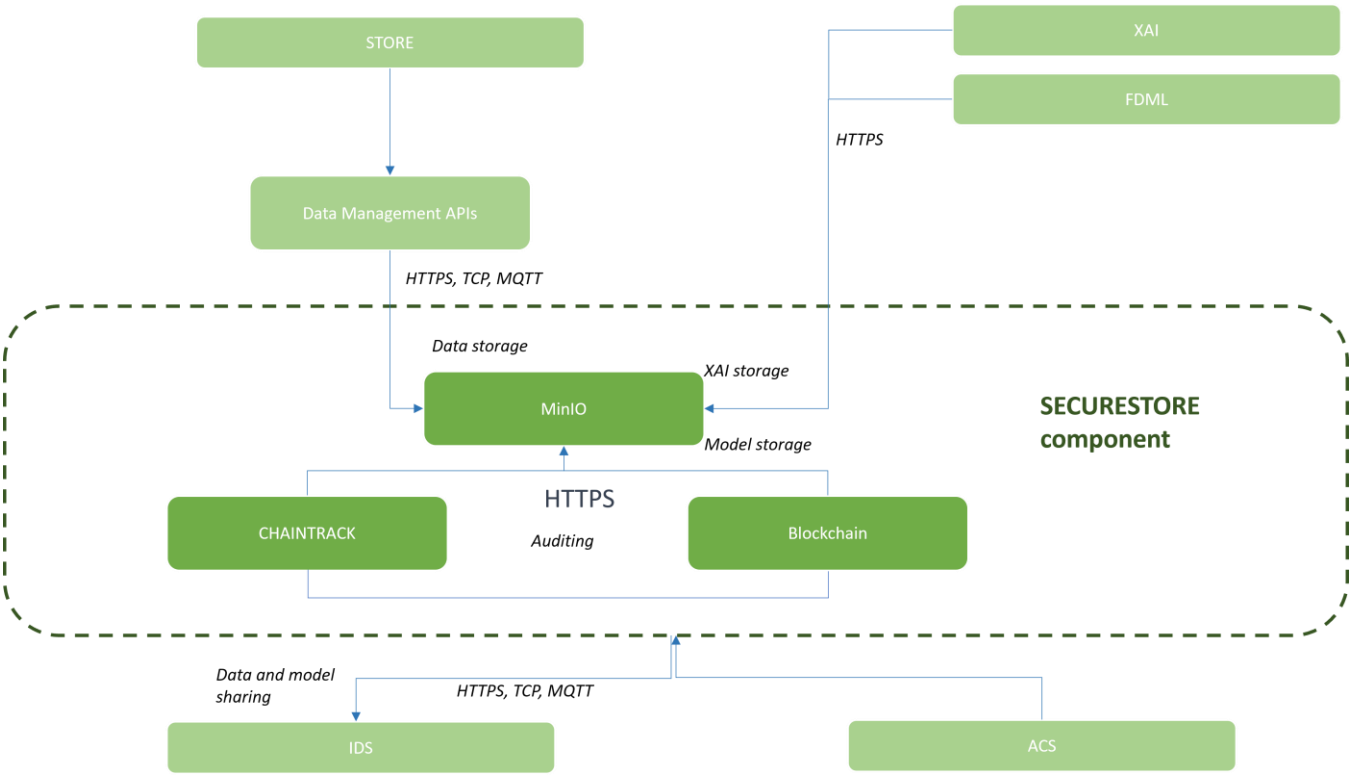

Figure 9: Logical view diagram – SECURESTORE

Figure 8 illustrates the logical view of the SECURESTORE components and its communication with i=other interacting components. Heterogenous data will come from the STORE component through the Data Management API and will be harmonized and stored for secure data sharing in a MinIO storage in SECURESTORE. The MinIO storage will also keep all incoming ML and XAI models and allow for secure sharing. Access and changes done to the data will be audited using the CHAINTRACK component and the smart contracts on the Blockchain. The sharing of data and models will be using REST APIs after access control by the ACS component and by using the IDS connectors for exchanges external to the ADS platform. The Blockchain sub-component included in Figure 9 is the same as the Blockchain Network sub-component included in Figure 10.

5.2.3.2 DLT-based supply chain tracking solution (CHAINTRACK)

This component will be used for tracking of supply chains of various agricultural products. The technologies that will be used are Ethereum and Hyperledger Besu.

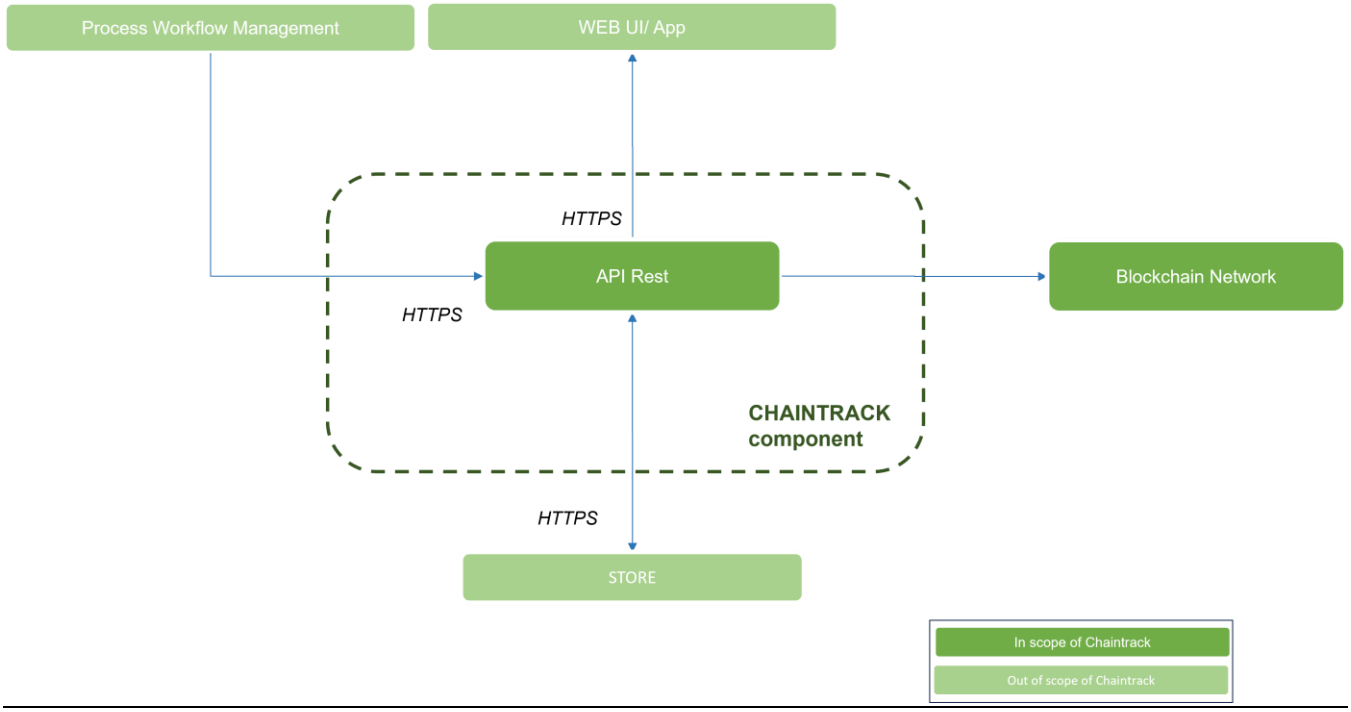

Figure 10: Logical view diagram – CHAINTRACK

The CHAINTRACK component includes a REST API that interfaces the Blockchain Network and the Store component. Information about a particular supply chain is registered in the STORE and bound with an account on the blockchain network.

Supply chain process data coming from workflow management is saved on the blockchain network in an hashed format. Actual data is saved in the STORE associated with the relevant hash. Retrieval of process information from the UI/APP is directed to STORE; information and the corresponding hashed version are returned. The transaction data can on-demand be verified against the blockchain, by comparing the hash with the one returned from the blockchain.

5.2.3.3 Access Control System (ACS)

This component will include robust security measures that will be implemented to protect sensitive data and prevent unauthorized access, including encryption mechanisms, access controls, authentication. The technologies that will be used are Keycloak (https://www.keycloak.org/) and PfSense (https://www.pfsense.org/). Figure 11 depicts in a very simple and basic way that any ADV component, before it is able to communicate with another ADV component, first needs to go through an authentication and authorization procedure with the ACS component. The ACS component aim is to offer a reliable access control system along with an authentication mechanism. The Authentication and Authorization services are related to the functionalities supported by the ACS component. The ACS component's goal is to confirm the user or component's identity when they attempt to access

ADV resources and, as a result, to make sure they can only access ADV resources for which the necessary permissions have been explicitly granted.

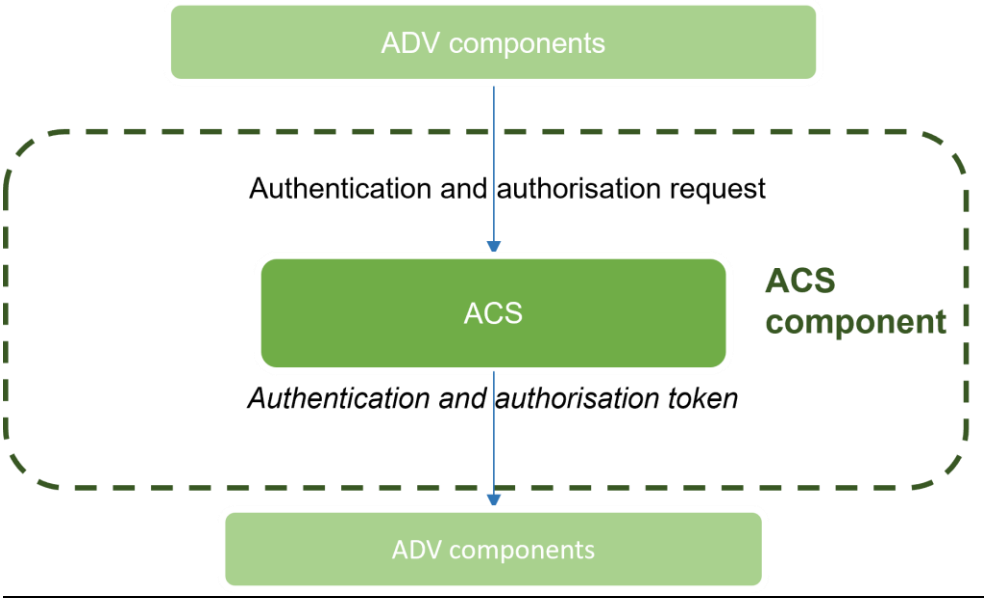

Figure 11: Logical view diagram - ACS

5.2.3.4 International Data Spaces (IDS) components

The IDS Connector and IDS Metadata Broker modules will be implemented and included in the ADV instance that will be deployed alongside the rest of the ADV components at the cloud infrastructure deployment. The IDS Metadata Broker [19]will act as a central component in the data space that manages the publication, storage, and retrieval of Self-Descriptions for IDS Resources and Connectors. Data Providers submit their Self-Descriptions to the Broker, while Data Consumers use it to find relevant data offers. The Broker maintains these Self-Descriptions and responds to search requests, also offering remote endpoints for its own Self-Description and the stored description graph. The implementation will be based on the specifications [20] published by IDSA.

The IDS Connector is a key component in the IDS network, facilitating data exchange through Data Endpoints. Whether on-premises or in the cloud, it employs container management technology for a secure environment for IDS Apps and functionalities. Deployable on multiple devices, the Connector offers data and metadata management, contract handling, IDS App oversight, and IDS protocol authentication. For the implementation of IDS Connector, DataSpaceConnector[21], which is one of the existing open-source connectors will be used. More details on this component will be included in the first WP2 deliverable, D2.1 as it is still under investigation.

[19] https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_5_0_system_layer/3_5_4_metadata_broker
[20] https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-g/Components/MetaDataBroker
[21] https://github.com/International-Data-Spaces-Association/DataspaceConnector

5.2.4 AI-Based Cloud Platform

5.2.4.1 Decentralised Knowledge Management (DKM)

This module is located on the cloud and is formed by the Server of the Federated Learning architecture. The main functionality of this module is to aggregate the local weights from the FDML Clients to generate a global model. In order to detect and avoid poisoning attacks, this module will also count with an anomaly detection algorithm to detect malicious weights and remove them from the final aggregation. The FL-AgriDataGen component will assist the DKM with this anomaly detection process by generating synthetic databases of attacks. The technology that will be used is ATOS' own Federated Learning framework. This component is depicted in Figure *7* along with its client counter-part, FDML.

5.2.4.2 Storage (STORE)

This will be the component that will provide the actual storage service as well as the data management API that will handle the data access from the rest of the ADV components. The data to be stored in the database(s) of this component will be defined by the end-users and it could be data, metadata, model parameters, etc. In case real data need to be stored, this data needs to be anonymised ore pseudonymised. The technologies that will be used are Postgres, PostGIS, MongoDB, and Redis. Figure 12 describes how ADV components will store their data at the ADV storage system. ADV's storage system is capable of including various data storages to support heterogenous data sources and streams. Relational databases as well as document-based databases will be included, and in-memory stores to assist specific needs, if needed. All those storages will be made accessible through a Data Management API which will provide transparent and homogenous access to the ADV components. The Data Management API will support various communication protocols (HTTPS, TCP, MQTT) to assist the interfacing with the ADV components. The contents of the STORE component can be further transferred to the SECURESTORE component, according to the needs of the users, as expressed via the user scenarios.

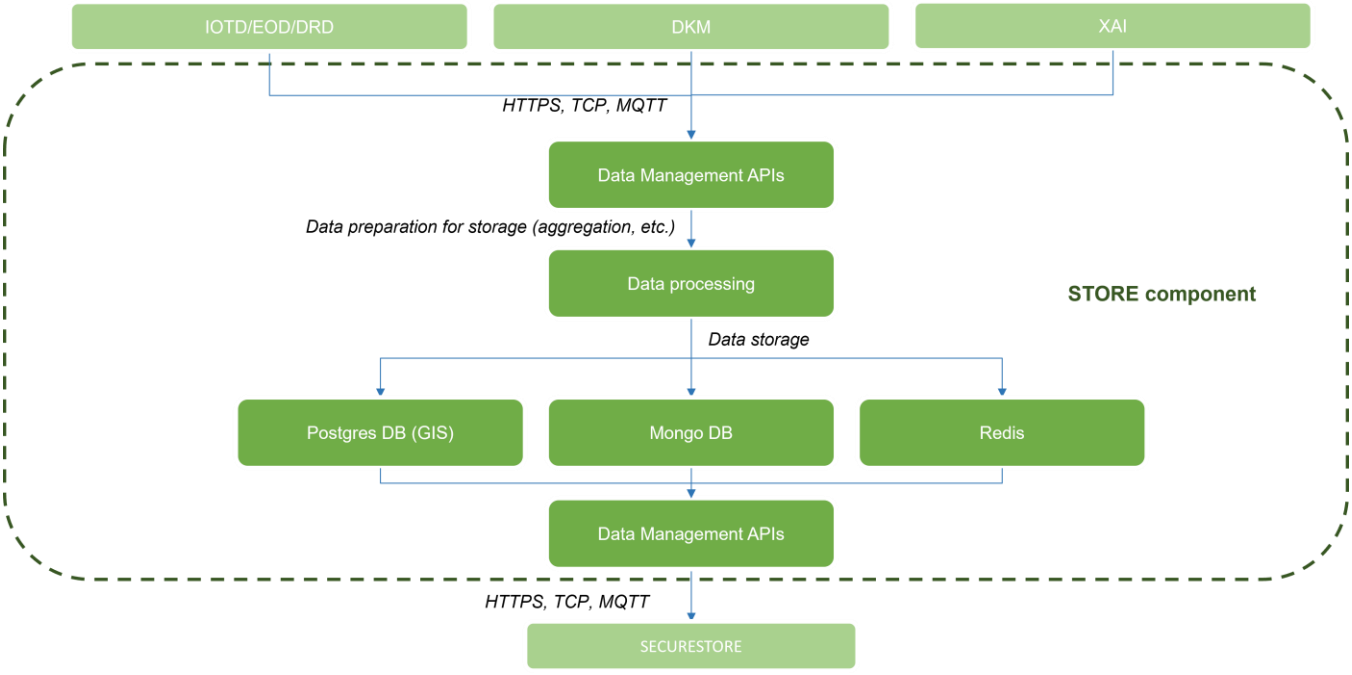

Figure 12: Logical view diagram - STORE

5.2.4.3 FL-AgriDataGen (DATAGEN)

FL-AgriDataGen is short for Agri-environmental Data Generator. This component aims to generate synthetic labeled samples based on AgriDataValue systematic analysis. To accomplish this, it will utilize state-of-the-art generative models, such as Autoencoder or Generative Adversarial Networks (GANs). FL-AgriDataGen will generate FL poisoning datasets of attacks to assist the evaluation of FDML anomaly detection algorithms. The FL poisoning attacks can target the training phase of FL networks and can be distinguished on data and model poisoning attacks. FL-AgriDataGen will focus mainly on the generation of data poisoning attacks. Attacks are considered an anomaly to the FL system and therefore they should be identified and eliminated. Before aggregating the model updates, the FL server must check if the local models coming from the clients could have suffered attacks, and act accordingly. The synthetic attacks of FL-AgriDataGen will be used to evaluate the detection and defence mechanisms of FL server. The technology that will be used is Python (Pytorch or TensorFlow). The initial logical view of this component is depicted at Figure 13.

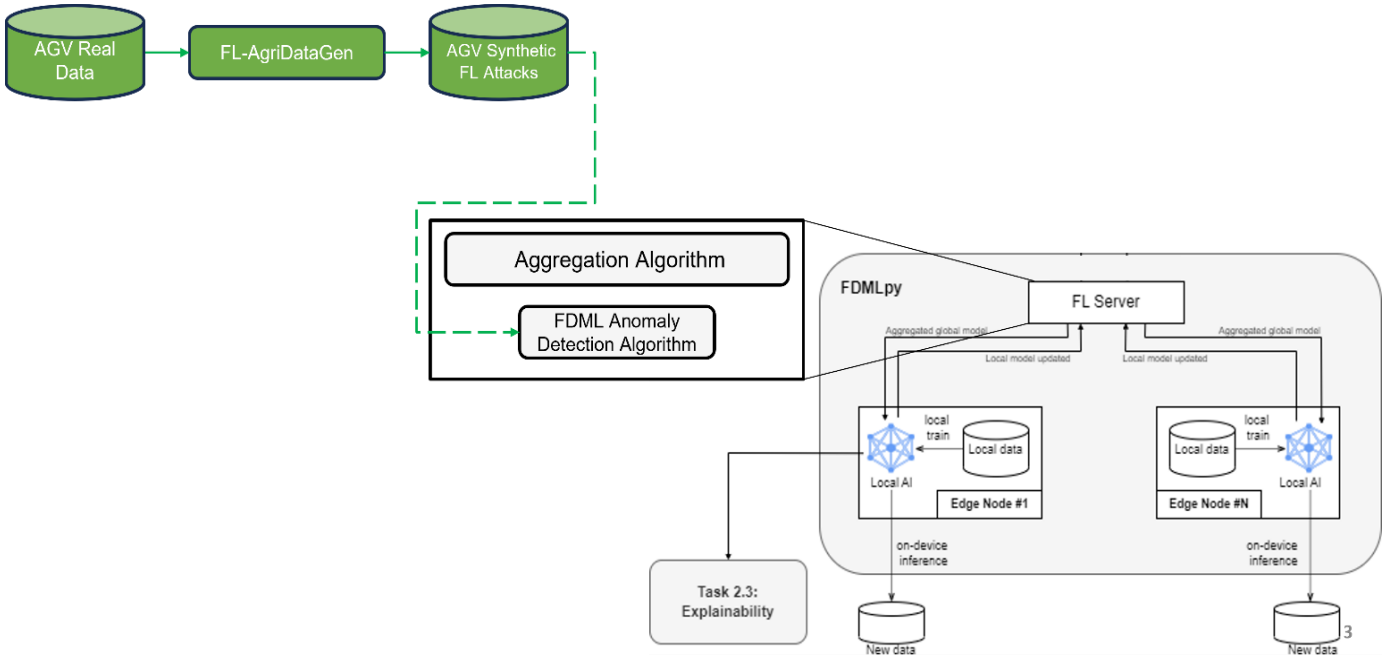

Figure 13: Logical view diagram - DATAGEN

5.3 Process view

The **process view** deals with the dynamic aspects of a system, describes the system processes and their interactions, and focuses on the run time behaviour of the system. The process view is created for individuals who design the entire system before integrating the system into a system of systems or into individual subsystems. This view displays the system's tasks and processes, interfaces with the outside world and/or between system components, as well as messages sent and received.

In ADV, for the Reference Architecture, we identified a synthetic, yet indicative, platform-wide process that involves all the components (except for the IDS component, the functionality of which will be described in detail

in the upcoming deliverables of WP2, D2.1/2/3) of the integrated platform, based on the user scenarios as they were described in deliverable D1.1:

1. ADV synthetic use case

For that platform-wide process, we have created their corresponding Sequence diagram. This diagram is presented in the following subsection.

5.3.1 Sequence diagram

The diagram in Figure 14 depicts the sequence in the interactions between the integrated platform's components from the moment the data collection toolboxes collect data from various resources to the final storing of the data (aggregated, metadata, models, etc.) to the storage components of the integrated platform.

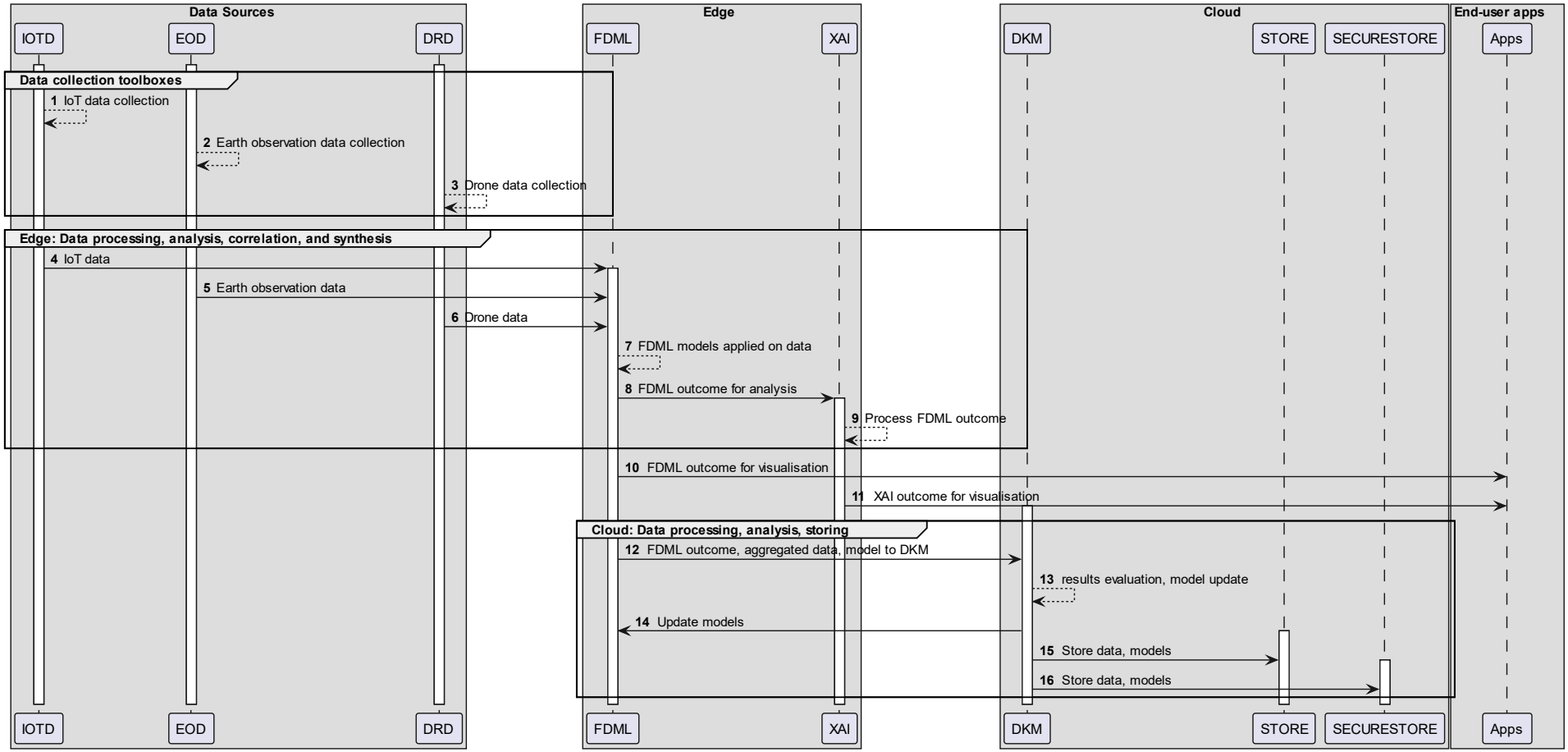

Figure 14: Sequence diagram

In more specific, the sequence diagram above describes how the components that belong to the Data sources (IOTD, EOD, and DRD) collect data, then, this data is transferred to the FDML component so that the AI models can be applied on the datasets locally, while the results of this process along with any source data is also received by the end user application as well as the XAI component so that meaningful and useful insights can be provided to the end-users. Subsequently, the DKM component receives data and results that will be evaluated towards the models' update. The last step of this scenario is storing data, metadata, models' parameters so that further actions can be taken on them.

5.4 Data view

The purpose of this section is to describe how the ADV RA data view is represented. This viewpoint focuses on the architecture of data storage, data retrieval, processing, and security administration. The view emphasizes data flows and the elements required to support and control key operations, like archiving and processing. There were minor improvements and enrichments made in regards to the identification of components that participate in the data management in the ADV platform, but there were no significant departures from the criteria used to define the methodology that served as the inspiration for this view. Any information system's ultimate goal is to manipulate data in some way, of course. This information may be transiently altered during program execution, persistently stored in a database management system, or simply stored in regular files.

The main ADV data archives are identified as follows:

- STORE
- SECURESTORE

In this data view, it is necessary to link the ADV modules that address the platform's data management requirements to the remaining ADV components.These ADV modules are connected to the remaining ADV components, including the DKM, XAI, and FDML components that are part of the data processing group of components. By standardizing the information flow originating from the ADV components using a formal definition of the ADV data sources, the STORE and SECURESTORE components address the issue of the physical representation of the ADV data information model. After being normalized, this data can be quickly interpreted by the ADV's core software components using a data management API layer.

The technologies that will support the STORE and SECURESTORE databases will be based both on the SQL and the NoSQL paradigms, optimized for the representation of a flexible data model (ADV data model). In addition, the SECURESTORE will be blockchain-based in the sense that it will use the blockchain technology to store information regarding (meta)data that will be stored and need to be traced back in the future. The data management APIs and the technologies to be used will allow the dynamic modelling of an entity: i.e., if in the future the formal representation of the information model has to change, this will not represent a problem for the ADV platform. Furthermore, an important aspect that needs to be highlighted is that the full interoperability and flexibility of the ADV data model and the IDS component's technology are suitable for future integrations with other technologies in the agricultural space (in other spaces, too).

It is essential to think about the API frameworks able to support and work with these technologies in addition to the data store itself (only data containers or the technologies that enable data persistence). In reality, it permits operations like reading and writing data while preserving the business logic of individual services and the overall functionality of the technology they use when viewed collectively. It should be noted that in order to contextualize

these frameworks with the ADV components and place them in the macro functional block of competence, it is important to note that the STORE and SECURESTORE represent the data management system, which is the true center of the data management software block, and the Access Control management APIs (through the ACS component) frame this block, representing a vertical solution that enables secure data exchange and communication between systems.

The diagram in Figure 15 below brings together all above concepts, i.e., entities, information, and their interactions within ADV processes, in order to structure the data flows between them.

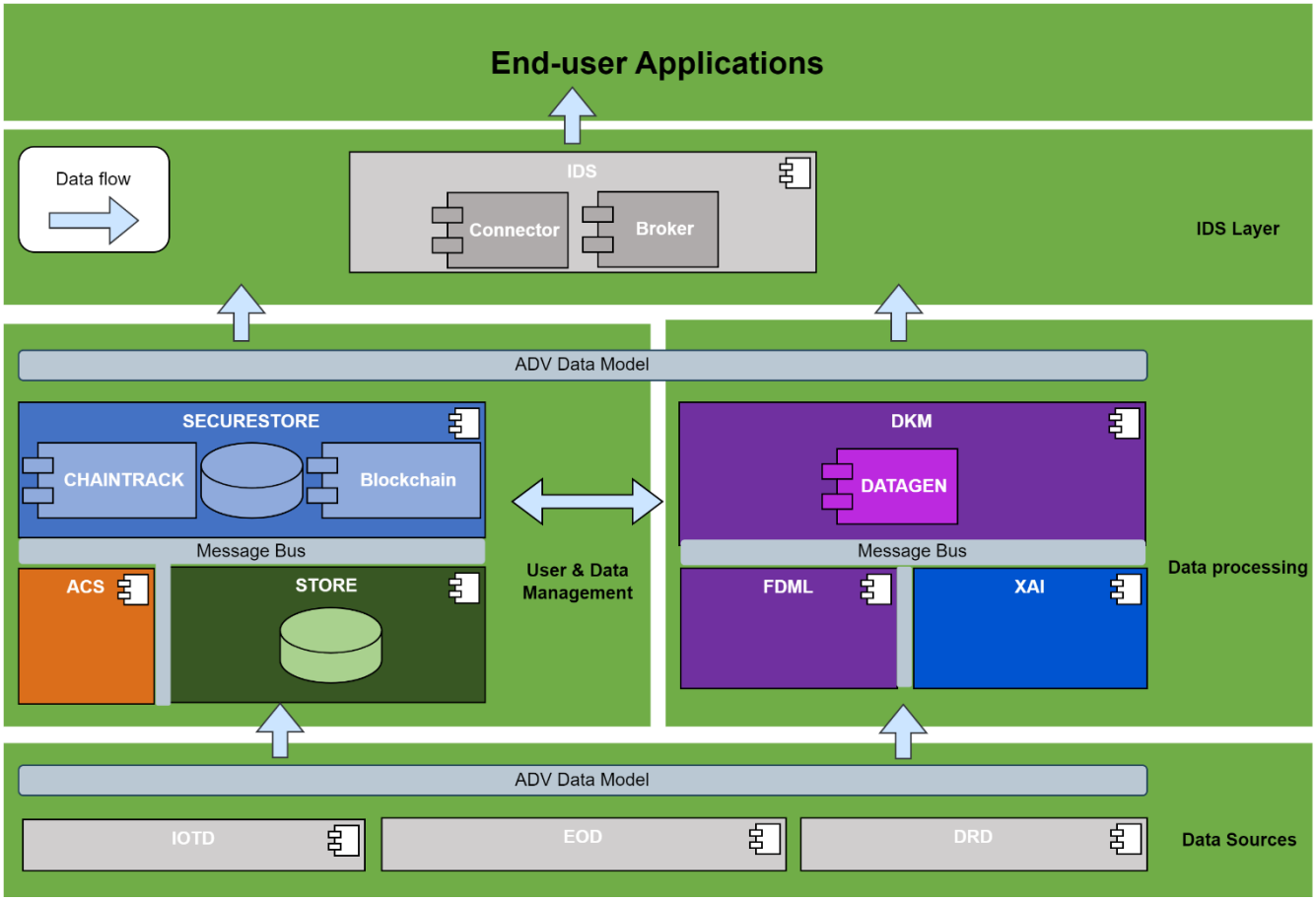

Figure 15: Data view

Any data sources (e.g., sensors, devices, services), that can feed the ADV platform with its own data, communicate with the ADV toolbox components' APIs. Specific wrappers/translators on these ADV toolbox components side deal with translating the sensor data that are not compliant and not aligned with the ADV data model. This allows support for data interoperability by combining the use of a data model with the respective data translation/management mechanisms, even in case other standardised solutions are used by the ADV stakeholders.

ADV offers the necessary structures to support data transformation and data exchange in accordance with the ADV data format, enabling complete interoperability between heterogeneous data modelling/semantic approaches. The ADV platform components can then make use of the translated and semantically aligned data. The ACS component and the data management APIs are integrated to allow the application of data security

policies. The ACS is crucial for potential data-sharing connections with other platforms or infrastructures (toolboxes) because it serves as the initial point of contact with the ADV infrastructure, opening the door for future ADV platform integration and potentially secure collaborations with other research projects. The other ADV components can use the injected data through APIs once it has been safely stored inside the ADV databases. By utilizing the APIs, this information can be made available throughout the ADV architectural layers. The metadata for each resource is made available through the STORE and SECURESTORE APIs, and the data are packaged as ADV resources. Then, all business processes are fed by these APIs in order to fulfil all use case scenarios specified in the project.

To enable full interoperability with end users that would like to use the data available in an ADV platform instance, the IDS is employed, offering secure interexchange of data based on mutual agreements and pre-agreed policies among the participating entities.

5.5 Deployment view

The Deployment view depicts the system from a system engineer's point of view. It is concerned with the physical connections between these components as well as the topology of software components on the physical layer. After the system has been tested and is prepared to enter live operation, the Deployment view concentrates on components that are crucial. The mapping of the software components to the runtime environment in which they will be executed is described in this view, along with the physical environment in which the system is intended to operate. Any information system that has a required deployment environment that is not immediately apparent to all interested stakeholders falls under the deployment viewpoint.

In ADV we identified the nodes that will be present upon the run-time of and the execution of the entire system along with the components that participate, both internal and external ones. In the ADV deployment diagram in Figure 16, the locations where the several software artefacts reside are depicted.

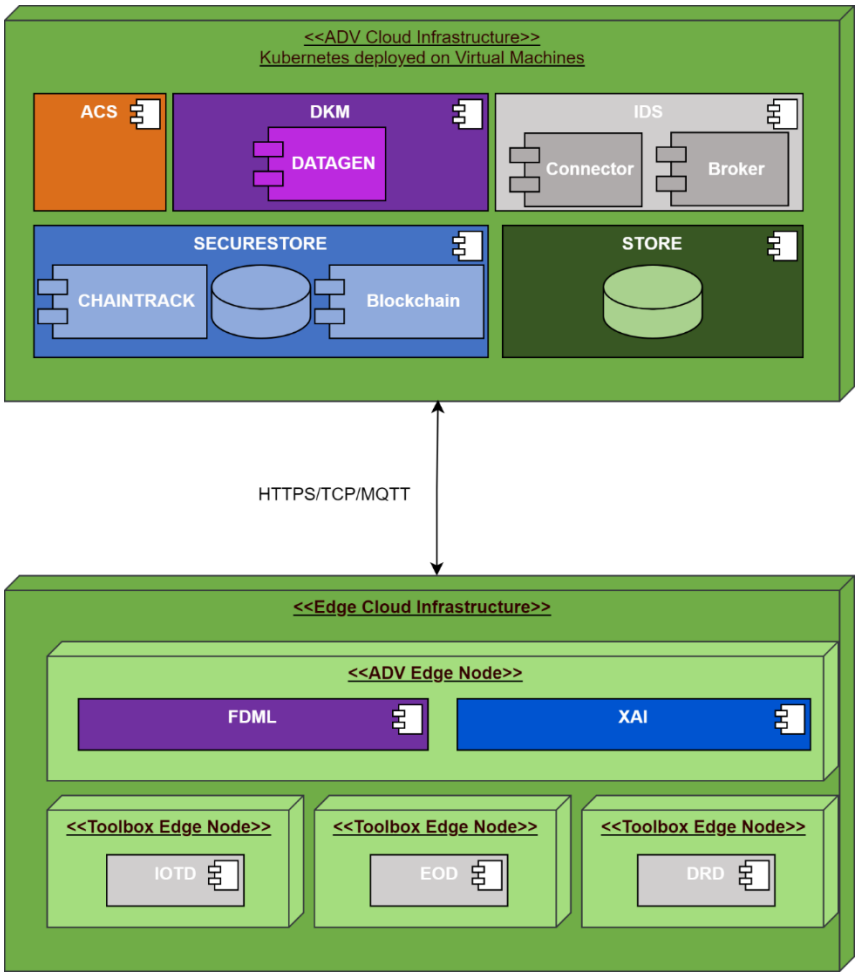

Figure 16: Deployment view

As depicted in Figure 16, the ADV platform deployment is split into the cloud deployment and the edge deployment. The cloud-based infrastructure will host the cloud-based components that belong to the AI-based cloud platform block and the Data security, Privacy, Traceability & Sharing blocks (as depicted in Figure 4). On the other hand, the deployment at the edge cloud infrastructure will include the components that will be located closer to the pilot sites and belong to the Edge Cloud Analytics Suite block and the Decentralised data capture management & in-situ pre-processing tools block. Data exchange will be performed by the appropriate (authorized) components using the pre-defined protocols (HTTPS, TCP, MQTT), based on the authorization policies employed by the ACS component, and by using the ADV adaptors to convert to/from the ADV data model. The edge cloud infrastructure can of course include several edge nodes, each of them hosting one or more components.

It is worth mentioning that the ADV platform is envisioned to be a multiple instance platform, i.e., multiple instances of the ADV platform (both cloud and edge parts) can potentially be deployed, utilizing the IDS component as the interoperability enabler for their interconnection and communication. Therefore, ADV platform instances serving data sources at a local, regional, national, and continental level can potentially be deployed.

6 Interfaces between main architecture components

In the ADV Reference Architecture, modularity and adaptability are supported by exploiting the Microservices Architecture patterns and scalability is implemented to support both a cloud-based and on-premises approach.

This section presents the main interfaces of the ADV platform. The concerns of various ADV stakeholders, particularly business and technical partners who want to build a system that is technologically capable of demonstrating the execution of the ADV pilots, can be addressed by having a complete view of the key system interfaces. Based on these presumptions—to provide an overall view that is more realistic and the application context in which this view is located—the interfaces between the primary architectural layers were created.

The ADV main interface view (depicted in Figure 17) is designed using an iterative development process, dealing with the design and implementation of a high-level structure based on the discussions during the first period of the project. It is the end result of putting together a specific number of components in a few well-chosen configurations to fulfil the primary functional requirements of the system. First off, almost every component has interfaces or APIs that can connect to other ADV systems. As a result, when software components share the ADV data model and become interoperable by exposing common APIs, they can communicate with one another via remote interface. Defined interfaces are also used to control management procedures and decision-making. Finally, even though they are not centrally deployed on the cloud but rather locally deployed in the pilot infrastructures, the component developers are able to expand their field of action within various infrastructures thanks to the adoption of these interfaces.

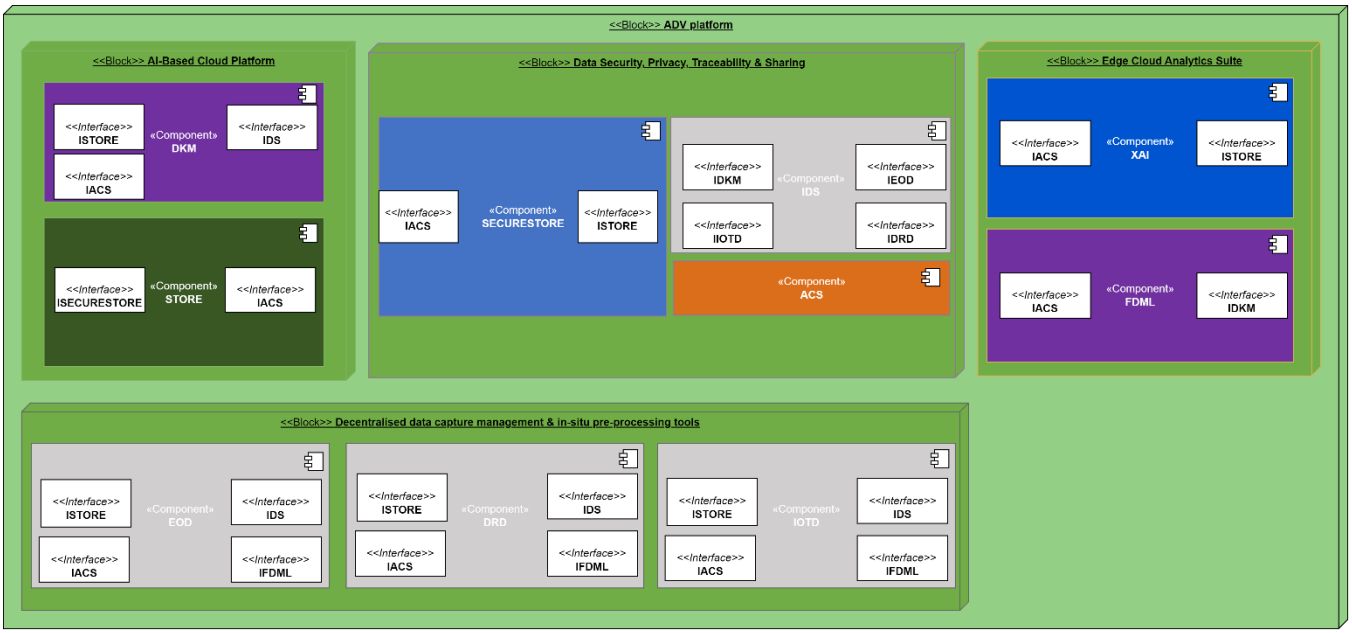

Figure 17: Interfaces view between components

The Decentralised data capture & in-situ pre-processing tools block is the lowest level of the ADV architecture, which implements the interfaces that interact with the ADV integrated platform components (either the ones

located at the edge, or the ones located at the cloud) to send and/or store data collected from the data resources. ADV integrated platform's components are represented by three blocks: Edge Cloud Analytics Suite, Data Security, Privacy, Traceability & Sharing, and AI-Based Cloud Platform. The components in these blocks implement the interfaces of the components they need to communicate with as described in the Functional view section (subsection 5.2).

The implementation and instantiation of the main interfaces will enable a set of operations per block of components, which will all be described in upcoming WP2 deliverables (i.e., D2.1/2/3). The IDS component also supports the interoperability of the ADV integrated platform, and the data generated and stored, with user interfaces or web GUIs, enabling their utilisation by web-based applications.

The ADV components will essentially provide a set of necessary APIs to support basic operations like data acquisition, data retrieval, data management and storing. The majority of the functional and non-functional requirements outlined for the ADV project are covered by the business logic functions that are present in the components and exposed through APIs REST framework. In order to ensure continuous integration and continuous delivery, it is obvious that these business functions will be further improved, enhanced, and refined over time, all the way up to project completion. By using this approach, continuous releases of new versions will be anticipated (reasonably until the end of the last round of Pilots), in addition to covering the project milestones, which in each case support at most three releases of the overall platform. These releases will address changing business needs, address unmet requirements, or mitigate incorrect behaviour of the individual software blocks. The components use REST interfaces of message queues, such as Kafka and/or RabbitMQ, to interact with each other. Finally, this section clarifies the components' interaction using their interfaces and is meant to assist the developers in identifying the components they should access through the provided interfaces.

7 Security, Privacy, and GDPR considerations

As part of the ADV project, we will collaborate with agrifood industry experts from all over Europe to assess the platform's outcomes. As a result, there are a number of factors to take into account when it comes to the technological side of the GDPR, including data integrity and confidentiality, security, access control, traceability, and data provenance. Dataset reviews, GDPR guides, advice, and other GDPR management considerations are outside the scope of this article. The actions listed below must be taken in order to comply:

- Control of data access: The access control system must make sure that only authorized individuals or organizations use the ADV platform and must limit access to personal data in accordance. Mechanisms for authentication and permission will do this. In compliance with Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, the data analysis will be conducted anonymously, and it won't be possible to identify specific users. No information will be moved outside of the pilots' facilities.
- Equipment access control: Regardless of the specific implementation (in-house services or cloud services), the access control system must prevent unauthorized individuals or entities from accessing processing equipment.
- Storage control: The ADV platform shall prohibit unauthorized personal data input as well as unauthorized access to, modification of, or deletion of stored personal data. The data produced by ADV should be saved in an unchangeable and traceable manner to ensure effective control over it. The ADV platform must put in place enforceable user limitations and keep track of every time data is handled. The nature, scope, and context of any processing of personal data shall be taken into consideration when determining the relevant technical and organizational security measures to be used against unauthorized access and modification. Each pilot user will have the option to opt out of the measurements at any time, and recruitment of the pilot users will be done on a voluntary basis (and a signed informed consent in the participant's language will be required). Additionally, the pilot user's data can be deleted from the database and excluded from any analyses upon request.
- Communication control: No information will be sent or made accessible to anyone outside of the pilots' premises. Users of the ADV platform may share data with one another within the parameters of their individual privacy laws. The ADV platform won't be able to successfully stop improper data sharing. As part of the access control system, the data export from the ADV platform could be restricted to particular users or organisations in this regard.
- Input control: make sure it's easy to confirm and determine which personally identifiable information has been entered into automated processing systems, as well as when and by whom. By keeping track of how the tools are being used, the ADV platform will guarantee input control.
- Transport Control: Data managers are required to prevent unauthorized access to, or the copying, modification, or erasure of, personal data during data transfers. The ADV platform would put security safeguards in place like encryption to prevent data from being used without permission.
- Recovery: Make that installed systems can be restored in the event of an interruption. The ADV platform will be set up such that stored data is properly backed up.
- Reliability: The software must ensure that the functions of the system perform correctly and that the appearance of faults in the functions is reported. The ADV platform will be subject to logging and traceability, thereby making the analysis of possible technical issues possible.

- Integrity: ensure that stored personal data cannot be corrupted by means of a malfunctioning of the system. The integrity of data stored by the ADV platform can be secured by the implementation of backup routines and access control.

The design and operational use of the ADV platform should adequately take into account the aforementioned concepts. Although some of the standards go beyond what can be guaranteed during the ADV platform's development phase, the project seeks to create tools that help data controllers follow these values. However, there is still a concern with data misuse that cannot be entirely avoided.

7.1 Other technical measures

The majority of the elements indicated in the previous section are also contained in the Security requirements that were reported in deliverable D1.1 and determined by the users. To guarantee that the ADV integrated platform takes these factors into account, this document mapped them to Technical Requirements (please, check Section 2.2) to ensure the ADV integrated platform addresses these aspects. To summarize, the following technical considerations have been taken into account to ensure the arguments made in the previous subsection:

7.1.1 Access control

A subject's identity (whether it be a person or a smart object) is authenticated to confirm that the subject is who or what it claims to be. It permits associating a subject with an identity. The subject can be authenticated using something they know (like a password), something they have (such smart cards or security tokens), or something they are (like a fingerprint or retinal pattern).

Based on the supplied credentials, an authentication component enables the authentication of individuals and smart objects. The credential may take the form of a shared key, a digital certificate, or a login and password. Following the authentication procedure, an assertion is produced that may be used to state that a certain subject was successfully authenticated by the issuing authority. This will guarantee that access to the ADV's data and equipment, as well as its transport and input control, will be restricted.

7.1.2 Traceability

Since many years ago, traceability has been a well-known aspect in several businesses. Traceability in logistics refers to the ability to track items from suppliers to retailers along the distribution chain. In the case of ADV, it is essential to rely on it throughout pilot user operations to tell all stakeholders of the platform's outcome completely. There are many ways to trace things and the information associated with them, including DLTs, CAS, bar coding, etc.

When it comes to traceability services, access control in ADV is crucial in order to keep track of who enters what and when. All parties involved will need to communicate data and analysis, and doing so necessitates keeping a log for addressing potential audit logs or security breaches.

Regarding DLTs (Digital Ledger Technologies), many approaches have been studied for traceability in transactions and in general, aiming to explore the advantages of having a cryptographically secure and immutable record of transactions. One of the solutions can be integrating the data into a private blockchain, which offers stakeholders a more data-private solution.

7.1.3 Data provenance

IoT's widespread nature poses major security and privacy problems because extremely sensitive data is routinely transmitted, even when users are unaware of it. With European stakeholders interacting in settings that are increasingly globalized, personal information and individual identities are becoming more and more susceptible in a digital environment. The ongoing lack of confidence is caused by a lack of solutions, including regularly used technologies and procedures for reliable enrolment, identification, and authentication procedures. In particular, the usage of online credentials with weak authentication assurance is to blame.

In these situations, managing the data provenance becomes crucial. Data provenance is a procedure that establishes a data product's history, beginning with its original sources. In the IoT infrastructure, access breaches can be found using data with assured provenance. Developing verified data provenance is still a crucial issue, though. Additionally, provenance data may include private information about the owners of the original data and the data itself. Therefore, it is important to protect not only the data but also the provenance data's integrity and reliability.

In this regard, the data provenance metadata shall enable revealing the true identity of the owner connected with the exchanged IoT data when the inspection grounds are satisfied (for example, identity theft or related crimes). Additionally, the provenance information for the data should be permanently attached to it so that it can be tracked and audited wherever it is kept or shared, in transit or at rest.

7.1.4 Privacy and Security by-design technologies

AgriDataValue platform of platforms will ensure that all necessary confidentiality and privacy related processes will be incorporated into the data flow among its components and services. In this context, ADV will consider the security/privacy by-design technologies described in the following subsections: identity management and privacy-preserving group communication.

7.1.4.1 Identity Management (IdM)

IdM manages controlling some entity information, including identities, credentials, and pseudonyms, and it includes APIs that allow system administrators to change entity information. The IdM system allows distributed and scalable deployment in an IoT context to achieve excellent performance with a large number of devices and identification data. For the IdM system to be deployed in constrained IoT gateways, it must also support limited computational resources.

IdM will offer cutting-edge technologies and procedures in ADV to manage secure access to identification data and maintain the components' and data's privacy. The capability provided by both the ACS and IDS components will be used to accomplish this. Deliverables D2.1, D2.2, and D2.3 will include specific details about their functionality.

7.1.4.2 Attribute Based Encryption (ABE)

AgriDataValue intends to use well-known tools and mechanisms to offer confidentiality, GDPR-compliant privacy-by-design. The tool that is under consideration is Keycloak[22] (as part of the ACS component), which supports fine-grained authorization policies and is able to combine different access control mechanisms such as:

- Attribute-based access control (ABAC)
- Role-based access control (RBAC)
- User-based access control (UBAC)
- Context-based access control (CBAC)
- Rule-based access control
    - Using JavaScript
- Time-based access control
- Support for custom access control mechanisms (ACMs) through a Service Provider Interface (SPI)

It is evident from the subsections above that Keycloak can support both CP-ABE and KP-ABE. CP-ABE seems to fit better and more naturally with RBAC, whereas KP-ABE not so much. Considering attributes, instead of the users, as tags of the objects/documents encrypted, seems to describe that fact in a better way.

[22] https://www.keycloak.org/

8 Conclusions and Next Steps

This report defines and specifies the AgriDataValue Technical Specifications & Reference Architecture.

The result of this deliverable is a modular, coherent, and optimized architecture that is specified at a sufficient level of detail to continue with the implementation of the AgriDataValue platform of platforms and its components. It initially presents the technical requirements and their connection to the user requirements, along with the initial technical specifications' guidelines, as a basis for the implementation of the ADV platform's components. The discussion of the approach to be used for defining the ADV data model, which will determine the degree of interoperability the ADV is capable of achieving, is then brought up. The principles of the reference architecture design methodology are then presented, and the core of this document follows. That is a thorough explanation of the ADV Reference Architecture that was designed, presented in five different architecture views (high-level view, functional view, process view, data view, and deployment view). Next, a discussion of the high-level interfaces between the primary ADV architecture blocks follows. The document concludes with a summary of the information covered and the current plans for the future before presenting the key GDPR considerations that have been put in place.

The content of this deliverable is the result of collaborative work of partners from T1.3, T1.4, T1.5, who worked upon sections 3, 2, and 4-5-6-7 respectively, as well as technical partners from WP2, WP3, and WP4 who provided their input regarding their components when requested.

It is necessary to mention that the Reference Architecture presented in this deliverable will be complemented by two more deliverables that will include more technical and implementation-related details:

- D2.1 AgriDataSpace Underlying Technology
- D2.2 AgriDataSpace Platform of Platforms V1

These deliverables will provide additional information on the specific components developed, on their interfaces, on the technologies and any existing solutions/tools to be leveraged, etc., which are not in the scope of this deliverable.

A new version of this document will be made in deliverable D1.4 - AgriDataValue Reference Architecture Update, presenting the revised ADV Reference Architecture in M32.

9 References

[1] European Commission, “EC H2020 Programme, Guidelines on FAIR Data Management in Horizon 2020, Version 3.0.,” [Online]. Available: https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf.

[2] International Standards Organisation, “ISO 26324:2012(en) Information and documentation — Digital object identifier system,” [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:26324:ed-1:v1:en.

[3] Wordpress, [Online]. Available: https://wordpress.org.

[4] Twitter, [Online]. Available: https://twitter.com.

[5] Linkedin, [Online]. Available: https://www.linkedin.com.