D3.4 SCALEAGDATA GENERIC ARCHITECTURE AND DATA GOVERNANCE, SHARING META-ARCHITECTURE AND INTEGRATION OF THE RI LABS, V2.1

13 May 2026

Authors		
Name	**Organization**	**Draft release date**
Giorgos Tsilimanis	ICCS	22/10/2025
Theodoros Giannakas	ICCS	22/10/2025

Approval Signature on behalf of the Technical Steering Committee		
Name	**Organization**	**Date of approval**
Koen Van Rossum	VITO	31/10/2025

Revision Records			
Version	**Date**	**Changes**	**Authors**
0.1	16/01/2024	Original document	ICCS
0.2	24/01/2024	Revised based on reviewers' comments	ICCS
1.0	31/01/2024	Final D3.1	ICCS
1.1	22/09/2025	Update of D3.1 to D3.4 first draft	ICCS
1.2	27/09/2025	Revised based on reviewers' comments	ICCS
2.0	31/09/2025	Final D3.4	ICCS
2.1	13/05/2026	Revised version that takes the comments of the EC and external reviewers into account	All WP3 partners

Acronyms and Abbreviations

Acronyms and Abbreviations	
AAD	Azure Active Directory
ABAC	Attribute-Based Access Control
AGROVOC	Agriculture Vocabulary
AI	Artificial Intelligence
AIM	Agricultural Information Model
AMQP	Advanced Message Queuing Protocol
ANN	Artificial Neural Network
API	Application Programming Interface
APSIM	Agricultural Production Systems sIMulator
AsyncAPI	Asynchronous API Specification for event-driven architectures
ATB	Institut für angewandte Systemtechnik Bremen GmbH
AUTH	Aristotle University of Thessaloniki
AVR	AVR BVBA
BioPAR	Biogeophysical PARameters
BLE	Bluetooth Low Energy
CA	Consortium Agreement
CRUD	Create, Read, Update, Delete
CSV	Comma-Separated Values
CSW	Catalogue Services for the Web
DA	Data Act
DCAT	Data Catalog Vocabulary
DES	Deimos Spain
DEMETER	H2020 Project on Smart Farming Data Interoperability
DGA	Data Governance Act
DHI	DHI A/S
DID	Decentralized Identity
DME	DEIMOS ENGENHARIA SA
DMK	DMK Deutsches Milchkontor GmbH
DSS	Decision Support System
DSSC	Data Space Support Center
EC	European Commission
EDC	Eclipse Dataspace Components
EDIB	European Data Innovation Board
EGM	Easy Global Market SAS
EO	Earth Observation

EOD	Earth Observation Data
ETL	Extract, Transform, Load
ETSI	European Telecommunications Standards Institute
EU	European Union
EURAC	Accademia Europea di Bolzano (Eurac Research)
EVA	Evapotranspiration
EV ILVO	Eigen Vermogen van het Instituut voor Landbouw en Visserij Onderzoek
ExBo	Executive Board
FAIR	Findability, Accessibility, Interoperability, and Reuse
FC	Federated Catalogue
FIWARE	Future Internet WARE
FMIS	Farm Management Information System
fPAR	fraction of absorbed Photosynthetically Active Radiation
GA	General Assembly
GDPR	General Data Protection Regulation
GeoJSON	Geographic JSON
GeoTIFF	Geographic Tag Image File Format
GIS	Geographic Information System
GNDVI	Green Normalized Difference Vegetation Index
GPP	Gross Primary Productivity
gPRC	GPRC Remote Procedure Call
GraphQL	Graph Query Language
HORTA	HORTA SRL
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IA	Innovation Area
IAM	Identity and Access Management
ICCS	Institute of Communication and Computer Systems
IDSA	International Data Spaces Association
IFAPA	Instituto Andaluz de Investigación y Formación Agraria, Pesquera y Alimentaria
IoT	Internet of Things
IPR	Intellectual Property Rights
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
JWT	JSON Web Token
Keycloack	Open-source Identity Provider

KUVA	Kuva Space Oy
LAI	Leaf Area Index
LDES	Linked Data Event Streams
LoRaWAN	Long Range Wide Area Network
LUE	Light Use Efficiency
LUKE	Natural Resources Institute Finland
MIGAL	MIGAL Galilee Research Institute
ML	Machine Learning
MQQT	Message Queuing Telemetry Transport
NDMI	Normalized Difference Moisture Index
NDRE	Normalized Difference Red Edge
NDVI	Normalized Difference Vegetation Index
NGSI-LD	Next Generation Service Interface with Linked Data
NP	Neuropublic SA
NPP	Net Primary Productivity
OAS	OpenAPI Specification
OAuth2.0	Open Authorization Framework
OCB	Orion Context Broker
OGC	Open Geospatial Consortium
OHB DS	OHB Digital Services GmbH, Bremen, Germany
OIDC	Open ID Connect
OSLO	Open Standards for Linked Organizations
OWL	Web Ontology Language
PDA	Personal Digital Assistant
PEF	Product Environmental Footprint
PET	Privacy Enhancing Technologies
pH	potential of Hydrogen
PSNC	Instytut Chemii Bioorganicznej Polskiej Akademii Nauk
RACI	Responsibility Assignment Matrix
RBAC	Role-Based Access Control
R&D	Research and Development
RDF	Resource Description Framework
REST	REpresentational State Transfer
RF	Random Forest Regressor
RFC	Request For Comments
RGB	Red, Green and Blue
RH	Relative Humidity

RIE	Research and Innovation Environment
RIL	Research and Innovation Lab
S3	Simple Storage Service
SAR	Synthetic Aperture Radar
SAREF4AGRI	The SAREF ontology (the Smart Appliance REFerence ontology)
SDK	Software Development Kit
SME	Small and Medium Enterprise
SOC	Soil Organic Carbon
SPI	Service Provider Interface
SSL	Secure Sockets Layer
SVM	Support Vector Machine
T	Temperature
TBD	To Be Decided/Determined
TLS	Transport Layer Security
UGent	Universiteit Gent
UMA	User Managed Access
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Universal Time Coordinated
UTF-8	Unicode Transformation Format-8-bit
VI	Vegetation spectral Indices
VITO	Vlaamse Instelling voor Technologische Onderzoek
VRI IES	Foundation "Institute for Environmental Solutions"
VSDS	Flemish Smart Data Space
VTT	Technical Research Centre of Finland Ltd.
Wi-Fi	Wireless Fidelity
WKT	Well-Known Text
WMS	Web Map Services
WODR	Wielkopolski Osrodek Doradztwa Rolniczego w Poznaniu
WP	Work Package
WPS	OGC Web Processing Service
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Project Overview

ScaleAgData is a response to the call HORIZON-CL6-2022-GOVERNANCE-01-11. Upscaling (real-time) sensor data for EU-wide monitoring of production and agri-environmental conditions. The ScaleAgData project runs from January 2023 till December 2026 and consists of a consortium of twenty-six partners from fourteen countries. The vision of ScaleAgData is twofold. On one hand, it wants to obtain insights into how the complex data streams should be governed and organised (governance call). On the other hand, it aims to develop the data technology needed to scale data collected at the farm level to regional datasets, agri-environmental monitoring and the management of agricultural production.

To do so, ScaleAgData has five objectives:

- Developing innovative approaches for collecting in-situ data and applying data technologies.
- Enabling and promoting data sharing along the entire data value chain.
- Demonstrating how the sensor data can be scaled to agri-environmental data products at the national, regional or European level.
- Demonstrating the benefit of the improved monitoring capacities in a precision farming context.
- Demonstrating the benefit of upscaled regional datasets for the agricultural sector in general.

During its lifecycle, the project will explore seven innovation areas:

- innovative sensor technology
- edge processing
- data sharing architecture and data governance
- satellite data augmentation
- data assimilation to service development
- privacy-preserving technology
- data integration methodologies

Six Research and Innovation Labs (RILs) have been identified within the project, across various biogeographical regions of Europe, where different data upscaling and integration models or approaches will be evaluated and demonstrated. The six RILs are water productivity, crop management, yield monitoring, soil health, grasslands and sustain dairy. Recommendations will be formulated on how such integrated datasets can be capitalised to help national and regional policymaking to strengthen both the competitiveness and sustainability of European agriculture.

1.2 Scope

The scope of this deliverable is to document the current state of the updated architectures of each RIL as they have been initiated based on the common architectural blueprint presented in the first version of this deliverable – D3.1 "ScaleAgData Generic Architecture and Data Governance, Sharing Meta-architecture and Integration of the RI Labs v1".

1.3 Intended Readership

This deliverable is primarily addressed to the Consortium partners and the Commission services, acting as a comprehensive reference document. Its purpose is to (i) set out the specification of the overall

system architecture and its key components and (ii) present the architectures established within each lab.

1.4 Document Structure

The document is structured as follows:

- Chapter 1 – Introduction: Provides an overview of the ScaleAgData project, outlines the purpose and scope of the deliverable, identifies intended readership and explains the document structure.
- Chapter 2 – Describes the methodology that was followed for the architecture definition and presents the overall proposed architecture view. It then elaborates on the functional view, detailing the tools and modules implemented in the ScaleAgData. Chapter 2 also introduces the Research and Innovation Environment (RIE), discusses data interoperability and governance mechanisms, and presents the specifications of external dataspaces and APIs.
- Chapter 3 – References: Lists all sources and publications referenced throughout the document.

1.5 Evolution of the Document

The document aims to provide an updated version of the ScaleAgData generic architecture and data governance, sharing meta-architecture and its realisation in the context of the project's RILs. This report served as a living document, being updated after its initial submission (D3.1 submitted in M13), and is based on the implementation activities of the ScaleAgData architecture components.

Following the review of the second periodic report, the following changes were applied to the original document:

Section	Nature of change and reason (if applicable)
2.1.1	Yield estimation tool was updated with a status report (see 2.2.1.4).
2.2.2	Soil reflectance measurement was updated with a status report (see 2.2.2.3).
2.2.3	Soil hyperspectral processing module was updated with a status report (see 2.2.3.2).
2.2.4	Drought prediction from IoT and airborne sensor data was updated with a status report (see 2.2.4.3).
2.2.5	Drought prediction from satellite sensor data was updated with a status report (see 2.2.5.1).
2.2.6	Vegetation indices calculator was updated with a status report (see 2.2.6.1).
2.2.7	Agro-environmental policy indicators monitoring was updated with a status report (see 2.2.7.1).
2.2.8	DSS precision farming was updated with a status report (see 2.2.8.1).

2.2.9	Digital twin providing forecast and decision support was updated with a status report and details on the CMASS simulation model.
2.2.11	Milk quality and quantity forecaster was updated with a status report (see 2.2.11.1).
2.2.12	Grasslands improved biopars (LAI,fPAR) was updated with a status report (see 2.2.12.1).
2.2.13	Grassland primary production includes a series of updates to reflect the most recent developments alongside a status report.
2.2.15.2	AIM semantic translator for the Crop Management RIL was updated with a status report (see 2.2.15.2.1).
2.4	Research and Innovation Environment (RIE) was updated with subchapter 2.4.1 ScaleAgData catalogue and the purpose of the RIE was further clarified.

2. Methodology for Architecture Definition

The methodology followed to derive the ScaleAgData proposed system architecture is, on one hand, based on standard literature definitions and methods for architecture derivation[1,2] which suggest the use of 'views' and 'perspectives' for a holistic and successful description of the system components and functionality. On the other hand, architecture derivation relies on the ScaleAgData user scenarios and system specifications while also taking into consideration the already established process and functions of the RILs' infrastructure.

A well-established approach is to decompose the architectural description into views. Each view deals with a different aspect of the system. A formal definition of an architectural view is provided below: "A view is a representation of one or more structural aspects of an architecture that illustrates how the architecture addresses one or more concerns held by one or more of its stakeholders."[3]

The views that will be considered by the current architecture are the following:

- **Functional View**: defines and describes the system's functional components and their related functions and interfaces.
- **Information View**: defines the static information structure and presents dynamic information and data flows; in other words, it describes how to "define, structure, store, process, manage and exchange information"[4].
- **Deployment & Operation View**: handles installation, hardware, integration with existing infrastructure, and maintenance issues; it proposes selected technologies for the system deployment.

Additionally, perspectives are useful to describe non-functional features (sometimes referred to informally as "ilities") of the system. A formal definition may be found in "Understanding Architectural Perspectives"[5]: "An architectural perspective is a collection of activities, checklists, tactics and guidelines to guide the process of ensuring that a system exhibits a particular set of closely related quality properties that require consideration across a number of the system's architectural views."

2.1 Overall Architecture View

The ScaleAgData system consists of a variety of technological innovations that cover seven innovation areas/approaches: innovative sensor technology, edge processing, data sharing architecture and data governance, satellite data augmentation, data assimilation to service development, privacy-preserving technology, and data integration methodologies.

The aforementioned technologies and innovations leverage Earth Observation (EO), the Internet of Things (IoT), digital interfaces, advanced analytics, machine learning, artificial intelligence and business models.

At the same time, efficient mechanisms are employed in order to ensure interoperability with existing control systems, as well as improved accessibility and sharing of data through harmonised and

[1] Woods, E. (2005). Software Architecture Using ViewPoints and Perspectives. SET2005. Zurich.
[2] Michael, A. Ogush, D. C. (2000). A Template for Documenting Software and Firmware Architectures.
[3] Nick Rozanski, E. W. (2005). Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives.
[4] Magerkurth, C. (2012). IoT-A Deliverable D1.4 Converged architectural reference model for the IoT v2.0. IoT-A Consortium.
[5] Wood, E & Rozanski, N (2005). Understanding Architectural Perspectives

standardised means, whilst also demonstrating their uptake by relevant stakeholders for improved decision-making.

The following diagram (Figure 1) provides an overview of the high-level ScaleAgData architecture. The overall architecture follows a decentralised approach, where each RIL manages and processes its own data internally, from the initial data collection from the sensors at the edge to the data management and transformation. In this distributed way each RIL operates as an interconnected node inside a bigger network with other RILs. Each RIL can share data with each other directly and also with the Research and Innovation Environment (RIE) or any other external entities.

At the bottom layer is the deployment of the in-situ sensors. This is the primary collection data source for every RIL, together with EO data with which each RIL creates AI models and tools. In this layer the data types as well as the data transmission protocols may vary a lot for each RIL, as each may have its own instruments to measure different physical properties that depend on the lab's specific research focus.

On top of the sensor layer, there is the edge layer, whose main focus is to bridge the in-situ data collectors with the data management layer, inside each single RIL. This layer is located near the physical location of the sensors. The edge component can act as a simple gateway for the data management layer, supporting the protocols needed to collect data from sensors like Bluetooth Low Energy (BLE), Long Range Wide Area Network (LoRaWAN), and Wi-Fi. When necessary, in this layer real-time data processing components can be deployed to reduce the load to RIL internal systems and enforce data quality. It can also be used to deploy federated learning algorithms, which allows training of algorithms across multiple edge devices inside the same RIL.

The Data Management Layer is the component that stores and provides APIs for data consumption of other components within it. Data originating from the in-situ sensors are validated and stored in this layer and managed exclusively by each RIL. Depending on the complexity of the data, simple or more complex querying mechanisms may be included so that data can be easily provided to other components for further data analysis and research. It should be pointed out that only internal entities of the RIL can have access to the Data Management Layer. External parties should have access only through the Data Sharing Layer.

The Data Transformation Layer is crucial for the data sharing, as it is responsible for converting the data to a universally accessible format using the Resource Description Framework (RDF), which is the base of the semantic web. The primary function is to transform and enrich various datasets to a common format to ensure clarity and consistency. Given that each lab manages data using its unique data model internally, without this layer, an entity seeking data from different RILs would need to perform separate data conversions and may also be subject to misinterpretation.

The modules that are going to be integrated into the ScaleAgData system and may be common or specific to each RIL's research focus are the following:

- Yield estimation tool
- Soil reflectance measurement
- Soil hyperspectral processing module
- Drought prediction from IoT and airborne sensor data
- Drought prediction from satellite sensor data
- Vegetation indices calculator
- Agro-environmental policy indicators monitoring
- DSS precision farming
- Digital twin providing forecasts and decision support

- Reporting module
- Milk quality and quantity forecast
- Grasslands improved biopars (LAI, fPAR)
- Grasslands primary production
- Edge processing component
- Research and Innovation Environment (RIE)
- Data transformation – AIM semantic translator

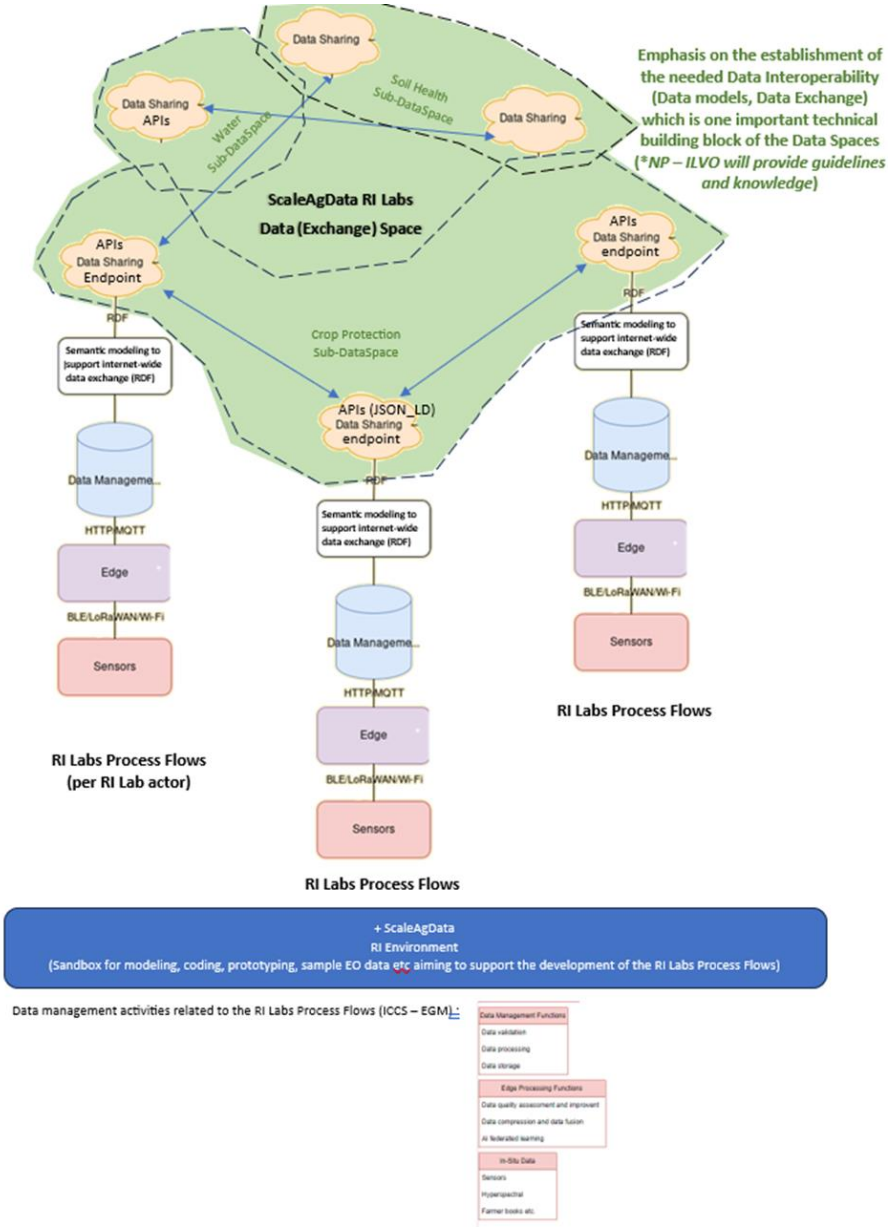


Figure 1: Overview of the proposed high-level ScaleAgData architecture.

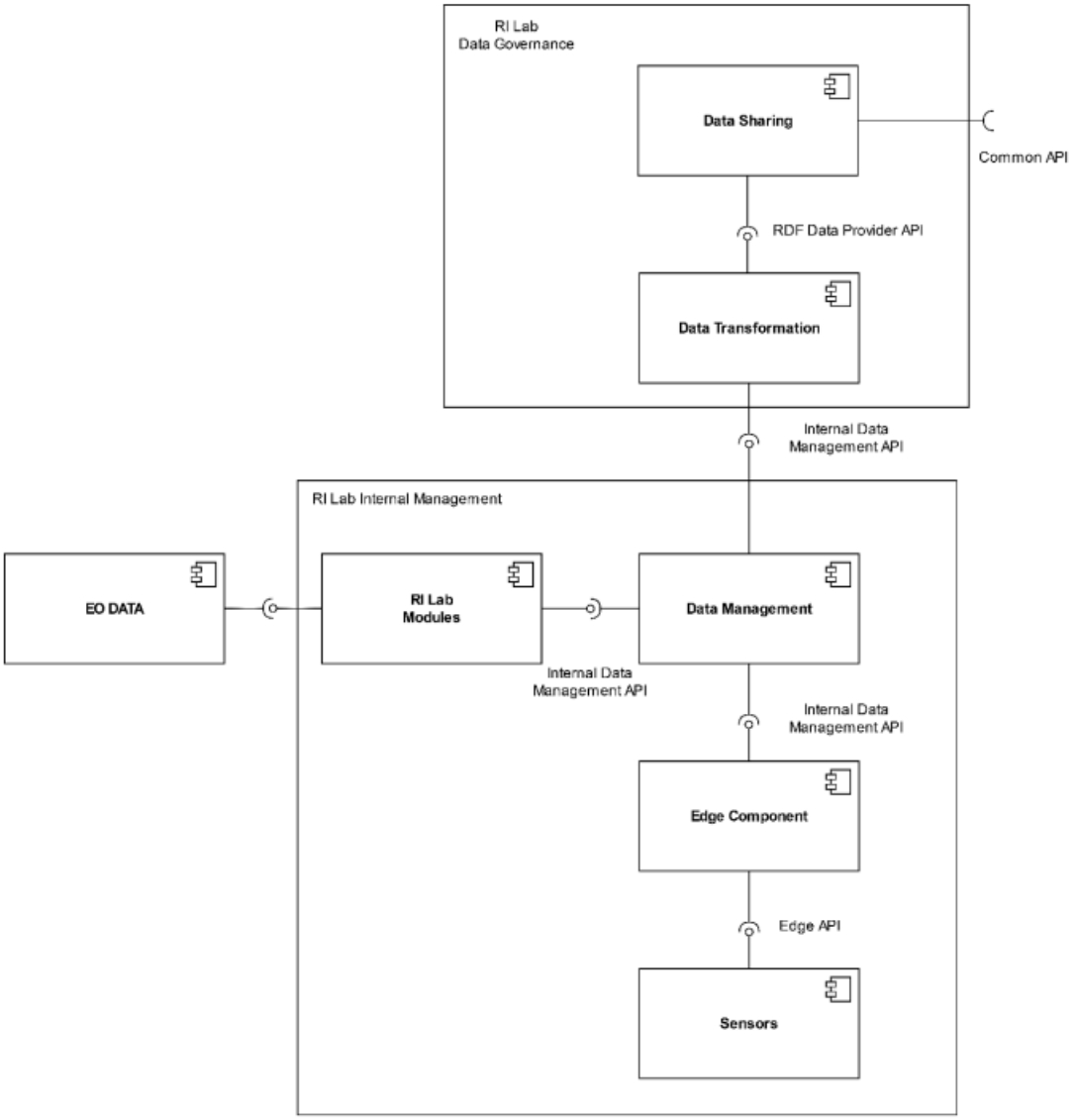


Figure 2: Functional view of the architecture of a node of the ScaleAgData system.

2.2 Functional View

In this section, we will present the modules that have been developed so far from the partners of the consortium. For the purposes of this deliverable, we decided to differentiate the tools into two categories. First, the ones that are based on machine or statistical learning and involve a training and an inference phase – we include a legend along with every architecture that indicates when an arrow is activated; we use the red colour for training, green for inference and black when an arrow is always used. Second, we have tools that are based on some physical simulation and for which there is no interesting differentiation; for these we always use black. Unless otherwise stated, this is the convention we will use in the rest of this section. Moreover, for arrows that start from the user as a command to fetch data, we indicate above them if they occur via an API call or a manual download. Finally, for the data we collect to train the model of a tool, we use dashed lines because they act as local storage. Remote source storages, e.g., Sentinel Hub, etc., are depicted with solid lines.

2.2.1 Yield Estimation Tool

The yield estimation tool is a machine learning-based tool that computes the yield production for a given (2D) area, using as input features images from Sentinel-1 and Sentinel-2 missions, soil and crop data, as well as weather data for the said area.

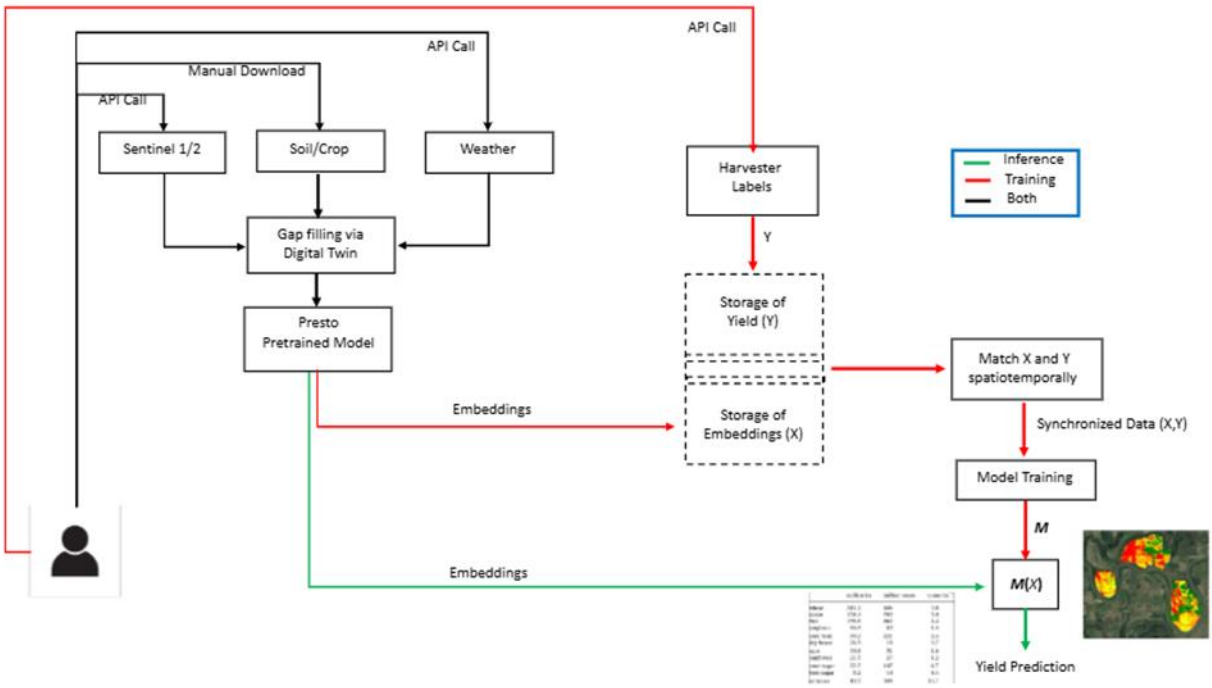


Figure 3: Yield estimation tool.

2.2.1.1 Data Collection: Sources and Access

Inputs and features (to be used in both the training and the inference stage):

a) Satellite images, in particular from satellite missions Sentinel-1 and Sentinel-2, can be queried through the API offered by the OpenEO platform.
b) Soil moisture and crop data are manually collected and provided by partners.
c) Weather data are accessed via the API of meteorological stations.
d) Labels (to be used only in the training stage) – Access through the API provided by the project partners; labels are collected N times per year.

2.2.1.2 Training

This model is trained using supervised learning. To do so, we have to create a dataset of pairs (X,Y) where X is the set of input features and Y is the corresponding label for that X. For a given area, i.e., for a single input data point, X, the following steps take place:

1) We collect the features X1, X2, and X3 from the corresponding sources.
2) A gap-filling procedure takes place where missing data (of the input features) are inferred with the use of a digital twin, provided by LUKE.
3) Now, the "full" feature (i.e., X) is passed through the pretrained model, Presto, which produces the embedding, which is a vector in a different space. These are stored locally.
4) We collect the yield values from the ground sensors, which serve as labels Y; these are also stored locally.
5) We pull the clean data, i.e., features X and labels Y, and match them temporally to make sure that the training is done in the correct way.

6) The synchronised data pairs (X, Y) are passed to the training algorithm; training takes place, and then the model M weights are computed and stored.

2.2.1.3 Inference

Once the model is trained, we can now use it for inference. For the moment our service works in an offline manner; namely, we collect the features, and we do a yield prediction on demand, that is, when the operator decides to.

For a given area, i.e., for a single input data point, X, we do the following.

1) We collect the features X1, X2, and X3 from the corresponding sources.
2) A gap-filling procedure takes place where missing data (of the input features) are inferred with the use of a digital twin, provided by LUKE.
3) Now, the "full" feature (i.e., X) is passed through the pretrained model, Presto, which produces the embedding.
4) We pass the full embedding X through the trained model M, which in principle returns a yield image, or a summary of the image statistics.

2.2.1.4 Status report

The service continues to operate in offline mode. Yield predictions generated from the model trained on harvester data from 2022-2024 are not yet sufficiently accurate. Additional harvester data from the 2025 season will be incorporated, together with continuous efforts to improve data filtering to further optimize the yield model.

2.2.2 Soil reflectance measurement

The soil reflectance measurement tool converts raw values from the VTT hyperspectral imaging sensor into calibrated soil reflectance data. It relies on supervised learning methods and a specific field setup.

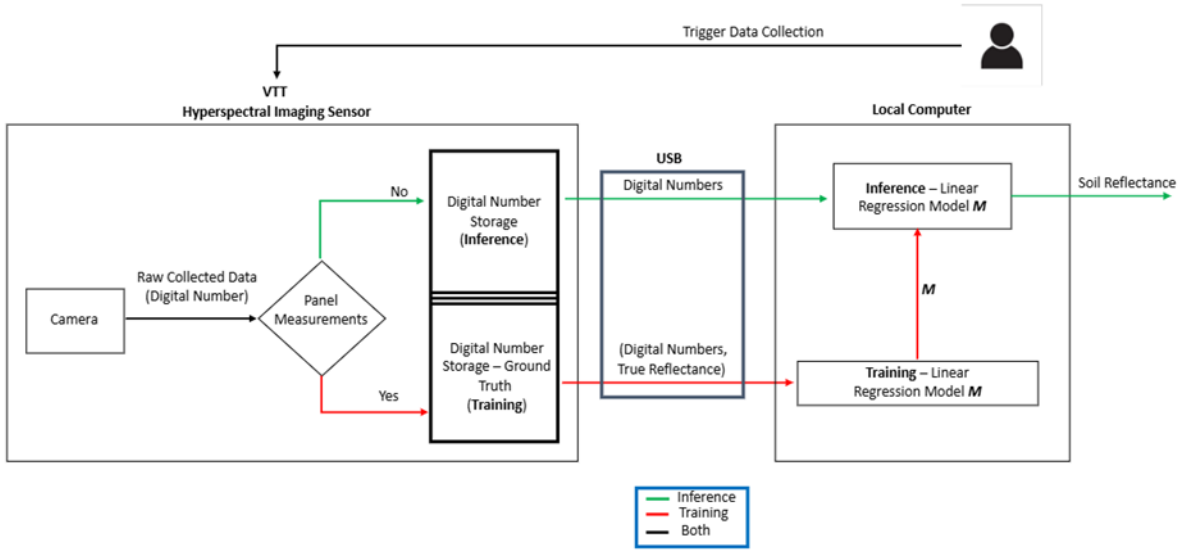


Figure 4: Soil reflectance measurement tool.

2.2.2.1 Data access and sources

The data collection process is manual; it involves an operator who moves through the field holding the sensor to get measurements. Below, we explain how the tool works by breaking down its operation into two stages, training and inference.

It is worth mentioning that since the relation between digital numbers (what the camera collects) and reflectance can change from measurement session to measurement session, the training and the inference essentially take place in the same session.

2.2.2.2 Data collection from training and for inference

1) The camera collects the raw data, whose format is a digital number.
2) If the measurements are on panels (see path below), the digital numbers are stored. The key calibration step is to include panels with known, stable reflectance values (50%, 75%, 95%). So, the digital number measurement we get for a panel of, e.g., 50% reflectance, will give us directly a pair (digital number, 0.5). Doing this for all the panels grants us access to multiple pairs of (X,Y) that we can use for supervised learning. Alternatively (see path above), the digital numbers are also recorded for later conversion to reflectance (this refers to inference).
3) Once the data collection is done, the operator transfers two sets of data:
    a. The ones for inference only
    b. The ones for training, which are in the form (X, Y) = (Digital Numbers, True Reflectance), are transferred by a USB to the local computer.
4) Training: The operator trains a linear regression model using the pairs of (X, Y).
5) Inference: Using the model M, we convert the digital number values we collect to predict reflectance, applying the model reflectance = M (digital number).

Note: Steps 1 and 2 take place within the sensor, while 4 and 5 are on the local computer.

2.2.2.3 Status report

The soil reflectance measurement tool is fully implemented and operational. The radiometric calibration pipeline, converting raw digital numbers from the VTT hyperspectral imaging sensor to calibrated reflectance values using reference panels (50%, 75%, 95%), has been completed and validated in field conditions. The implementation is available in the project repository at https://github.com/ScaleAGData/SOIL-RILAB, specifically within the VTT HSI sensor/ module (python_functions.py), which contains the calibration routines, and VTT_analysis_2026_train.py, which handles model training.

2.2.3 Soil hyperspectral processing module

The soil hyperspectral processing tool aims to infer the soil property of an area given its satellite image. To achieve this, we developed a supervised learning approach by building three models, where each model uses a different data source. In particular, the first model (M1) is trained to predict the soil property from Sentinel-2 images, the second one (M2) from PRISMA images, and the last one (M3) from Kuva images.

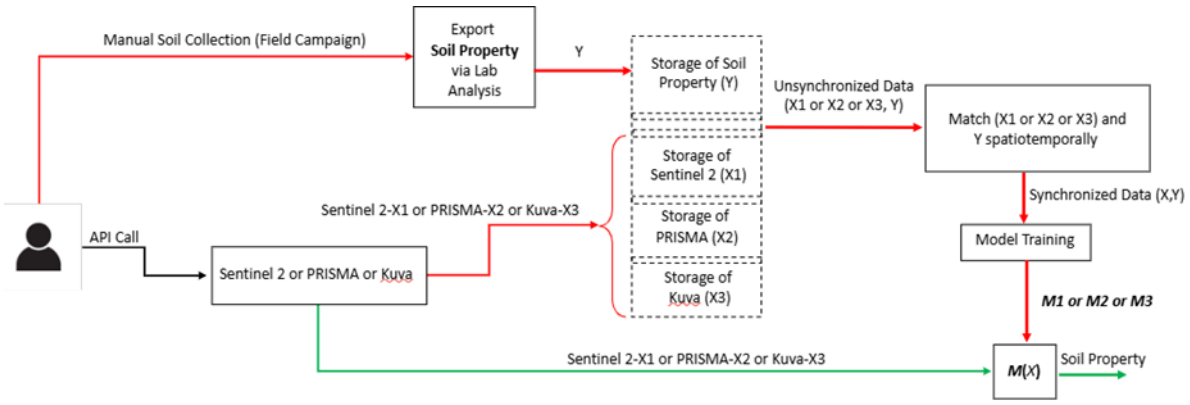


Figure 5: Soil hyperspectral processing module

2.2.3.1 Data Collection, Access and Sources

Input features are collected via API and are stored in a local machine:

1) Sentinel-2 images: Collected by the Sentinel Hub platform.
2) PRISMA images: Collected by the PRISMA Hub platform.
3) KUVA images: Collected by an API our partner provided.

Labels: We acquire soil property labels by conducting field campaigns, meaning that, first, we collect soil from specific areas; second, we analyse these measurements to convert them to the desired property; and third, we store the values on our local machine.

2.2.3.2 Status report

The soil hyperspectral processing module has been fully implemented, though with an important methodological revision relative to the original plan described in the grant agreement. Rather than treating Sentinel-2, PRISMA, and Kuva Space as three independent models (M1, M2, M3), the implemented approach adopts a **data fusion strategy** that combines VTT ground/airborne hyperspectral data with Kuva Space L2A satellite imagery and Sentinel-2 multispectral data into an integrated multi-source prediction pipeline for Soil Organic Carbon (SOC). PRISMA has been replaced by Kuva Space as the primary hyperspectral satellite source, in line with the project's objective to combine sensor data with satellite images for optimal soil parameter estimation (section 1.1.3.7). Four feature configurations are evaluated (VTT only; VTT + Kuva; VTT + Sentinel-2; VTT + Kuva + Sentinel-2), with bare-soil filtering applied via NDVI thresholding. Random Forest and XGBoost models are trained across all configurations using 5-fold cross-validation, with full-image SOC prediction maps generated as GeoTIFF outputs. The full implementation is available at https://github.com/ScaleAGData/SOIL-RILAB, under the data fusion soillab/ directory.

2.2.4 Drought prediction from IoT and airborne sensor data

The drought prediction airborne tool, with the help of a soil moisture sensor, is a system that has two goals: a) create irrigation alerts regarding a field that we monitor and b) compute maps with the soil type; the two outputs are computed independently.

2.2.4.1 Drought prediction with soil moisture IoT sensor

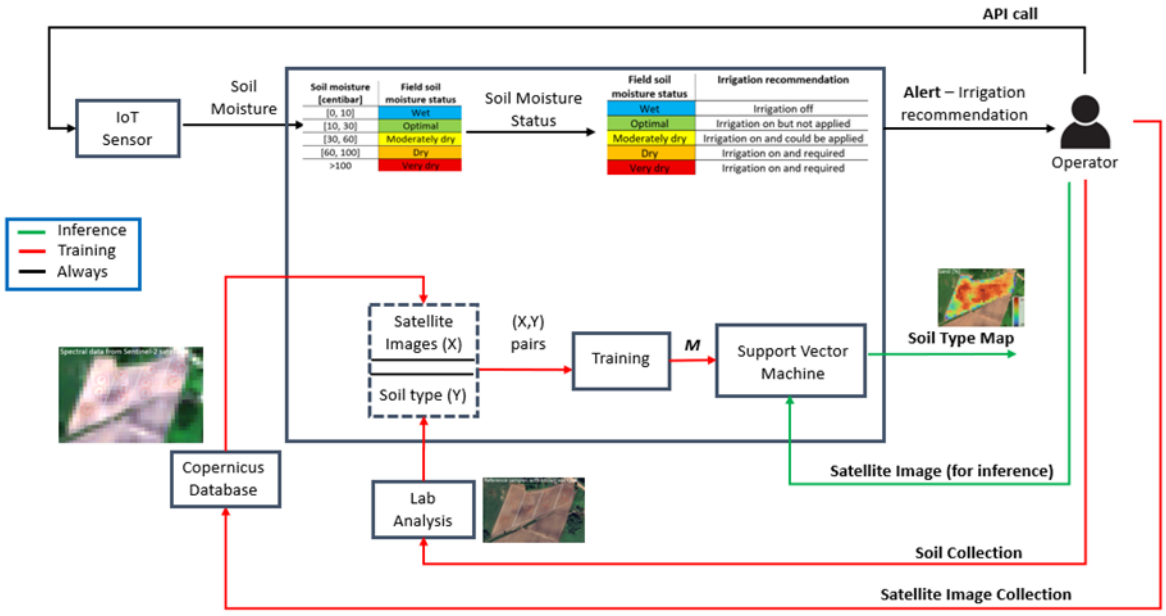


Figure 6: Drought prediction with soil moisture IoT sensor

2.2.4.1.1 Data collection, access and sources.

Irrigation recommendation alert:

- Input: We access the soil moisture measurements from an IoT sensor through its API.

Soil type map:

- Input: We get the satellite images from the Copernicus database through its API.
- Labels: We collect soil from the same areas in order to produce the labels for the learning process.

Irrigation recommendation alerts.

The system works in a deterministic way and does not have any machine learning components and works in the following way. The operator is essentially making an on-demand request to an IoT sensor to get soil moisture measurements from an API. After that, via a deterministic mapping, the soil moisture measurement (measured in centibars) is used to classify the soil moisture status – e.g., from [0,10] cb the status is "Wet", from [10,30] cb is "Optimal", and so on. The status is passed to a second deterministic mapping that, from a status, gives back the irrigation recommendation alert to the operator. E.g., when the status is "Wet", it returns "Irrigation off"; when "Dry", it returns "Irrigation on and required", and so on.

Soil type map

The operation of this part into the operation training and inference. Each will be described in terms of steps.

- Training:
  1) The operator collects satellite images (X) from the Copernicus database.
  2) Soil is also collected, analysed and finally characterised in a lab environment – this creates the label (Y) we need for training.
  3) Data are spatiotemporally matched and then are used to train a support vector machine model which basically solves a classification problem by predicting discrete classes, e.g., "sandy", "clay", "loam", "silt", etc.

Inference

This part is rather straightforward – we download images for an area of interest, and we pass it from the SVM model we have built, which returns a map of soil types.

2.2.4.2 Drought prediction from weather data

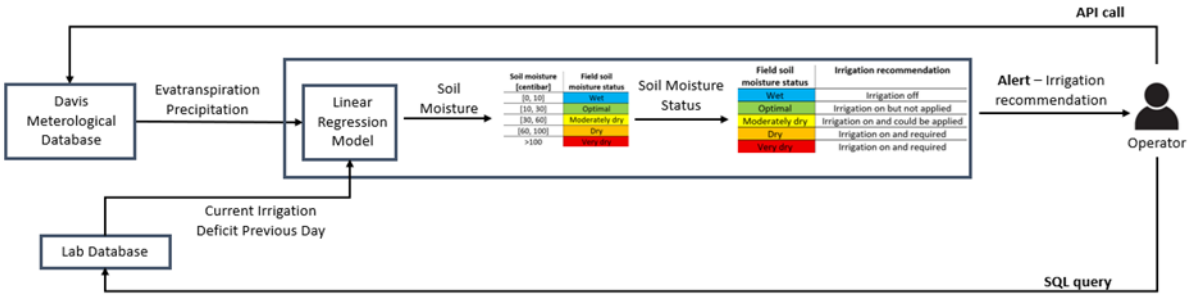


Figure 7: Drought prediction from weather data

We have built a tool that creates irrigation recommendation alerts to the operator – without the need for IoT soil moisture sensors. Essentially, the main issue here is that in the absence of a soil moisture sensor, it is necessary to estimate the soil moisture from other quantities that we have available. After some analyses we made, we found that the water deficit has a high correlation with the soil moisture, which is in fact quite reasonable. To this end, it was found that a simple linear regression statistical model suffices to predict the soil moisture value given the water deficit. So, the key quantity we need to compute is the water deficit, and it can be computed in the following simple manner:

$D_{t} = D_{t-1} + ET_{t} – P_{t} – Irr_{t}$

Where:

D_{t} Current day soil water deficit (to be calculated for each day)

D_{t-1} Previous day's soil water deficit (D_{t} from previous day, set to 0 if negative, assuming that all water has been runoff or absorbed by deeper soil layers)

ET_{t} Evapotranspiration (calculated from meteorological sensor data)

P_{t} Precipitation (measured by meteorological station)

2.2.4.3 Status report

Both drought prediction approaches were implemented in a dashboard provided to the farmer for field water status assessment. Data collected by the IoT sensors were stored on a local server and used as inputs for the models. The field water status was displayed on the farmer's desktop in near-real time, alongside other weather and soil parameters of interest. Based on the provided data, the farmer was able to make informed decisions regarding irrigation application to the pappaermint crop. However, the main obstacle in testing the utility of the provided data for decision-making was the unfavourable weather conditions for irrigation in Latvia during 2025 – precipitation exceeded the climatic norm by a factor of 2 throughout the entire vegetation season, and the farmer lost the full peppermint yield. Nevertheless, the farmer is interested in continuing the IoT-sensor-supported irrigation decision-making experiment during the 2026 season with a different crop – chamomile instead of peppermint.

2.2.5 Drought prediction from satellite sensor data

The drought prediction from the satellite tool aims to compute irrigation recommendation alerts to an operator. It does, in fact, upscale the procedure with the IoT soil moisture sensor we described earlier. It works in the following way. The operator makes an API call to the DHI soil moisture satellite database, which basically gives back a 2D map of the area with soil moisture measurement per pixel.

Then, soil moisture (measured in centibars) is deterministically classified as a status, and the latter becomes an irrigation recommendation.

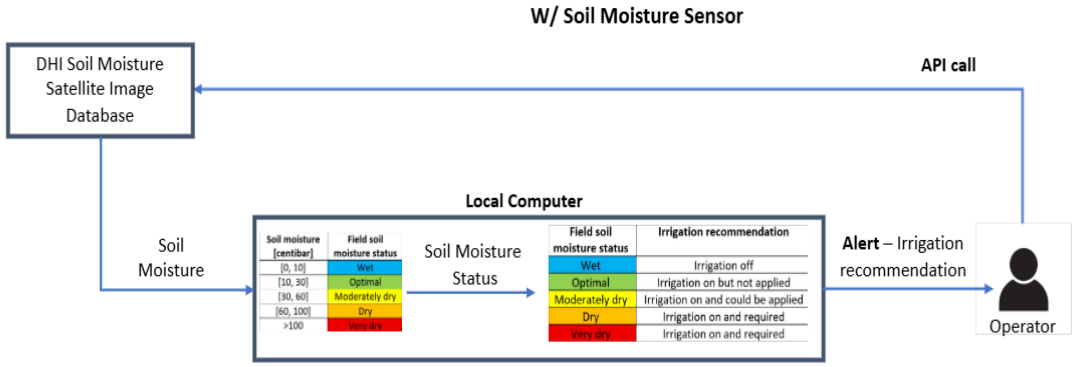


Figure 8: Drought prediction from satellite sensor data

2.2.5.1 Status report

Soil moisture and evapotranspiration data products were provided by DHI at the end of each growing season (2024 and 2025). Consequently, it was only possible to evaluate their performance for field water status assessment using archived data. For the 2026 growing season, DHI has agreed to provide partners with near-real-time soil moisture and evapotranspiration data, thereby enabling EO-based field water status assessment within the dashboard provided to the farmer. It is planned to test this approach for chamomile field water status assessment during the 2026 growing season and to compare the results obtained with predictions derived from IoT sensor data.

2.2.6 Vegetation indices calculator

The Normalised Difference Vegetation Index (NDVI) module is a tool developed to give us back NDVI statistics from satellite images that are collected from the Sentinel Hub. It works as follows. The operator writes a simple function call in the form getNDVI (from, to, location) – the NDVI module transforms this to a proper API call to the Sentinel Hub. Then, the NDVI images that correspond to the "from, to, location" are sent back to the NDVI module, where an aggregation operation takes place. The output of this function produces the NDVI statistics, which are returned to the user/operator.

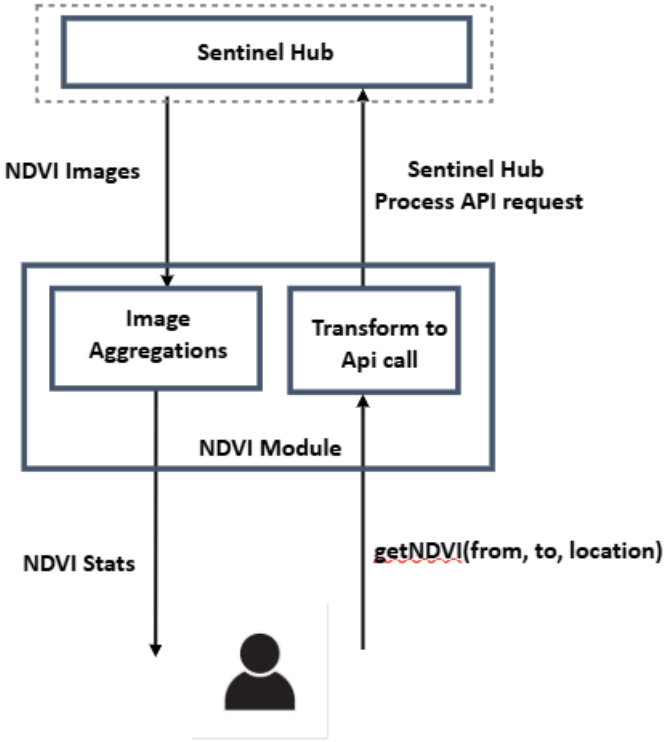


Figure 9: Vegetation indices calculator

2.2.6.1 Status Report

The vegetation indices calculation methodology is finalised and operational as a subcomponent of the crop classification product. The full multi-index stack (NDVI, NDRE, NDWI, and PSRI) is implemented through Sentinel Hub Evalscript workflows and runs in production for both Thessaly and Crete pilot areas, with daily temporal aggregation across the 2023, 2024, and 2025 growing seasons. The masking logic (denominator-zero exclusion, SCL water masking, and Sentinel Hub dataMask filtering) is applied consistently across all indices to ensure clean inputs to the classification step. The corresponding code is part of the crop classification example case being prepared for upload to the project's GitHub repository, accompanied by a README that documents the band selections, masking rules, and aggregation parameters so that the full indices-to-classification chain can be independently reproduced.

2.2.7 Agro-environmental policy indicators monitoring

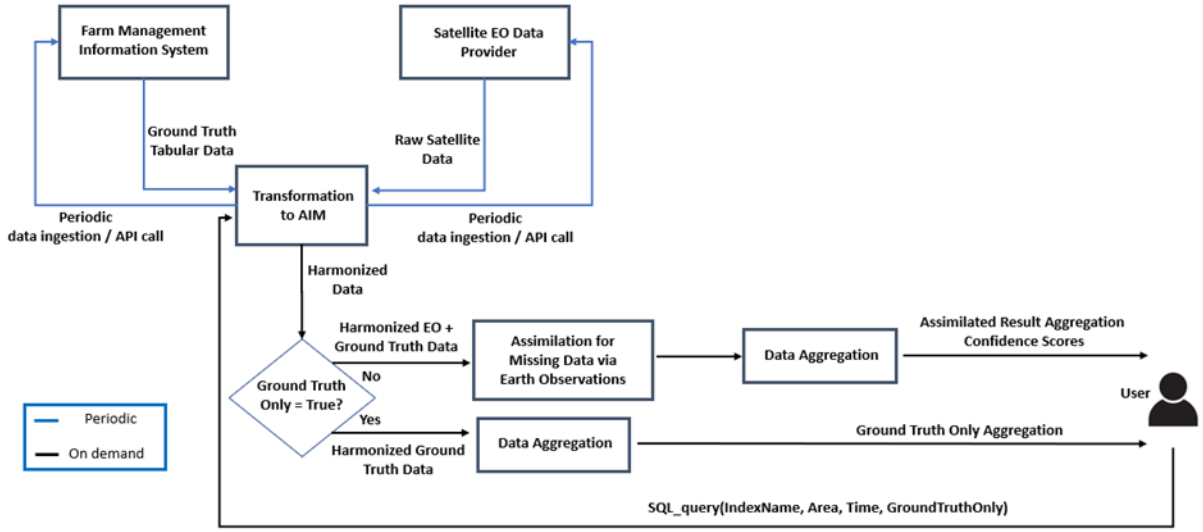


Figure 10: Agro-environmental policy indicators monitoring

The agro-environmental policy indicators tool is an API service that lets users (e.g., policy analysts, regional authorities, dashboards) request agro-environmental performance indicators for a given region + indicator type + time period.

The system's target is to return calculated metrics (together with their confidence levels) based on a mix of:

- Farm data (from FMIS/digital farm books)
- Satellite EO data
- Assimilation & inference algorithms to fill data gaps
- Privacy & governance mechanisms

Throughout the entire data processing chain, robust data governance and privacy mechanisms ensure that all information is collected, processed, and shared in compliance with relevant legal and ethical frameworks (e.g., GDPR). Farm-level data is anonymised or aggregated before use, and strict access control and provenance tracking are applied at every stage. These measures guarantee that sensitive farm information remains protected while allowing reliable, region-level agro-environmental indicators to be derived and disseminated securely.

Operation of the system

The AIM transformation block performs periodic data ingestion API calls to a) various Farm Management Information Systems (FMIS) – here we only depict one; and b) Satellite Earth Observation data providers (e.g., Sentinel Hub). Then, the AIM transformation block gets the data from these two sources and harmonises them to a standard format. This way the system stays up-to-date in an automated way. Now, on the end-user side of things, the user makes a query in the form "**SQL_query(IndexName, Area, Time, GroundTruthOnly)"** where the IndexName is the physical quantity we are interested in and GroundTruthOnly is a binary variable that indicates whether the user wants to receive measurements from FMIS (i.e., ground truth measurements from in-situ sensors) or assimilated measurements that combine/fuse FMIS data with satellite earth observations.

2.2.7.1 Status report

The policy indicators methodology is finalised and deployed in a live, publicly accessible dashboard built on Python and React.js, hosted on NP infrastructure at https://scaleagdata.neuropublic.gr/. The

dashboard allows users to query agri-environmental indicators by region and crop, drawing on two full growing seasons of processed data. Pre-processing of the 2025 pesticide sensor data has been completed for both growing seasons, and integration into the dashboard is the immediate next step. A dedicated RIL2a repository has been created on the project's GitHub (https://github.com/ScaleAGData), which is now public. The full dashboard code, together with the associated Sentinel-2 crop classification GeoTIFFs and configuration files, are also uploaded. Architecture diagrams (both a high-level overview and a detailed component and data-flow diagram) are published on the repository's main page. As a validation metric, we have calculated the Training Area Coverage ratio: the proportion of ground-truth area used for training relative to the total area subject to classification, which provides a first-order reliability indicator for the classification outputs by region and crop. Integration of this metric into the dashboard is planned as a next step. The complete code package, with README and sample data, will be finalised and committed to the repository imminently.

2.2.8 DSS Precision Farming

The Decision Support System (DSS) computes predictions for two quantities: a) the water balance, as well as b) the yield prediction.

Regarding the water balance, when the users need a prediction, they make three API calls, to a weather database to query data for precipitation, to the irrigation schedule and to the soil texture. The three inputs are combined in a simulation model that predicts the water balance.

As for the yield prediction, the workflow integrates both manually collected field information and remotely sensed data to drive a crop simulation model for yield prediction. Users provide key input variables such as crop variety, sowing date, phenological observations, weather record, geographical location, and details of fertilisation interventions through manual data entry. In parallel, satellite-derived vegetation indices (Airbus NDVI) are retrieved automatically via an API call. Both data streams feed into the simulation model, which processes the combined information to generate yield predictions . This approach allows for the integration of localised management data with spatially continuous remote sensing information, improving the accuracy and scalability of yield estimates.

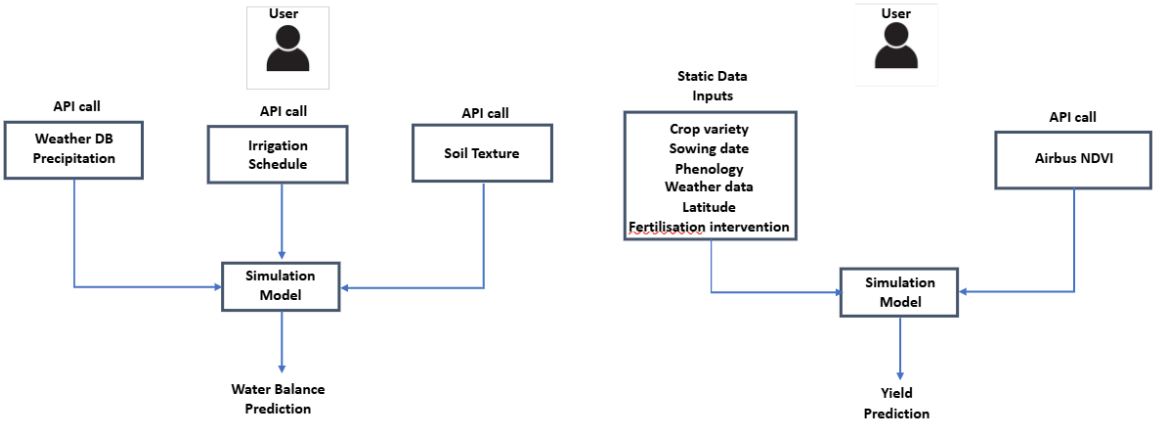


Figure 11: DSS precision farming

2.2.8.1 Status report

The water balance functionality is implemented in the decision support system for the sustainable management of wheat. For yield prediction, the model integrating NDVI data and in situ collected

information still runs locally. Analyses were carried out to assess the potential improvements in predictions generated by the integration of NDVI data as an input model with respect to the model running only on in situ collected data. The analysis highlighted that prediction improvements were significant only in cases in which the crop canopy was heavily disrupted by accidents, which are not usual conditions.

2.2.9 Digital twin providing forecasts and decision support

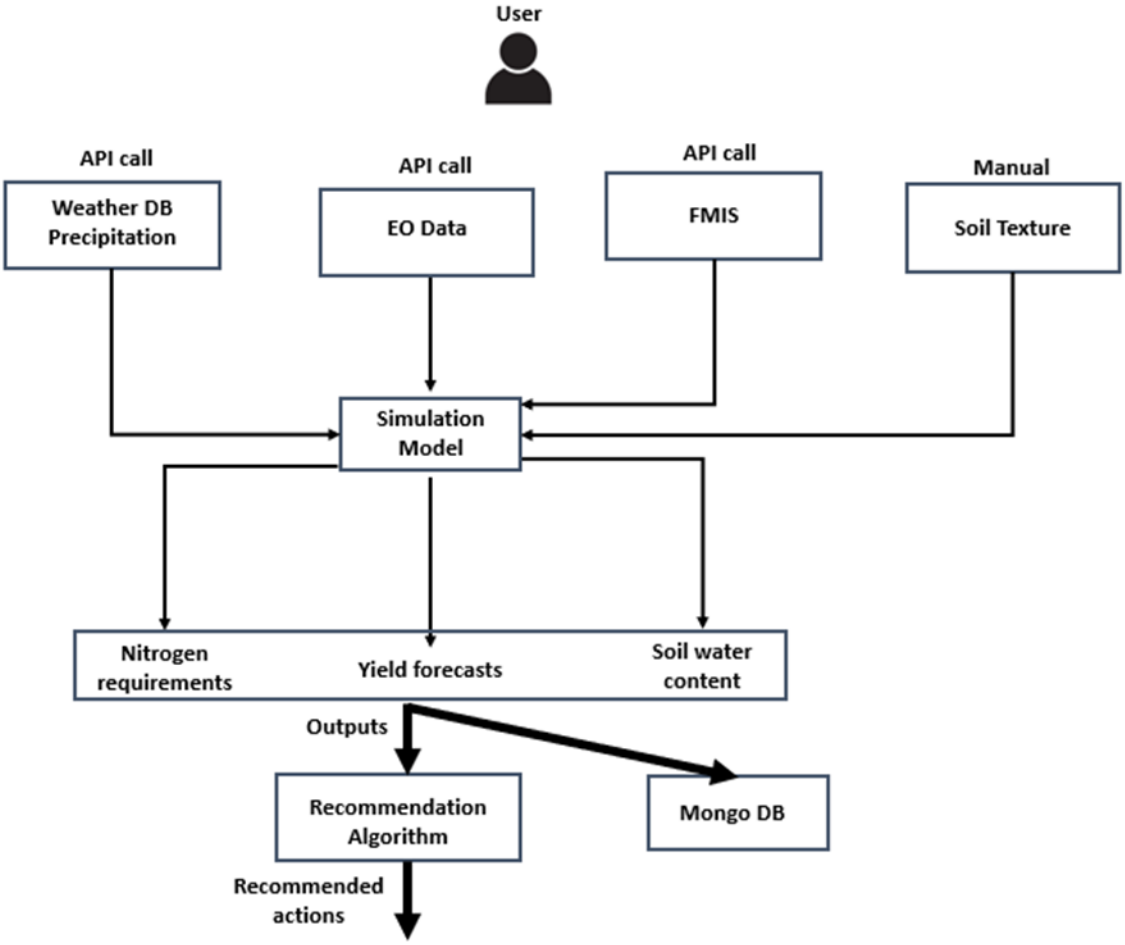


Figure 12: Digital twin providing forecasts and decision support

CMAAS Simulation Model

A simplified wheat crop model and a digital twin service were developed to combine crop management data, weather data and up-to-date remote sensing data for estimating the status of the crop and forecasting potential yield and N uptake. The service is called "Crop model as a service" (CMAAS). The model was developed with the aim to include a minimal set of calibration parameters needed to support variable rate N application (VRNA) and irrigation decisions during the growing season based on data available from commercial farms. The model is described in more detail in D4.4.

The user posts spatial data about fertiliser application, crop leaf area index time series obtained from remote sensing and local weather data. The service returns a yield forecast and forecast of potential nitrogen uptake of the crop until the end of the growing season. This information can be used to create maps for fertiliser application tasks. APIs enable a systematic and user-friendly approach to calibrating and running the models.

The CMAAS crop model described in this section was implemented in Kotlin programming language, and a REST API for calling the model was developed. The entire model and API was packaged as a single JAR file which can be executed using a Java runtime. The CMAAS service includes endpoints for forecasting, data assimilation and model calibration. The endpoints are documented using the OpenAPI specification, and API documentation is packaged in the service and served under its own endpoint.

In addition to the user being able to use their own weather data from, e.g., their own weather station, an integration to Open-Meteo weather service was developed. A location-based API was added to the CMAAS service where the user only needs to post the management data and the location, and the weather data is retrieved from Open-Meteo in the background. The weather service integration retrieves historical weather data and short-term weather forecasts. Additionally, crop model scenarios based on historical weather can be run to estimate the uncertainty of the model forecasts.

Python Library

The Python library farmingpy (https://github.com/TwinYields/farmingpy) serves as the data interface and orchestration layer around the CMAA simulation. It automates data retrieval and processing and defines the data model of the digital twin. Farmingpy retrieves all model input data for the chosen time and area, including:

- Daily weather data and forecasts
- Daily Earth Observation (EO) data (e.g., reflectance, vegetation indices, LAI, evapotranspiration, phenological stages) Daily soil sensor measurements (if available)
- Farm management data (e.g., machinery task files or FMIS records)

Besides using the retrieved data as model input data, it can be used for calibration of the model or data assimilation during the simulated season.

Interfaces for Inputs and Outputs

Input and output variables of the digital twin are described as entities defined by the data model. All digital twin entities can be transferred through the NGSI-LD context brokers to enable easy integration with external services or dashboards.

Main Services

Service Description:

- Get weather data. Retrieves daily weather data and forecasts if available.
- Get EO data. Retrieves spatial EO layers (reflectance, vegetation indices, LAI, evapotranspiration estimates, and phenology stages) for the selected time and area.
- Get soil sensor data. Retrieves daily soil sensor measurements from the RIL if available.
- Get farm management data. Retrieves farm operation data from FMIS or machinery task files.
- Run daily CMAAS simulations.

2.2.9.1 Status report

The CMAAS service has been packaged into a service which is offered to the different labs through the projects' internal Github repository. Jupyter notebooks for interfacing the API have been developed. The service is ready to be deployed by the RILs, and the model has been calibrated for winter wheat in Belgium. The validation of the accuracy of the data assimilation service is in progress. The model licensing discussions are in progress with the consortium.

2.2.10 Reporting module

The reporting module is an automated system designed to generate comprehensive reports that summarise crop health and environmental conditions across monitored areas. It integrates data from multiple sources – including Earth Observation (EO) imagery, in-situ sensors, and weather datasets – to detect and visualise anomalies such as early crop stress, NDVI deviations, and temperature or soil moisture irregularities. Upon identifying such conditions, the module compiles the findings into a structured PDF report and automatically distributes it to assigned recipients via an e-mail, supporting timely decision-making.

2.2.11 Milk quality and quantity forecaster

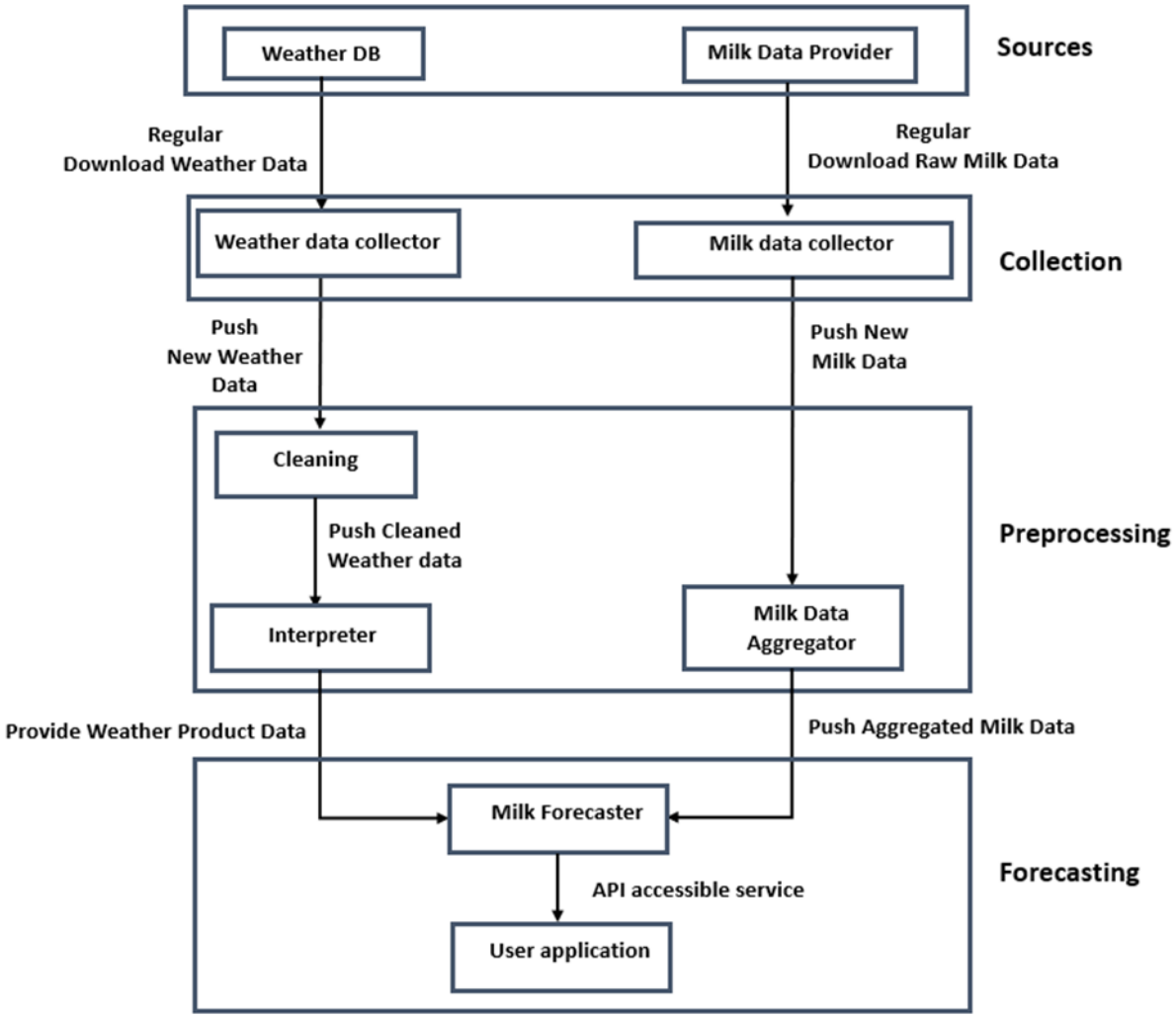


Figure 13: Milk quality and quantity forecast

The main objective of the Dairy Lab is to explore ways to improve forecasting of milk quality and quantity, which is critical for the dairy cooperative's production facilities. If successful, the architecture described here could be implemented in a future project phase. The system integrates satellite Earth Observation (EO) data and milk delivery data to provide actionable insights.

EO Data Provider: Existing public sources, such as Sentinel Hub, provide satellite imagery via APIs. EO data include multiple bands and indices that can be used to monitor grasslands and feed availability.

Milk Data Provider: A service (not yet fully implemented) will supply milk delivery data, including quality metrics (fat/protein percentages) and quantity for a given geographical area and timeframe.

Data Collection Layer: A Python-based data collection system regularly retrieves EO data and milk data according to configurable schedules. The EO collector pushes new imagery into pre-processing pipelines, while the milk collector feeds data into aggregation components. Configuration options include the collection interval, API URLs, authentication credentials, relevant bands, and area of interest.

EO Data Cleaner: Raw satellite images are filtered and cleaned to remove unusable data, such as cloud-covered images, ensuring only relevant observations are processed.

EO Data Interpreter: Cleaned EO data are processed into products, such as NDVI, NDMI, or more advanced metrics like biomass growth estimates. In the future, machine learning or AI-based processing may be applied here to derive vegetation growth or forage availability, but concrete EO products for milk forecasting are still under investigation.

Milk Data Aggregator: Raw milk delivery data are aggregated over geographical areas, e.g., farm groups supplying a processing plant. Aggregated outputs include average milk quality (fat/protein) and total quantity.

Milk Data Forecaster: This service is intended to produce forecasts of milk quality and quantity, using the latest EO and milk data. While the architecture anticipates the potential use of ML models for prediction, the current documentation does not provide details of any implemented ML algorithms. It appears that forecasting logic is still largely under design, pending correlation analysis between historical EO data and milk records.

End-User Application: The final outputs feed decision-support tools for dairy processor staff. The application would display forecasts via a graphical interface. Detailed functionality will be defined in the next project phase, depending on the success of the ongoing analysis.

Summary of the Pipeline
EO and milk data → collection → cleaning and aggregation → forecasting → decision support. This pipeline aims to provide timely, data-driven insights to optimise production and milk quality management.

2.2.11.1 Status Report

The overall milk quality forecaster was implemented for testing and validation, making use of daily data updates for a region between Bremen and Hamburg in Northern Germany. Data acquired in the scope of milk deliveries and laboratory results of checking the milk quality are provided in real-time. The dashboard with the milk quality forecast is currently validated by the core team in the dairy cooperative to check its use in daily operation. Since the dairy cooperative was already highlighting their interest in a larger prediction time horizon (i.e., longer than 14 days), additional meteorological forecast data is currently being analysed to see if this will still allow reasonable forecasting accuracy. At the same time, the Dairy Lab is currently preparing the provision of the weather data provider via GitHub. This shall enable an extended use of the module also by an extended target audience.

2.2.12 Grasslands improved biopars (LAI, fPAR)

Leaf Area Index (LAI) provides critical information on vegetation structure, yet its optical retrieval is often interrupted by clouds, which frequently cover large areas in Central Europe. To overcome this limitation, Sentinel-2 LAI values are used as reference labels to train a machine learning model that links radar and environmental signals to vegetation conditions. This approach allows accurate LAI predictions even during cloudy periods, effectively filling gaps in the optical time series.

The tool described in this subsection is a machine learning system that computes LAI from satellite data. Since it is ML-based, its operation is described step by step for training and inference.

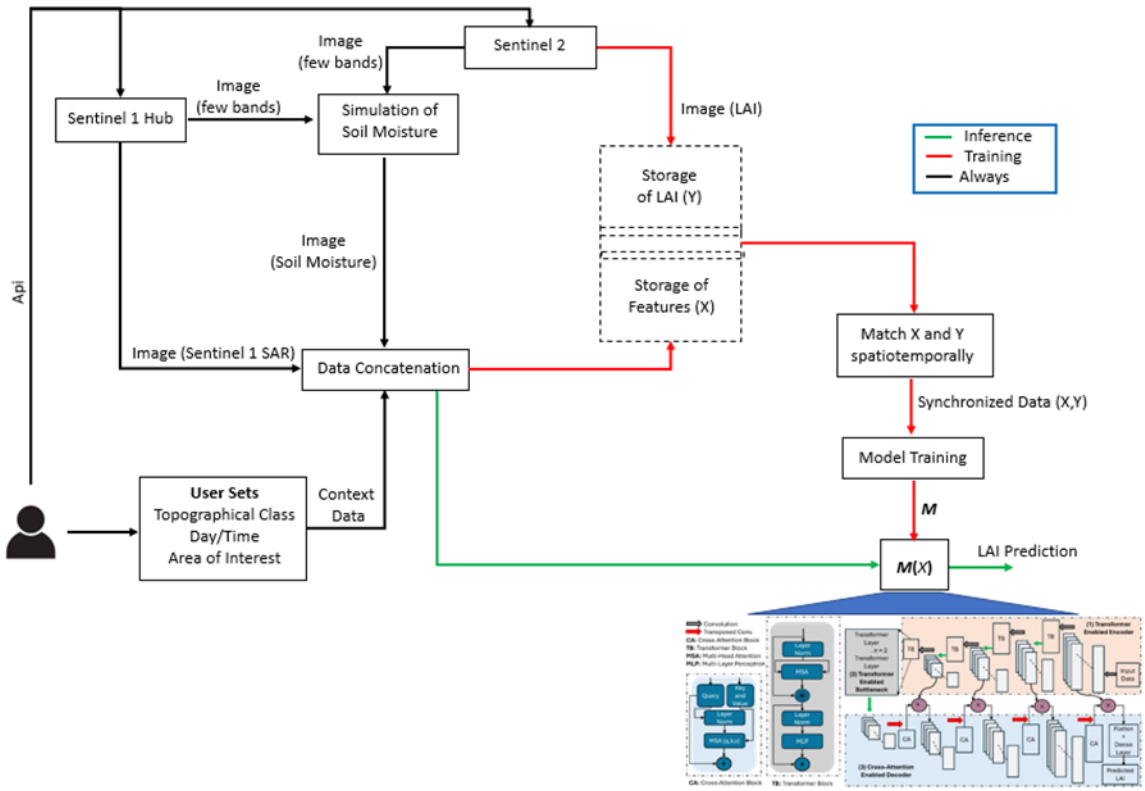


Figure 14: Grasslands improved biopars

Training:

Step 1: The user manually downloads Sentinel-1 and Sentinel-2 images.

Step 2: Selected bands from Sentinel-1 and Sentinel-2 are extracted and used as input along with simulated soil moisture.

Step 3: The user provides topographical class, day-of-the-year, and area of interest; these context variables help the model interpret the environmental conditions.

Step 4: Soil moisture, Sentinel-1 data, and context data are concatenated into a single input variable X, which will be fed into the model.

Step 5: A by-product of Sentinel-2 (i.e., a calculation derived from multiple bands) is used to compute LAI maps, which serve as labels Y for supervised learning of the transformer model.

Step 6: Training pairs (X, Y) are matched spatiotemporally, and the synchronised data are used to train the model.

Inference

Steps 1–4: Same as in training.

Step 5: The input variable XXX is sent directly to the trained model M, which returns the predicted LAI M(X).

2.2.12.1 Status report

The tool is complete in its current implementation and publicly available at the project repository https://github.com/ScaleAGData/Grassland-RILAB TransConvRegressor directory. This repository implements a 1D TransUNet-based regressor for Leaf Area Index (LAI) estimation using temporal Sentinel-1 backscatter signals and ancillary features. It includes training, evaluation, and prediction pipelines implemented in Python. The so-called S1 LAI time series has already been validated against Sentinel-2-derived LAI with standard statistical metrics, obtaining not completely satisfactory accuracy. The S1 LAI time series will be evaluated alone and in combination with Sentinel-2 LAI time series against in situ data collected in the first iteration of the project, in order to estimate the accuracy of the developed tool in enhancing the LAI estimation and its effect on the productivity estimation.

2.2.13 Grasslands primary production

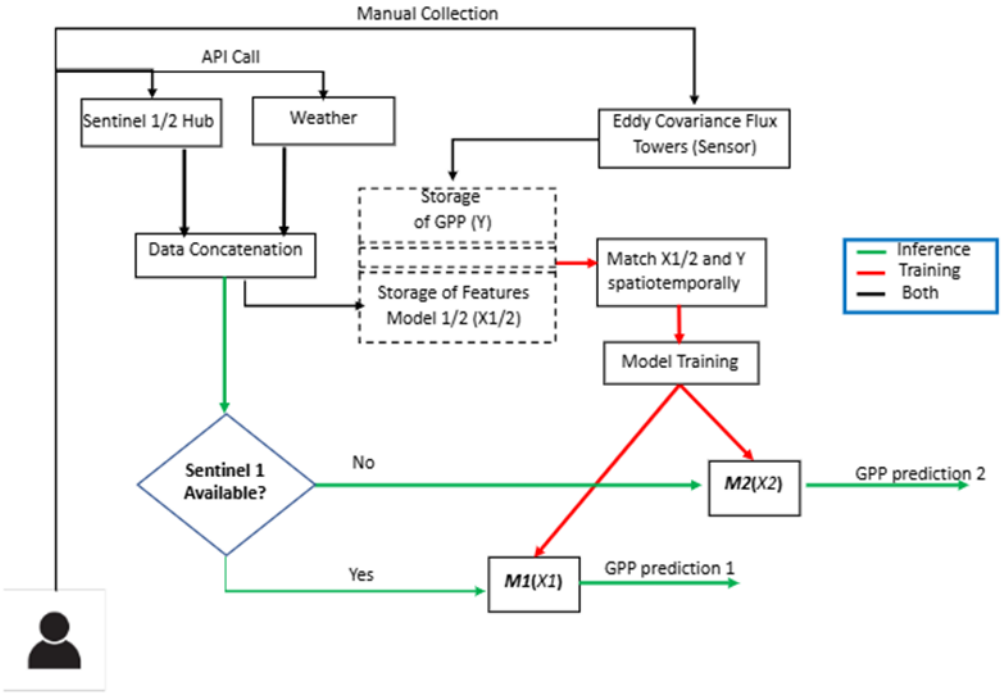


Figure 15: Grasslands primary production

2.2.13.1 Improved grassland GPP maps based on flux tower sensors

Gross Primary Production (GPP) is a key indicator of ecosystem productivity, but continuous monitoring with optical satellites is often limited by cloud cover. To overcome this, in-situ Eddy Covariance measurements are combined with satellite and contextual data to train machine learning models that can predict GPP even under cloudy conditions. Two separate models are maintained — one for cloud-sensitive Sentinel-2 optical data and one for radar-based Sentinel-1 — with the model selected at inference depending on data availability.

A third model using CropSAR2D data was developed.

CropSAR2D is an advanced satellite data product that provides high-resolution, gap-filled time series of biophysical vegetation parameters by fusing optical and radar observations. Developed primarily to overcome the limitations of cloud cover in traditional optical remote sensing, CropSAR2D integrates multispectral data from Sentinel-2 with Synthetic Aperture Radar (SAR) data from Sentinel-1. This data

fusion is achieved through a deep learning framework that leverages the all-weather capability of radar to "fill the gaps" in optical sequences, ensuring a continuous and consistent 10-meter spatial resolution dataset. By interpolating spectral signatures even during periods of persistent cloudiness, it enables the reliable monitoring of vegetation phenology and productivity throughout the entire growing season.
This dataset provides essential biophysical variables such as the Fraction of Vegetation Cover (FCOVER), Leaf Area Index (LAI), and the Fraction of Absorbed Photosynthetically Active Radiation (FAPAR). These variables are critical for modelling carbon fluxes and estimating Gross Primary Production (GPP), as they provide a stable representation of the photosynthetic capacity of the canopy. Because the dataset is standardized and synchronized, it allows machine learning models to ingest a clean temporal signal, reducing the noise associated with varying atmospheric conditions and sensor differences.

Deployment & Access
The three models were trained locally, but all the files for preprocessing the input data and for the actual training are published in the Grassland RILAb Github repository at the following link: https://github.com/ScaleAGData/Grassland-RILAB/tree/main/GPP_EO_Model .
The results obtained for each model are also published in the same repository.
By uploading this repository to the RIE (or another suitable machine), it is possible to repeat the training or, if new input data is accessed, improve it.
In the same repository, in the inference section of each model, the codes for inferencing the model and thus obtaining the GPP maps (based on the best models from the training phase) have been published.
To run the scripts in this repository, a Python 3.9+ environment is required. The use of the **Conda** package manager (via Miniconda or Anaconda) is strongly recommended to ensure cross-platform consistency and reliable dependency management. Setup is facilitated by an included environment.yml file, which automates the installation of all necessary libraries, including specialized frameworks such as TensorFlow, XGBoost, and Rasterio, via the command conda env create -f environment.yml.

In terms of hardware, a standard modern workstation is sufficient for basic operations; however, a minimum of 16GB of RAM and a quad-core processor are recommended for the efficient processing of high-resolution satellite imagery and model training. While the scripts are compatible with CPU execution, a dedicated NVIDIA GPU with at least 8GB of VRAM is preferred to significantly accelerate deep learning and geospatial inference workflows.

For all the scripts published in the repository, a README.md file is published containing: the description of the script structure, the required inputs, how to obtain and format them, the outputs, and how to run the script.

Comment on Training
As we mentioned already, we have developed two models for the prediction of GPP. The training procedure described below is the same for both models; for this reason, in the figure describing the tool, we have used common blocks to indicate Sentinel 1 and Sentinel 2, but in practice only one of the two occurs, depending on whether M1 (i.e., the one receiving Sentinel 1 data) or M2 (i.e., the one receiving Sentinel 2 data) is trained.

Training
Step 1: In-situ data from Eddy Covariance flux towers and local weather stations are collected. GPP is calculated from the flux measurements and aggregated to daily values.
Step 2: Satellite data (Sentinel-1 or Sentinel-2) corresponding to the same dates are downloaded via the Sentinel Hub API.

Step 3: Features are prepared from satellite data, including selected bands from Sentinel-2 and radar backscatter from Sentinel-1. Contextual information such as topography, day of year, and area of interest is also included.
Step 4: Input features are paired with daily GPP values to form training pairs (X, Y) for supervised learning.
Step 5: Neural networks (ANNs) are trained separately for Sentinel-2 and Sentinel-1:

- Sentinel-2 model: Higher accuracy but requires cloud-free optical data.
- Sentinel-1 model: Lower accuracy but robust to clouds.

Retraining
Retraining is manual, triggered by a researcher when new in situ data becomes available or performance needs improvement. There is no automated retraining pipeline.

Inference
Step 1: Input satellite data and context features are prepared as in training.
Step 2: A conditional model selection is applied based on data availability: If cloud-free Sentinel-2 data are available, the Sentinel-2 model is used. Otherwise, the Sentinel-1 model is used.
Step 3: The selected model MMM is applied to the input features X to produce predicted GPP values M(X).
Step 4: Predictions are aligned with the timing of satellite acquisitions, typically corresponding to the Sentinel overpass.

Temporal & Spatial Alignment
Sentinel-1 and Sentinel-2 differ in spatial resolution and revisit times. Initially, only acquisitions within 36 hours were paired, with Sentinel-1 resampled to Sentinel-2 resolution.

In the current approach, separate models are used, each aligned to daily aggregated GPP values from in situ measurements.

To address data gaps during specific periods of the year and the challenges of simultaneous training with Sentinel-1 and Sentinel-2 data, a fourth model was developed using CropSAR2D data. This optimized machine learning pipeline is designed for high-precision estimation of Gross Primary Production (GPP) by integrating biophysical variables with advanced feature engineering.

The workflow begins by ingesting multispectral and biophysical inputs—such as NDVI, FCOVER, and FAPAR—and enhances the feature set with an interaction term to capture Light Use Efficiency (LUE) logic. Additionally, the system automatically parses acquisition dates to generate cyclical sine and cosine transformations of the Day of Year (DOY), enabling the models to accurately account for seasonal phenological shifts in grassland productivity.

At the core of the pipeline is a high-iteration optimization framework that independently trains Random Forest and XGBoost regressors over 1,000 cycles. To ensure model representativeness and generalizability, a stratified splitting strategy maintains a uniform distribution of GPP values across the training, validation, and test sets. By tracking the best-performing version of each algorithm based on validation metrics, the pipeline identifies the most stable architecture and subjects it to a final evaluation using advanced statistical indicators, including Relative Root Mean Squared Error (RRMSE) and Relative Absolute Error (RAE).

The execution concludes with the generation of a comprehensive inference bundle and a suite of diagnostic outputs. These results are organized into a timestamped directory containing the optimized model artifacts and a detailed JSON metadata file to ensure consistent feature alignment for future spatial predictions. Furthermore, the pipeline produces an Excel report of statistical results and high-

resolution visualizations, such as observed-versus-predicted scatter plots and aggregated feature importance charts. This automated end-to-end approach provides a robust framework for translating satellite-derived biophysical parameters into accurate, georeferenced maps of ecosystem productivity.

2.2.13.2 Status report

GPP_EO_Model is available in the project grassland's RIL repository (https://github.com/ScaleAGData/Grassland-RILAB/tree/main_/GPP_EO_Model). It contains all the models developed for the product, improved grassland GPP maps based on flux tower sensors. All the details are provided in the correspondent README file.

2.2.13.3 Biophysical integration

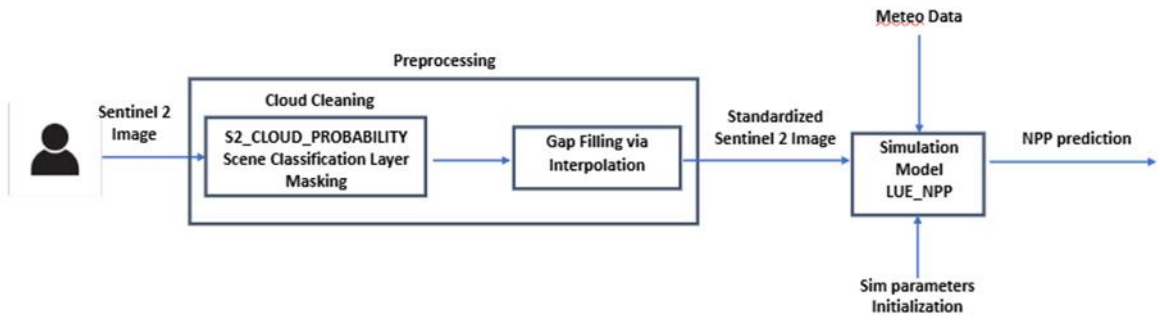


Figure 16: Biophysical integration

Monitoring Net Primary Production (NPP) is essential for understanding ecosystem carbon dynamics, but frequent cloud cover often disrupts optical satellite observations. To overcome this, the LUE_NPP model uses a simulation-based approach, combining satellite vegetation indices, environmental data, and physiological parameters to estimate daily NPP. The model reconstructs continuous time series even when raw satellite acquisitions are missing, providing a consistent view of vegetation productivity.

Deployment & Access

The model runs locally in the free and open-source programming language R on a standardised folder structure. Outputs are generated in raster files and CSV tables. No API is currently available, but the workflow is fully reproducible and could be adapted to a service if needed.

Data Pre-processing & Gap Filling

Step 1: Sentinel-2 images are filtered for clouds and shadows using the S2_CLOUD_PROBABILITY collection combined with the Scene Classification Layer (SCL) in Google Earth Engine.
Step 2: Temporal gaps caused by clouds or missing data are interpolated using the na_interpolation function from the imputeTS package, reconstructing a smooth, continuous time series pixel by pixel. A daily linear interpolation standardises the temporal frequency, ensuring the NPP time series has daily resolution.

Simulation and its Initialisation

Environmental parameters related to the simulation are set, including:

- Maximum light uses efficiency (εmax)
- Temperature and vapor pressure deficit threshold controlling vegetation response
- Vegetation type and study area characteristics

The LUE_NPP model calculates daily NPP for each pixel using meteorological data. Outputs are generated as raster and tabular formats, ready for visualisation or integration into other workflows. The mode is run on demand, while Sentinel-2 acquisitions occur roughly every 5 days. Daily interpolation ensures a continuous time series despite irregular satellite overpasses.

2.2.13.4 Status report

The grassland biomass model is fully implemented and operational. The implementation is being prepared to be uploaded to the project repository at https://github.com/ScaleAGData/Grassland-RILAB during the next months, including the code, instructions, and a test database.

2.2.14 Edge processing component

The detailed specifications for the Edge Spot can be found in I_APN_ESPT_04_2.0 – EdgeSpot Specification.pdf. In summary, the Edge Spot is a versatile and extensible IoT device capable of connecting to a wide range of sensors and actuators and transmitting data across various networks.

Its flexibility is enabled by three mikroBUS™ slots, which can accommodate over 1,000 existing extension cards, and additional custom cards can be easily designed to meet specific data collection needs. The device also features a highly adaptable power management system: it can harvest energy from solar panels and manage battery loads, or, in deployments with limited energy sources, enter an ultra-low-power mode, consuming only a few tens of microamps, while still responding to external triggers or periodically waking to take measurements.

At its core, the Edge Spot is built around an STM32L4 microcontroller, providing sufficient memory and processing power to run small AI models (TinyML) or more complex applications. As such, the Edge Spot functions as a far-edge sensing and computing platform, offering edge computing capabilities that can be tailored to different applications depending on the types of connected sensors.

2.2.15 Data transformation – AIM semantic translator

2.2.15.1 Semantic translation for the Water Productivity and Grasslands RILs

The initial phase of our work involved integrating observational data from a series of Excel (.xlsx) datasets provided by Grasslands RIL (MORR Fram Pilot, ST EURAC Farms) and Water Productivity RIL (VRI Meteorological Station, Quinoa Pilot). These datasets were ingested into our central database through a set of customised ETL (Extract, Transform, Load) pipelines.

For each data source, we developed dedicated ETL scripts tailored to the structure and semantics of the respective spreadsheets. The extraction process parsed the raw data directly from the .xlsx files, while the transformation stage handled data cleaning, normalisation, and restructuring to ensure consistency with our internal data schema. This schema was specifically designed to support both long-term storage of raw observations and standardised data exchange across systems.

During transformation, efforts were made to detect and correct inconsistencies, unify timestamp formats, validate measurement units, and map column headers to canonical field names. The final load phase systematically populated the database, preserving data lineage and source integrity for traceability.

By implementing this modular and scalable ETL framework, we ensure that the heterogeneous data contributions from different RIL pilots can be harmonised and made interoperable within the broader data infrastructure.

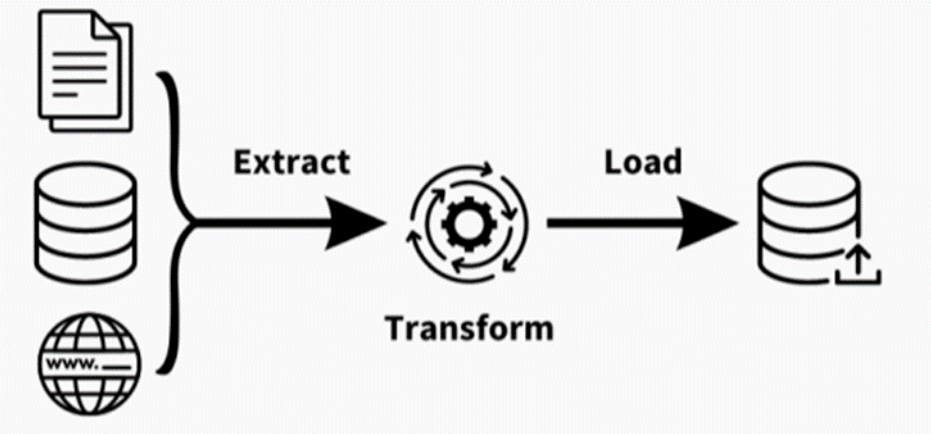


Figure 17: ETL Procedure (https://www.infobelpro.com/en/blog/etl-process)

To facilitate standardised data sharing, we designed and implemented an API structured around a core data model that captures the semantics of observational datasets. The following primary models describe the API architecture:

1. Devices: Each column in the original data sheets is abstracted as a unique device, representing a specific source of measurement. This abstraction applies uniformly, regardless of whether the data originates from a physical sensor, a human observer, or an external system. This approach allows uniform treatment of all observation sources.
2. Labels: Labels are used to categorise and group related devices based on their functional roles or deployment contexts. For example, devices associated with meteorological stations in both the VRI meteorological station and Quinoa pilot are tagged under a common label to support filtered queries and metadata enrichment.
3. DeviceLocations: This table captures the spatial context of devices, linking them to fixed or dynamic geographic coordinates. It enables spatial filtering and geospatial queries on the observational data and supports both stationary and mobile deployments.
4. ObservedPropertyTypes: This component defines the metadata associated with each type of observation, including the observed phenomenon, its expected unit of measurement, and the value type (e.g., numeric, categorical). It provides a formal specification for interpreting the semantics of the values reported by each device.
5. Observations: Each cell in the original datasets is stored as a distinct observation record. An observation is defined by its timestamp, associated device, observed property, and value. This fine-grained representation allows precise temporal and contextual indexing of the data and supports high-resolution analysis across multiple dimensions.

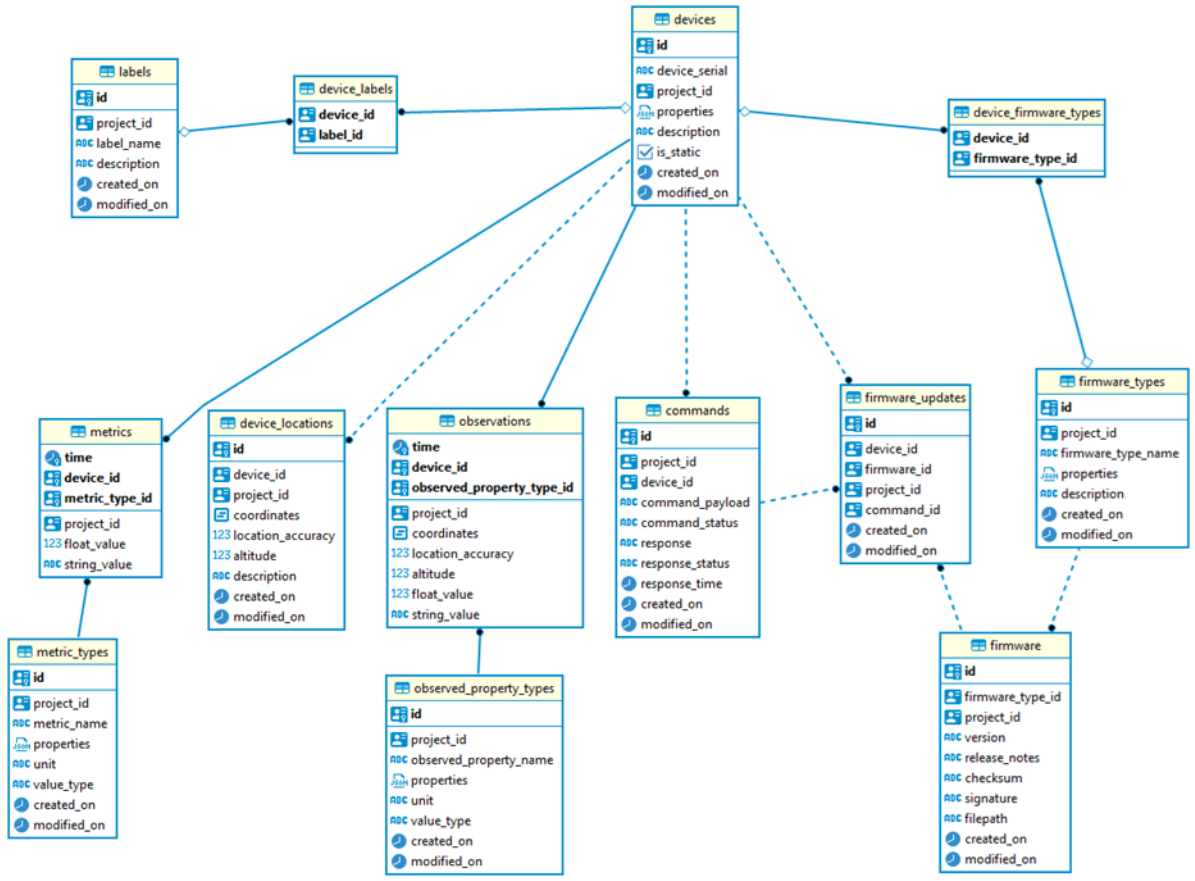


Figure 18: ScaleAgData Raw Observation API architecture

Following the development of the API and successful ingestion of the observational data, we structured and published comprehensive API documentation to support data access and integration by external users and systems.

The ScaleAgData API documentation is organised into two primary collections, aligned with the structure of the contributing RILs:

1. Grasslands RIL Collection: This section includes all relevant API endpoints associated with datasets originating from the Grasslands pilots, such as the MORR farm pilot and ST EURAC farms pilot.
2. Water Productivity RIL Collection: This section aggregates endpoints related to the Water Productivity RIL datasets, including those from the VRI Meteorological Station and Quinoa pilot.

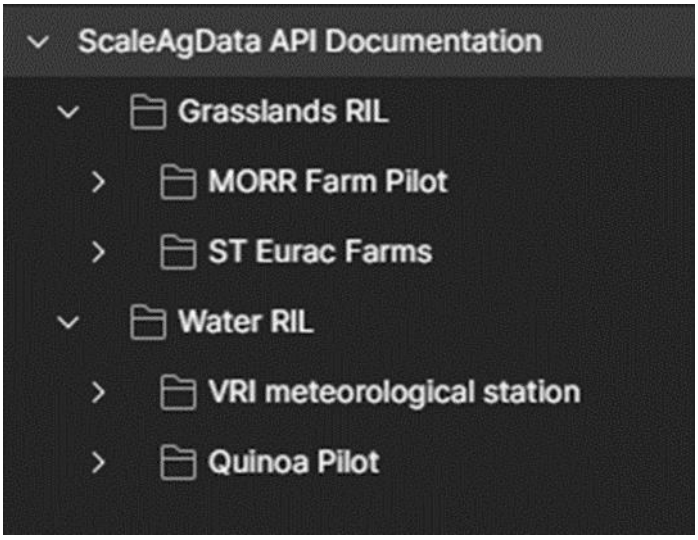


Figure 19: ScaleAgData API documentation

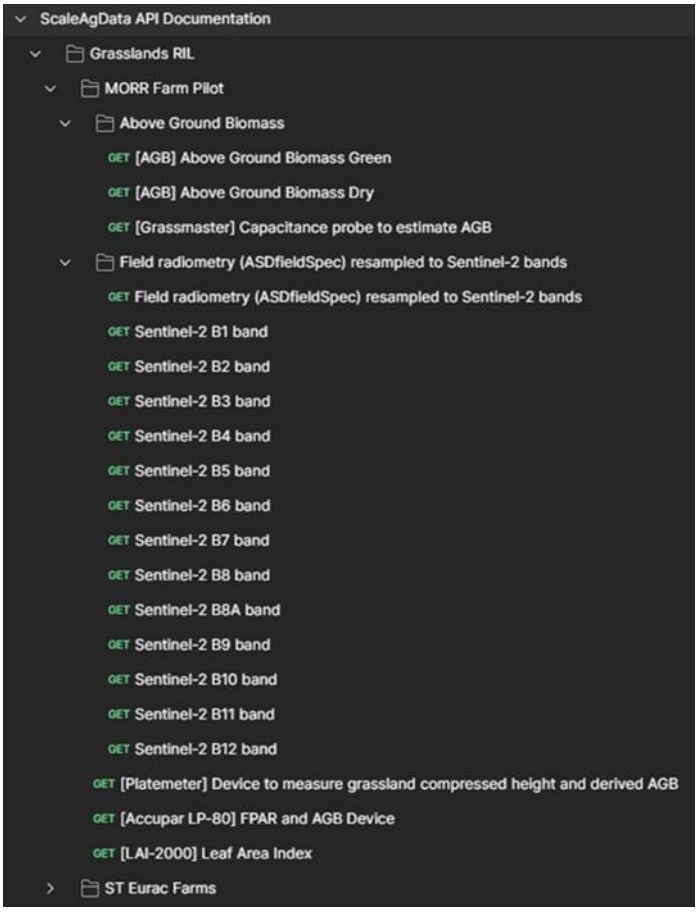


Figure 20: MORR farm pilot documentation

G84 | 190,201666666666

	B	C	D	E	F	G	H	I
82	Date	Plot	Rep	Lat	Long	AGB green fraction (kg DM/ha)	AGB dry fraction (kg DM/ha)	AGB (green+dry fractions) (kg DM/ha)
83								
84	5/6/2024	MORR_P1	11	343965,5	4247175,7	190	1785	1975
85	5/6/2024	MORR_P1	12	343965,7	4247165,7	180	1701	1881
86	5/6/2024	MORR_P1	13	343965,5	4247155,9	160	1533	1694
87	5/6/2024	MORR_P1	21	343955,5	4247175,0	170	1617	1787
88	5/6/2024	MORR_P1	22	343955,1	4247165,3	160	1533	1694
89	5/6/2024	MORR_P1	23	343955,5	4247155,6	140	1366	1506
90	5/6/2024	MORR_P1	31	343945,5	4247175,6	91	947	1037
91	5/6/2024	MORR_P1	32	343945,8	4247165,9	160	1533	1694
92	5/6/2024	MORR_P1	33	343945,6	4247155,8	150	1449	1600
93	5/6/2024	MORR_TO	11	345585,7	4247475,9	265	1411	1676
94	5/6/2024	MORR_TO	12	345585,2	4247465,8	266	1661	1927
95	5/6/2024	MORR_TO	13	345585,7	4247455,6	261	1955	2215
96	5/6/2024	MORR_TO	21	345575,2	4247475,8	266	1661	1927
97	5/6/2024	MORR_TO	22	345575,3	4247465,7	221	2707	2928
98	5/6/2024	MORR_TO	23	345575,3	4247455,9	266	1661	1927


```
{
    "time": "2024-06-05T00:00:00.507256Z",
    "float_value": 190.20166666666637,
    "string_value": null,
    "location": {
        "type": "Feature",
        "geometry": {
            "type": "Point",
            "coordinates": [
                343965.5,
                4247175.7
            ]
        },
        "properties": {
            "location_accuracy": null,
            "altitude": null
        }
    },
    "created_on": "2025-01-27T09:41:50.151783Z",
    "ObservedPropertyType": {
        "observed_property_type_name": "[AGB] Above Ground Biomass",
        "unit": " kg of dry matter(DM)/hectare (kg DM/ha)",
        "properties": {
            "qudt_unit": "qudt:KiloGM-PER-HA"
        },
        "value_type": "float"
    }
},
```


Figure 21: MORR farm pilot datasheet AGB entry and its correlated API response

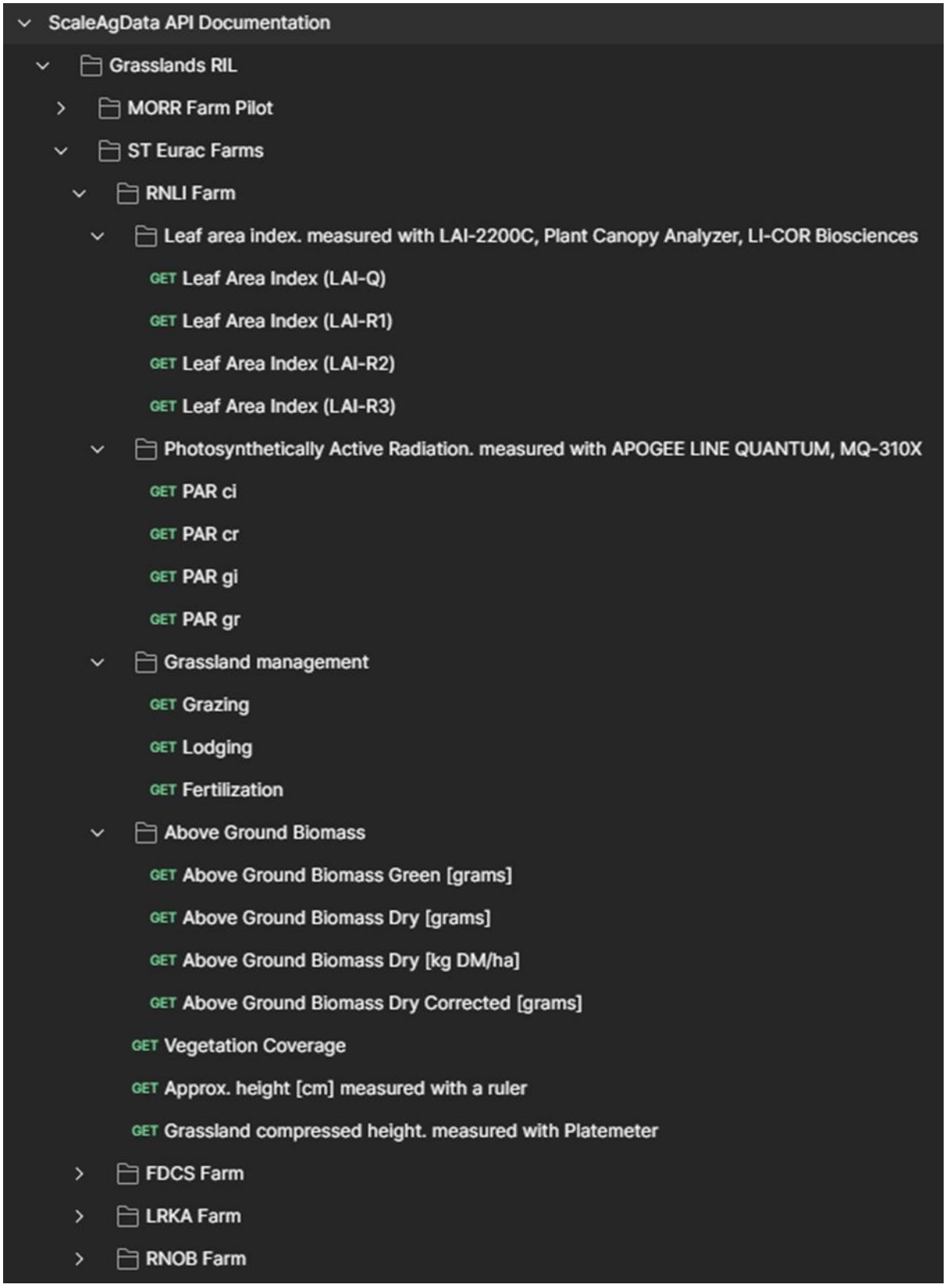
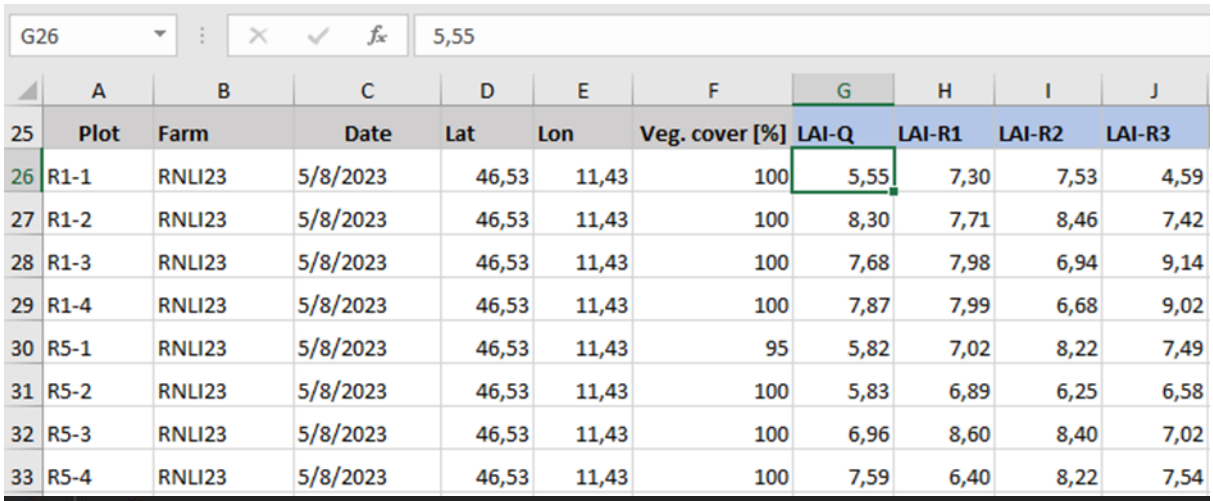


Figure 22: ST EURAC farms pilot documentation



G26 | 5,55

	A	B	C	D	E	F	G	H	I	J
25	Plot	Farm	Date	Lat	Lon	Veg. cover [%]	LAI-Q	LAI-R1	LAI-R2	LAI-R3
26	R1-1	RNLI23	5/8/2023	46,53	11,43	100	5,55	7,30	7,53	4,59
27	R1-2	RNLI23	5/8/2023	46,53	11,43	100	8,30	7,71	8,46	7,42
28	R1-3	RNLI23	5/8/2023	46,53	11,43	100	7,68	7,98	6,94	9,14
29	R1-4	RNLI23	5/8/2023	46,53	11,43	100	7,87	7,99	6,68	9,02
30	R5-1	RNLI23	5/8/2023	46,53	11,43	95	5,82	7,02	8,22	7,49
31	R5-2	RNLI23	5/8/2023	46,53	11,43	100	5,83	6,89	6,25	6,58
32	R5-3	RNLI23	5/8/2023	46,53	11,43	100	6,96	8,60	8,40	7,02
33	R5-4	RNLI23	5/8/2023	46,53	11,43	100	7,59	6,40	8,22	7,54


```
{
    "time": "2023-08-05T00:00:00.688362Z",
    "float_value": 5.55,
    "string_value": null,
    "location": {
        "type": "Feature",
        "geometry": {
            "type": "Point",
            "coordinates": [
                46.531104,
                11.432162
            ]
        },
        "properties": {
            "location_accuracy": null,
            "altitude": null
        }
    },
    "created_on": "2025-01-30T16:12:06.371646Z",
    "ObservedPropertyType": {
        "observed_property_type_name": "[LAI] Leaf Area Index (LAI-Q)",
        "unit": "Square meters of leaf area per square meter of ground area (m/m)",
        "properties": {
            "qudt_unit": "qudt:M2-PER-M2",
            "type": "LAI-Q"
        },
        "value_type": "float"
    }
},
```


Figure 23: ST EURAC farms datasheet, LAI entry and its correlated API response

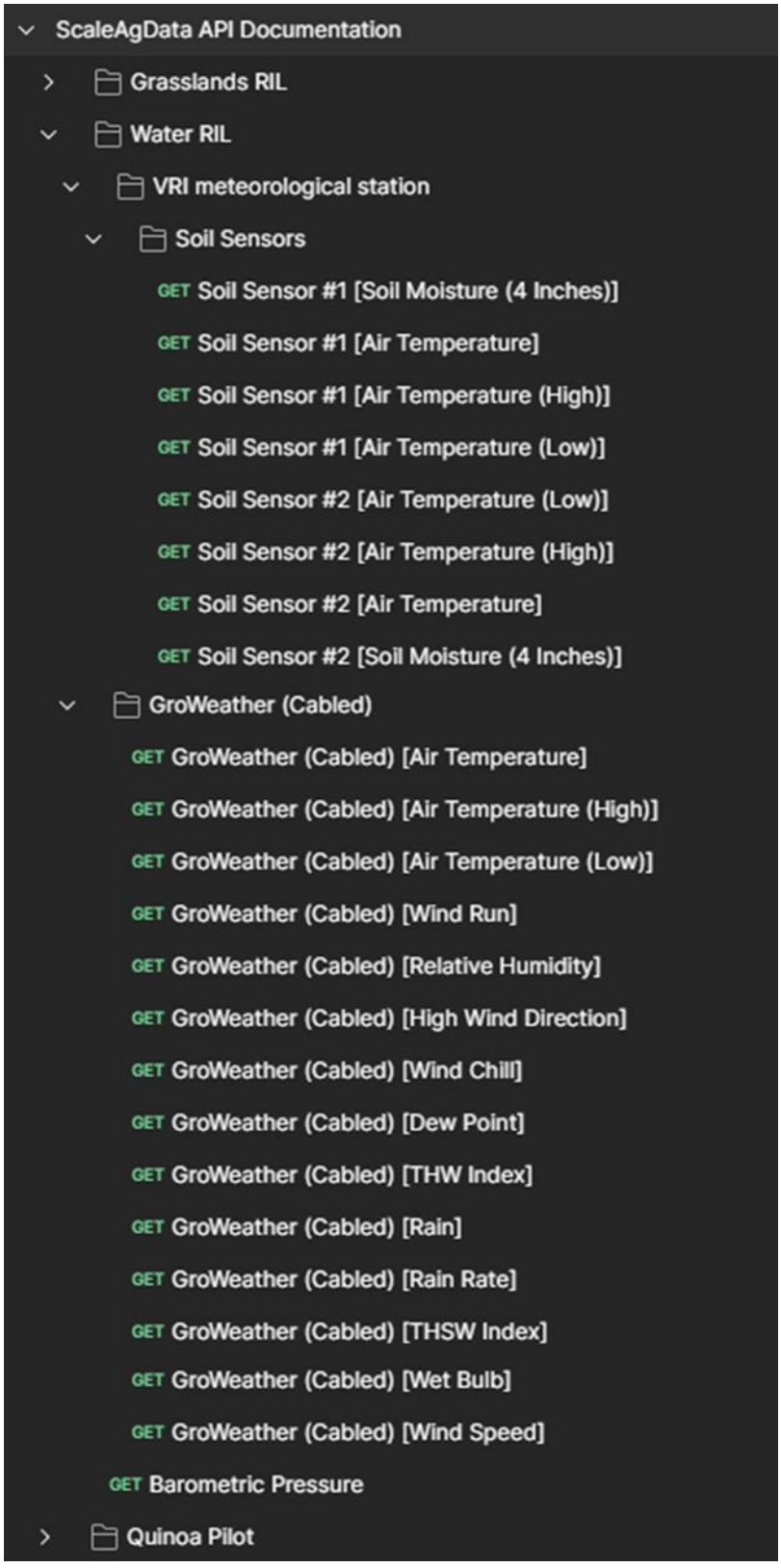


Figure 24: VRI meteorological station documentation

C5 | 17,1

	A	B	C	D	E	F	G	H
4	Date & Time	Barometer - mm Hg	Temp - °C	High Temp	Low Temp	Hum - %	Dew Poin	Wet Bulb
5	18/6/2024 00:00	762,9	17,1	17,1	17	97	16,6	16,8
6	18/6/2024 00:15	762,8	16,9	17,1	16,8	97	16,5	16,7
7	18/6/2024 00:30	762,7	16,8	16,9	16,8	97	16,4	16,6
8	18/6/2024 00:45	762,8	16,8	16,8	16,7	97	16,3	16,5
9	18/6/2024 01:00	762,7	16,6	16,7	16,5	97	16,1	16,3
10	18/6/2024 01:15	762,7	16,6	16,6	16,6	97	16,1	16,3
11	18/6/2024 01:30	762,8	16,6	16,6	16,5	97	16,1	16,3
12	18/6/2024 01:45	762,7	16,3	16,4	16,2	96	15,7	15,9
13	18/6/2024 02:00	762,8	16,1	16,2	15,9	96	15,4	15,7


```
    {
        "time": "2024-06-18T00:00:00Z",
        "float_value": 17.1,
        "string_value": null,
        "location": null,
        "created_on": "2024-11-01T13:24:25.824803Z",
        "ObservedPropertyType": {
            "observed_property_type_name": "Air Temperature",
            "unit": "Degrees Celsius(°C)",
            "properties": {
                "qudt_unit": "qudt:DEG_C"
            },
            "value_type": "float"
        }
    },
```


Figure 25: VRI meteorological station datasheet GroWeather(cabled) (Air Temperature) entry and its correlated API response

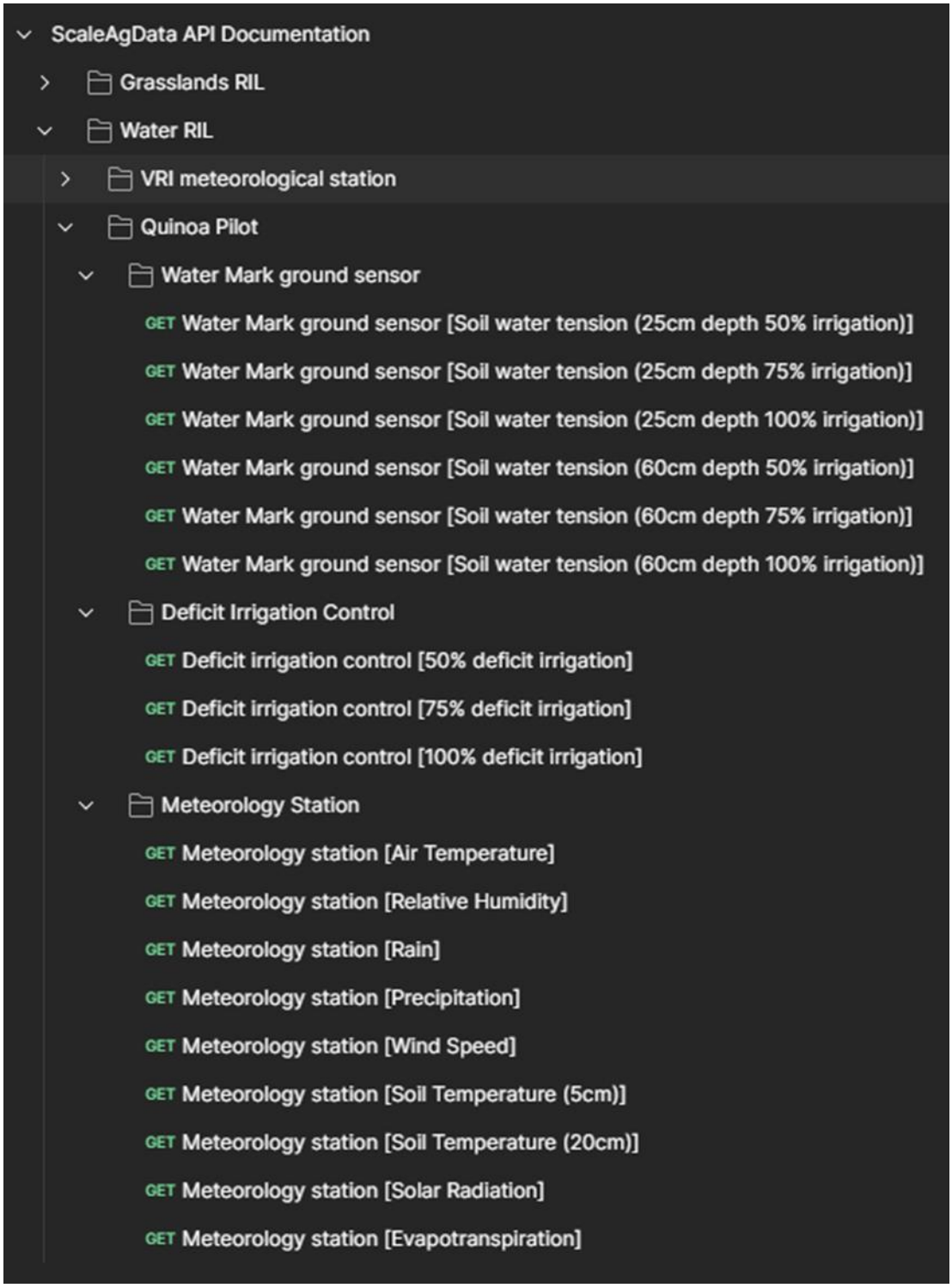


Figure 26: Quinoa pilot documentation

D4 -4,7396344

	A	B	C	D	E	F	G	H	I
3	week	Date	Time	WM_25_50%	WM_25_75%	WM_25_100%	WM_60_50%	WM_60_75%	WM_60_100%
4	36	06/09	11:45	-4,7396344	-7,5719424	-4,220975467	-12,4621552	-14,9610396	-7,851588
5			12:00	-4,80894827	-7,6149648	-4,2592176	-12,4645453	-14,9610396	-7,8802696
6			12:15	-4,88543253	-7,6866688	-4,2950696	-12,4884467	-14,9753804	-7,9161216
7			12:30	-4,95952667	-7,761958	-4,328531467	-12,5003973	-14,9789656	-7,963924267
8			12:45	-5,0479616	-7,8372472	-4,383504533	-12,512348	-14,986136	-7,990215733
9			13:00	-5,13639653	-7,9161216	-4,443257867	-12,5147381	-15,004062	-8,018897333
10			13:15	-5,2200512	-7,9985812	-4,5030112	-12,5386395	-15,0112324	-8,035628267
11			13:30	-5,306096	-8,0774556	-4,562764533	-12,5577605	-15,0148176	-8,071480267
12			13:45	-5,38736053	-8,15633	-4,627298133	-12,5768816	-15,0255732	-8,112112533


```
{
    "time": "2023-09-06T11:45:00Z",
    "float_value": -4.7396344,
    "string_value": null,
    "location": null,
    "created_on": "2024-11-01T16:22:25.924009Z",
    "ObservedPropertyType": {
        "observed_property_type_name": "Soil water tension (25cm depth /50% irrigation)",
        "unit": "Centibar (cBar)",
        "properties": {
            "qudt_unit": "qudt:CBAR",
            "depth": "25 centimeters",
            "irrigation": "50%"
        },
        "value_type": "float"
    }
},
```


Figure 27: Quinoa Pilot datasheet "Water Mark" ground sensor (soil water tension -25cm depth 50% irrigation) entry and its correlated API response

The ScaleAgData AIM API enables semantic access to agricultural observations in compliance with the AIM-Demeter Ontology. The API retrieves raw observational data from the ScaleAgData platform, transforms it into JSON-LD representations using the AIM semantic model, and exposes the results via RESTful endpoints:

- Device and location mapping: Devices (columns from the original data sheets) are fetched and semantically described, enriched with geospatial metadata if their location is fixed and provided.
- Observation transformation: Observations for a given device are fetched and reformatted into the AIM-compliant structure. Each observation includes sensor ID, timestamp, observed property type and values encoded with QUDT-compliant units.
- Time handling: Utility functions ensure timestamps are correctly parsed and formatted for semantic compatibility.

The API endpoints include:

- /devices: Returns available devices, optionally filtered by labels.
- /observedpropertytypes: Lists observed properties and associated units.
- /device/{device_id}/observations/aim-demeter: Generates a JSON-LD document containing semantically enriched device and observation data for a specified time window and property type.

Access is protected via OAuth 2.0 authentication and role-based authorisation.

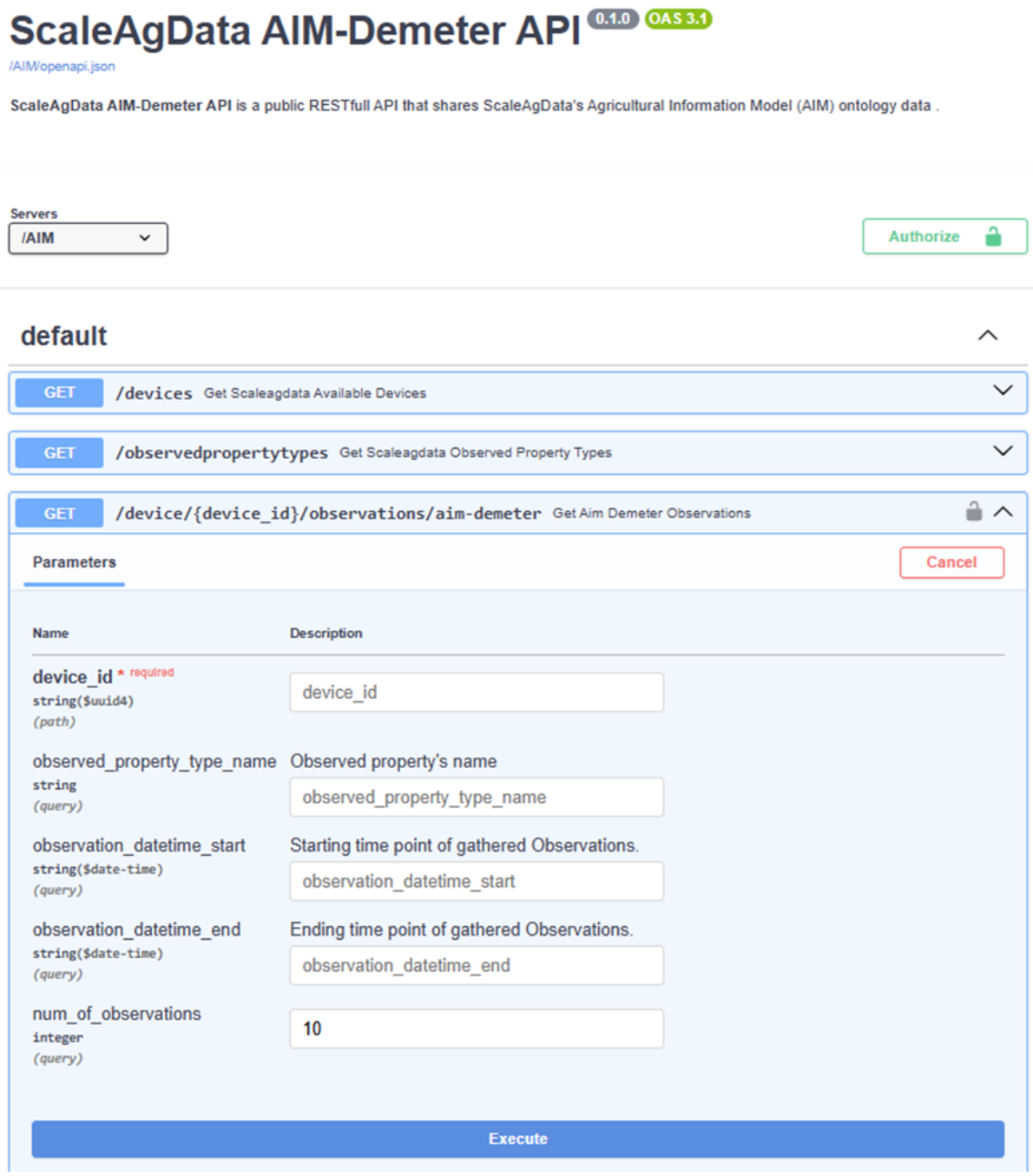


Figure 28: ScaleAgData AIM API Swagger

/device/{device_id}/observations/aim-demeter API Endpoint:
This endpoint is the core of the ScaleAgData AIM API, designed to expose observational data semantically enriched according to the AIM ontology. It retrieves device-specific measurements from the ScaleAgData platform, transforms them into JSON-LD, and structures them using standardised semantic vocabularies.

Functionality (upon request, the endpoint):

1. Fetches device metadata, including its labels and geolocation.
2. Retrieves observations associated with the specified device.
3. Transforms the data into a structured semantic format, compliant with the Demeter ontology, including:
    a. Observation

b. Quantity Value
c. Unit
d. Device with optional location

4. Packages the response within a JSON-LD schema that includes the required AIM context definitions (@context) and a graph-based representation of the data (@graph).

Request parameters (filters):
The endpoint supports several query parameters that allow the user to precisely filter the data retrieved:

Table 1: /device/{device_id}/observations/aim-demeter API Endpoint Query Parameters

Parameter	Type	Description
observed_property_type_name	Str	*(Optional)* Filters observations to a single property (e.g., Temperature, Humidity). If omitted, observations for all properties are returned.
observation_datetime_start	Datetime	*(Optional)* Defines the **start timestamp** of the observation window. Only observations collected at or after this time are included.
observation_datetime_end	Datetime	*(Optional)* Defines the **end timestamp** of the observation window. Only observations collected at or before this time are included.
num_of_observations	Int	*(Optional, default=10)* Limits the number of observation records returned for each property. Useful for reducing payload size or debugging.

These filters can be combined to efficiently query a subset of data—e.g., the latest 5 temperature readings for a specific device within a 48-hour window.

Security (access to this endpoint is restricted through:
- OAuth 2.0 authentication: Users must be authenticated and present a valid bearer token.
- Role-based authorisation: Only users with predefined roles are permitted to access the data.

This ensures the sensitive agricultural observations are shared only with authorised stakeholders.

Output (the response is a valid JSON-LD document with a structure aligned with the AIM ontology):
- @context: Semantic mapping to AIM's linked data definitions
- @graph: An array of objects describing:
  - The device and its labels
  - Associated observations
  - Quantitative results (numericValue) with semantic units
  - Geospatial information using WKT representation

AIM Output:

The Raw Data API is designed to provide direct sensor readings in a simple JSON format, making it suitable for quick data consumption in localised or internal applications. Its flat, application-specific data model offers limited interoperability due to its custom schema. In contrast, the AIM API delivers semantically enriched data using a hierarchical JSON-LD structure, grounded in the AIM and QUDT ontologies. This approach enables high interoperability. The API output conforms to multiple AIM JSON-LD contexts, embedding the data graph under an @graph node, which includes devices, observations, and quantitative values.

Following this is a comparison between the given Quinoa pilot datasheet Water Mark ground sensors (soil water tension (25 cm depth, 50% irrigation) API response), and the AIM API response.


```
    {
        "time": "2023-09-06T11:45:00Z",
        "float_value": -4.7396344,
        "string_value": null,
        "location": null,
        "created_on": "2024-11-01T16:22:25.924009Z",
        "ObservedPropertyType": {
            "observed_property_type_name": "Soil water tension (25cm depth /50% irrigation)",
            "unit": "Centibar (cBar)",
            "properties": {
                "qudt_unit": "qudt:CBAR",
                "depth": "25 centimeters",
                "irrigation": "50%"
            },
            "value_type": "float"
        }
    },
```

```
{
  "@context": [
    "https://w3id.org/demeter/agri/crossDomain-context.jsonld",
    "https://w3id.org/demeter/agri/agriFeature-context.jsonld",
    "https://w3id.org/demeter/agri/agriSystem-context.jsonld",
    "https://w3id.org/demeter/agri/agriProperty-context.jsonld",
    {
      "qudt-unit": "http://qudt.org/vocab/unit/"
    }
  ],
  "@graph": [
    {
      "@id": "urn:demeter:Water Mark ground sensor:22bc4465-26f3-432f-af0d-c800b27f4103",
      "@type": "",
      "name": "22bc4465-26f3-432f-af0d-c800b27f4103",
      "description": "Water Mark ground sensor",
      "labels": [
        "Quinoa",
        "Water Content"
      ]
    },
    {
      "@id": "urn:demeter:observation-20230906114500",
      "@type": "Observation",
      "observedProperty": [
        {
          "@id": "http://device-server-rest-api:8000/projects/0a6429e3-0d6a-477e-990b-0ddfb0b0af19/observedpropertytypes/name/Soil water tension (25cm depth /
          50% irrigation)/"
        }
      ],
      "madeBySensor": [
        "@id": "http://device-server-rest-api:8000/projects/0a6429e3-0d6a-477e-990b-0ddfb0b0af19/device/22bc4465-26f3-432f-af0d-c800b27f4103/"
      ],
      "resultTime": "2023-09-06T11:45:00.000000Z",
      "hasResult": [
        {
          "@id": "urn:demeter:Observation/20230906114500/Soil water tension (25cm depth /50% irrigation)"
        }
      ]
    },
    {
      {
        "@id": "urn:demeter:Observation/20230906114500/Soil water tension (25cm depth /50% irrigation)",
        "@type": "QuantityValue",
        "identifies": "Soil water tension (25cm depth /50% irrigation)",
        "numericValue": -4.7396344,
        "unit": {
          "@id": "qudt:CBAR",
          "description": "Centibar (cBar)"
        }
      }
    }
```


Figure 29: Quinoa pilot datasheet "Water Mark" ground sensors (soil water tension 25cm depth 50% irrigation) API response and the AIM API response

Data Structure Comparison

Table 2: ScaleAgData raw observations and AIM comparison

Attribute	Raw Data API Value	AIM API Equivalent

time	"2023-09-06T11:45:00Z"	"resultTime": "2023-09-06T11:45:00.000000Z"
float_value	-4.7396344	"numericValue": -4.7396344 inside a QuantityValue object
unit (human-readable)	"Centibar (cBar)"	"description": "Centibar (cBar)" inside unit
unit (semantic)	"qudt:CBAR"	"@id": "qudt:CBAR" (fully linked to QUDT ontology)
observed_property_type_name	"Soil water tension (25cm depth /50% irrigation)"	Included as identifier and also referenced as a dereferenceable URI
device	Implicit in context or separate call	Explicitly represented via @id, with metadata (e.g. name, description, labels)
created on	"2024-11-01T16:22:25.924009Z"	Omitted in AIM (as it focuses on observation time, not storage time)
location	null	Not shown in sample, but supported in AIM API as a has Location property

Semantic Richness:
The Raw Data API provides useful information but remains application-specific, lacking global identifiers or semantic links, with minimal ontology integration beyond basic unit tags. In contrast, the AIM API offers high semantic richness by assigning globally unique and meaningful @id values to each concept—such as sensors, observations, and units. It enriches each result with observation types, quantity values linked to QUDT units, sensor metadata, and descriptive labels. The use of cross-domain JSON-LD ensures the data is reusable and interoperable across platforms and domains.

2.2.15.2 AIM semantic translator for the Crop Management RIL

Neuropublic (NP) developed a translation pipeline to transform agricultural data retrieved from the proprietary GaiaSense API to the AIM semantic standard. We selected the AIM framework since it provides a standardised, semantic model specifically tailored for agriculture data as described in an earlier chapter for the accommodation of the overall project needs.

Dedicated translator modules were created for each category, such as irrigation, fertilisation, sprays, harvest, phenological stages, atmospheric measurements, and spatial crop data. This way, accurate mapping to the AIM vocabulary was developed to facilitate the semantic translation. Each translator script handles empty inputs, generates semantically rich objects with unique URNs, and structures observations using properties like resultTime, hasProperty, and @type, adhering strictly to the JSON-LD format (JSON-LD – JavaScript Object Notation for Linked Data – is a lightweight data format used to represent structured data using the principles of Linked Data). It is based on JSON but includes additional context to make the data machine-readable and semantically meaningful. The process is orchestrated via a central main.py, which offers users the choice of live API integration or offline data parsing, ultimately serialising the translated output to AIM-compliant files. This helps towards interoperability and semantic consistency and the integration of raw meteorological and agricultural data across various platforms.

2.2.15.2.1 Status report

The AIM semantic translation methodology is already publicly available on the project's GitHub repository. This component, which automatically converts heterogeneous farm-log data from the GaiaSense API into AIM-compliant, interoperable JSON-LD format, is available for inspection and reuse by any organisation working within the AIM or DEMETER ecosystem.

2.3 Data Exchange – NGSI-LD Data Platform

To validate data exchange principles, we implemented a data platform that supports an up-to-date NGSI-LD specification. It is intended to be a hub for the data exchanges. This data platform is depicted in Figure 30 and is comprised of several modules: an NGSI-LD context broker (STELLIO), an authenticator and access controller (KEYCLOAK), a data connector (NiFi), an administration interface (Twin-Picks), and some monitoring and dashboarding capabilities.

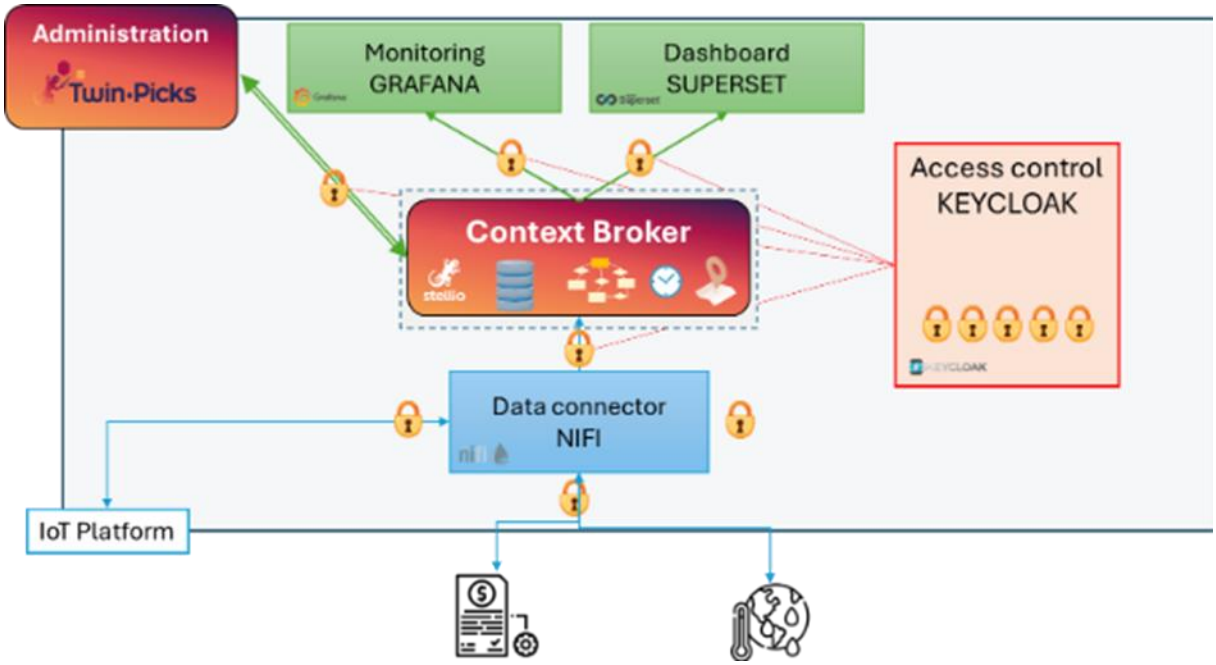


Figure 30: The NGSI-LD data platform

The Stellio context broker is an open-source implementation of the NGSI-LD specification. Adhering to the NGSI-LD standard, it provides and exposes three primary families of APIs: context information management, temporal and geospatial search, and subscription and notification management.

Beyond its compliance with the NGSI-LD standard, Stellio is fully integrated into the "FIWARE architecture". This integration is achieved through the management of structured and contextualised data according to the common NGSI-LD data model, as well as seamless interaction with other components of the FIWARE ecosystem (such as IoT agents and data collection tools). This interoperability is ensured by adherence to the NGSI-LD standard, guaranteeing both internal and external compatibility of the platform.

The principles embedded in the Stellio context manager reflect a modern, responsive, and scalable architecture:

- Business microservices are organised according to the main NGSI-LD API families, subscribing to relevant topics and utilising databases tailored to their specific needs.
- A central message bus of the pub-sub type (Kafka) facilitates the exchange of messages and events between platform components in a decoupled, extensible, scalable, and reactive

manner. The data generated by various internal components integrated into the platform (such as sensors and external data streams) are stored by the context manager in a PostgreSQL database with two specific extensions:

1. The temporal evolution of data is stored in the TimescaleDB database, an extension of PostgreSQL specialised in temporal data storage ("timeseries"). This makes it ideal for storing sensor measurements and any data that changes over time (e.g., analyses performed at specific points in time).
2. Geo-information is managed with PostGIS, another PostgreSQL extension, enabling spatial queries on the data stored in TimescaleDB, whether temporal or not.

2.3.1 Data sharing scenario between AIM and NGSI-LD

As explained in D3.1 "ScaleAgData Generic Architecture and Data Governance, Sharing Meta-Architecture and Integration of the RI Labs v1", the AIM format was selected to be used in some ScaleAgData RILs to store and exchange data. However, this AIM format, even if it is based on NGSI-LD, cannot be directly used in an up-to-date NGSI-LD broker. The reason is that the AIM format was developed with a very early version of NGSI-LD and that it took some liberties with it. Moreover, it merges other ontologies which do not map directly in NGSI-LD.

This illustrates a very common issue in the data exchange processes related to versioning of the protocols and the tiny details of their implementation. To solve this, a very common practice is to implement small software modules called "connectors" that not only get the data from one side to push them to the other; during this process, they can transform the data format.

The simple strategy for the transformation is to process the source to have its triple representation in RDF. This graph is then queried to transform each of the known types into its NGSI-LD counterpart. Figure 31 shows an AIM agriCrop object which is converted into NGSI-LD in Figure 32.


```
{
  "@id": "urn:demeter:neuropublic:harvest:agriCrop-3b7c101b-62a9-4c5b-b45e-1f4a57303392:1",
  "@type": "agriCrop",
  "resultTime": "2022-07-29",
  "hasProperty": [
    {
      "@id": "urn:demeter:neuropublic:harvest:agriCrop-3b7c101b-62a9-4c5b-b45e-1f4a57303392:1/hasHarvestDate",
      "@type": "hasHarvestDate",
      "value": "2022-07-29 "
    },
    {
      "@id": "urn:demeter:neuropublic:harvest:agriCrop-3b7c101b-62a9-4c5b-b45e-1f4a57303392:1/ProductionAmount",
      "@type": "ProductionAmount",
      "value": "3868.0"
    },
    {
      "@id": "urn:demeter:neuropublic:harvest:agriCrop-3b7c101b-62a9-4c5b-b45e-1f4a57303392:1/UnitOfMeasure",
      "@type": "UnitOfMeasure",
      "unit": "http://qudt.org/vocab/unit/KiloGM"
    }
  ]
},
```


Figure 31: AIM representation of an agriCrop object

Figure 32: Imported NGSI-LD entity

The AIM-object types currently supported by the converter are:

1. AgriInterventions
2. Plot
3. agriCrop
4. agriFeature
5. ObservationCollection
6. Observation

2.4 Research and Innovation Environment (RIE)

The RIE made available within ScaleAgData is based on the Virtual Lab solution, which is part of the Data Exploitation Platform developed by Deimos in the last years to: a) support EO service providers to build, deploy and operationalise their algorithms/applications; b) provide users with user-friendly interfaces to access those applications.

The services4EO Virtual Lab provides an interactive development environment that allows the users to develop algorithms online coded in Python and execute them over discovered data. This functionality can be accessed via Jupyter Notebooks with a subset of SDKs that helps the user during the data discovery, data access, data visualisation and data execution operations.

Once the development is finished, it is possible to deploy the new/updated script so that it is available to be triggered by the rest of the users through the ScaleAgData RIE as a standard toolbox.

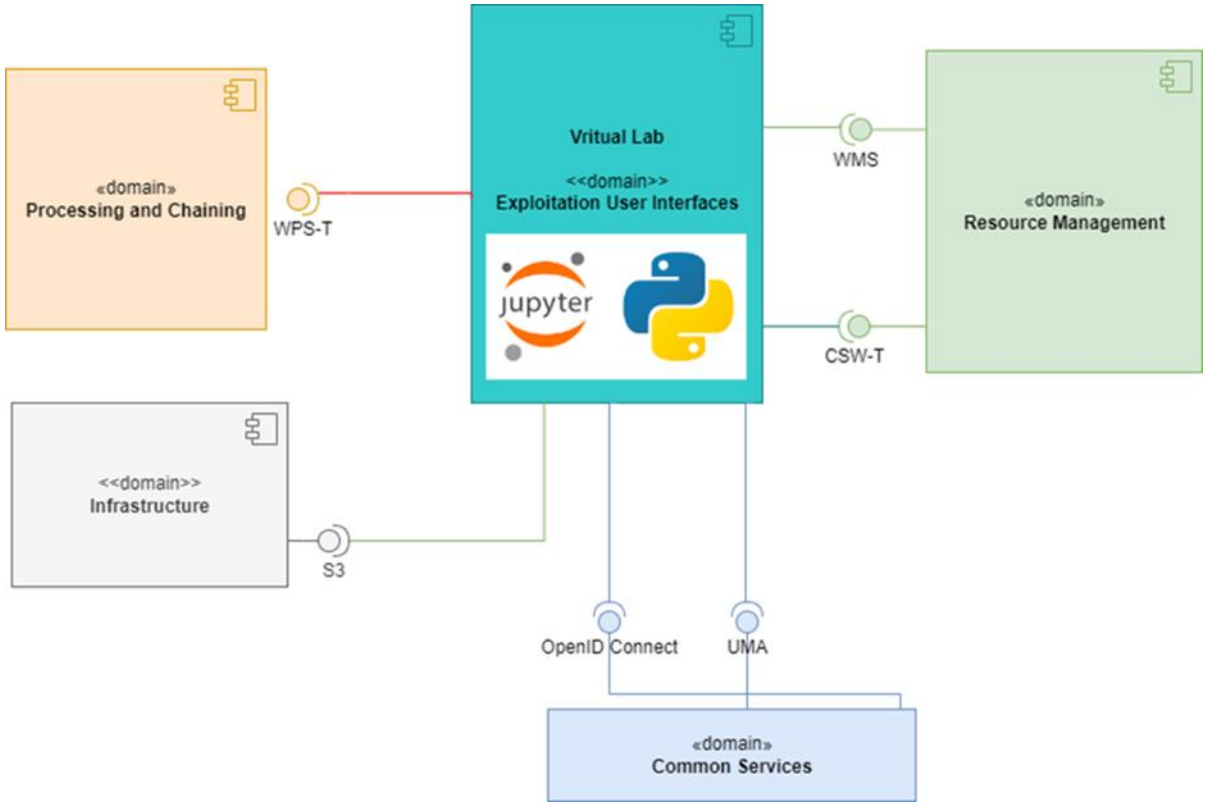


Figure 33: Research and Innovation Environment (RIE)

The Virtual Lab exposes a Jupyter Notebook interface and consumes the following main interfaces provided by the services4EO Platform.

- OpenID Connect (OIDC) and User Managed Access (UMA) for **Identity and Access Management** functionalities provided by the Common Services Domain of services4EO.

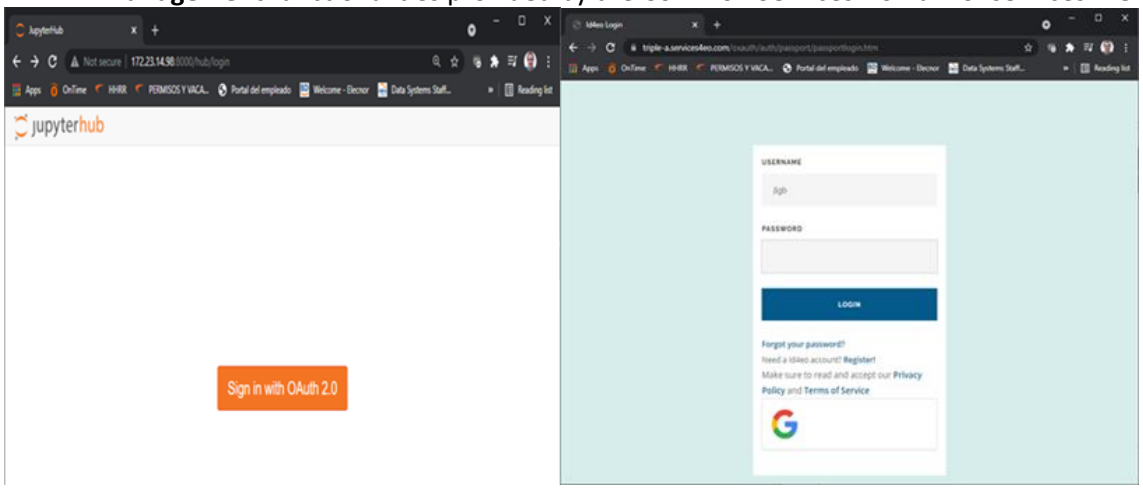


Figure 34: Identity and Access Management

- OGC Catalogue Services for the Web (CSW) for Data Discovery provided by the Resource Management Domain of services4EO.

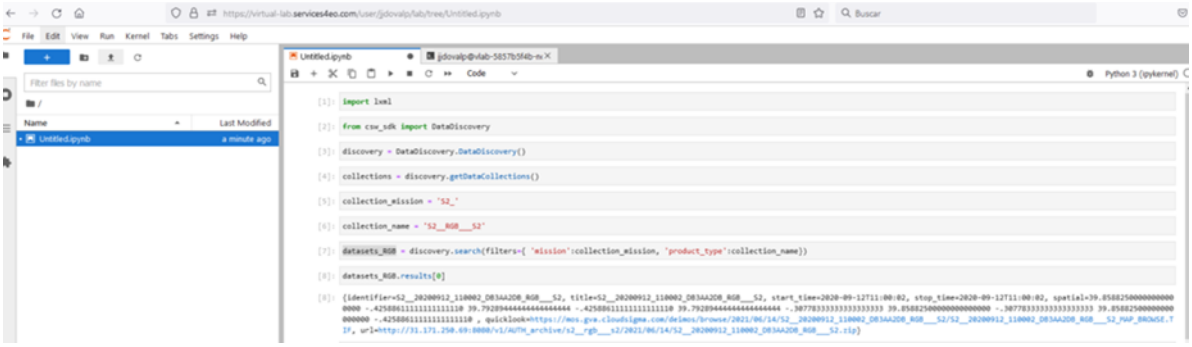


Figure 35: Data discovery

- OGC Web Map Services (WMS) for **Data Visualization** provided by the Resource Management Domain of services4EO.

```
array_list=[]
for image in images:
    img=io.imread(image)
    array_list.append(skimage.util.img_as_ubyte(img))
```

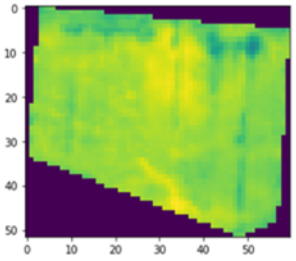
```
array_out=np.mean(array_list, axis=0)
```

```
np.mean(array_list)
```

```
60.988247863247864
```

```
plt.imshow(array_out)
```

```
<matplotlib.image.AxesImage at 0x7fc13a1718d0>
```




```
io.imsave(os.path.join(outputs_dir,"test.tif"), arr=array_out)
```

Figure 36: Data Visualization

- Simple Storage Service (S3) for **Data Access and Download is** provided by the infrastructure domain of services4EO

Download products and store them in a folder inside virtual lab

```
for product in subset:
    url = product.url
    product_download_path = os.path.join(os.path.join(download_dir, url.split("/")[-1]))
    print(product_download_path)

    # check if not already downloaded
    if not os.path.isfile(product_download_path):
        print("Downloading from Archive product " + product.title + '... ')
        urllib.request.urlretrieve(product.url, product_download_path)

    else:
        print('Product: '+ product.title + ' already downloaded in path: ' + product_download_path)

print("Download has finished")
```

```
./Downloads/zips/S2__20220430_000000_FE1D9F21_VITONDVI
Downloading from Archive product S2__20220430_000000_FE1D9F21_VITONDVI...
./Downloads/zips/S2__20220515_000000_1CBC5008_VITONDVI
Downloading from Archive product S2__20220515_000000_1CBC5008_VITONDVI...
./Downloads/zips/S2__20220510_000000_70955634_VITONDVI
Downloading from Archive product S2__20220510_000000_70955634_VITONDVI...
Download has finished
```

Figure 37: Data Access and Download

The Virtual Lab can be extended for generating an SDK that allows the users to create online an Application Package (composed of the algorithm artefact and an Application Metadata file) that can be deployed as a data processing service and executed, monitored, and controlled via OGC Web Processing Service (WPS) provided Processing and Chaining Domain of services4EO.

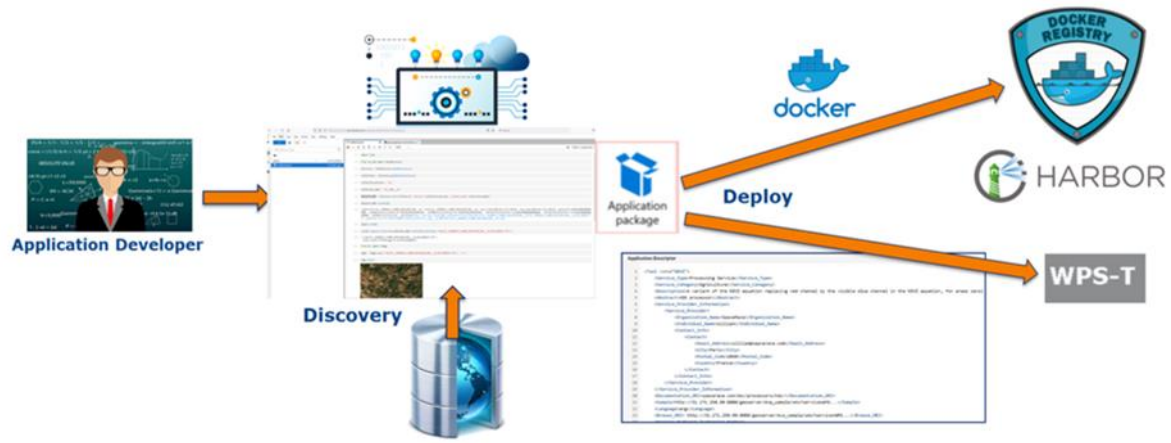


Figure 38: Data Exploitation and Marketplace interfaces

Those processing services could also be consumed from the end users via Data Exploitation or Marketplace interfaces. More information on how to use the RIE is presented in deliverable D4.1 "The Research and Innovation Environment"

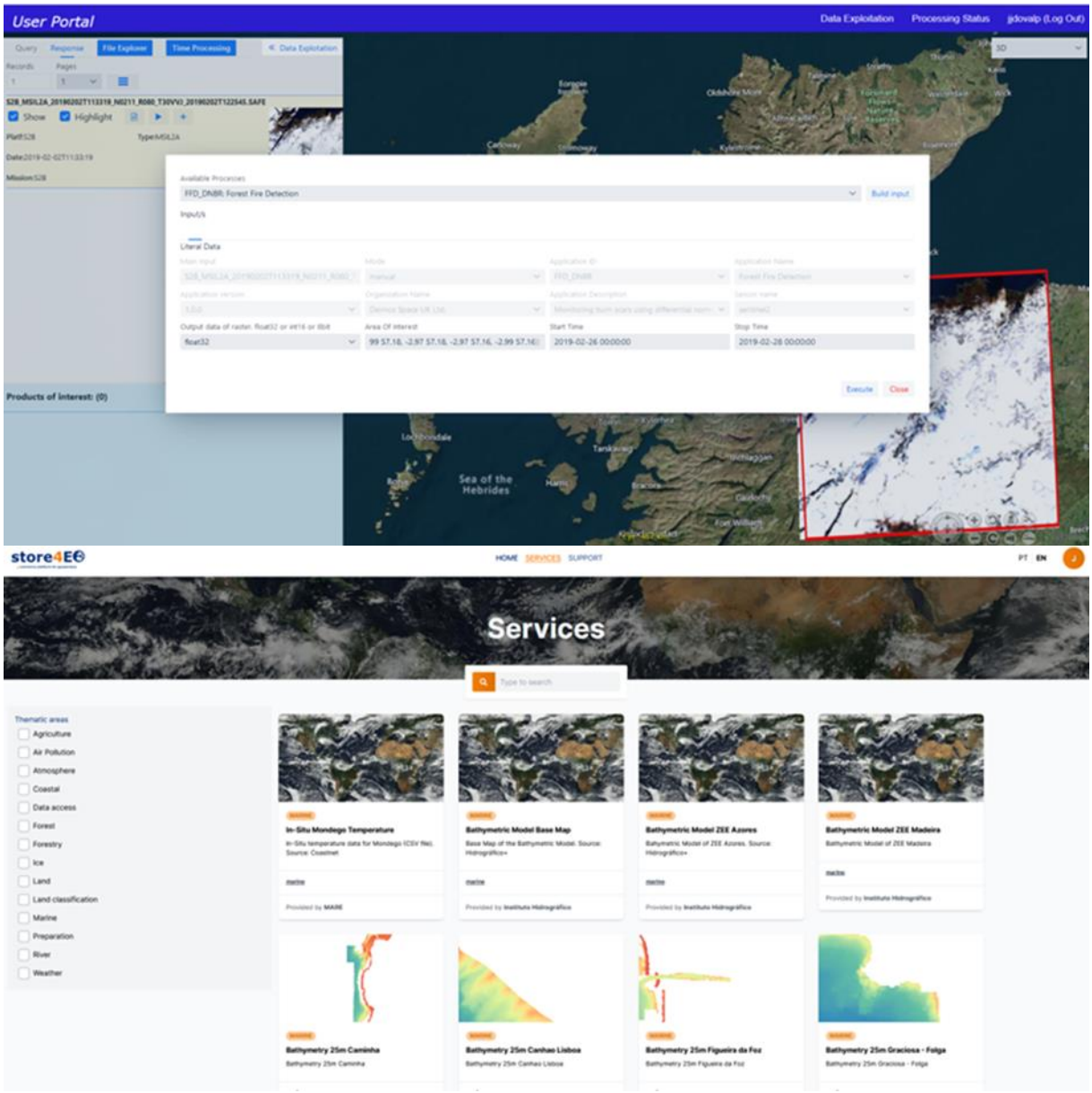


Figure 39: User portal

It is important to understand that for ScaleAgData the RI Environment is not meant to be an operational environment for external users' execution of the methodological frameworks/applications developed in the project. The main objective of the RIE is to provide an internal testing and validation environment to be used by project partners during the project. Therefore, the capabilities of application integration/deployment and consumption by users described in the previous paragraphs is not relevant for the activities of the project.

The main legacy of the RIE that will be extended beyond the project is the RIE catalogue where all project related input and output datasets will be made available for external users. Additionally, main the methodological frameworks/applications developed in the project will be available in a dedicated Github repository. The next section is devoted to the architecture of the ScaleAgData catalogue. The next section is devoted to the architecture of the ScaleAgData catalogue

2.4.1 ScaleAgData catalogue

The ScaleAgData catalogue (https://scaleagdata.nextgeoss.eu/) is based on an instance of the NextGEOSS Catalogue, developed by Deimos and its partners in the scope of a Horizon 2020 project, aimed at facilitating access, integration, and use of Earth observation (EO) data from multiple sources. Its goal is to create a federated, user-friendly, and standards-based platform where scientists, policymakers, and businesses can discover, combine, and exploit EO datasets and services for environmental monitoring, climate research, and societal applications.

It is designed as a **federated, distributed, and standards-based system**, allowing users to discover and access heterogeneous EO resources from multiple providers. Its architecture can be summarized as three main layers:

Data and Service Providers (Federated Sources)

- Various EO data providers (Copernicus, ESA, national agencies, research institutions) exposed datasets and services.
- Resources are described using **metadata standards** (ISO 19115/19139)
- Providers can include data archives, processing services, and applications.

Catalogue Core / Broker Layer

- Central CKAN based component managing from providers via standard protocols (CSW, OpenSearch).
- Indexing and normalizing metadata for unified search.
- Providing a REST Opensearch API for programmatic access. More information on the usage of that API can be found here: https://catalogue.nextgeoss.eu/opensearch
- The CKAN broker ensures interoperability between different metadata formats and standards

User Interfaces and Applications:

- **Web Portal:** Interactive web interface for searching, filtering, and previewing EO datasets and services.
- **APIs**: REST APIs for external applications and data consumers.
- **Integration with Third-Part Tools:** Supports integration into GIS software, Jupyter notebooks, and EO processing platforms.

Technical Components:

Table 3. ScaleAgData Catalogue Technical Components

Component	Purpose
Metadata Standards	ISO 19115/19139 – ensures standardized description of EO data and services.
Harvesting Mechanisms	Collect metadata from federated providers; includes support for CSW, OAI-PMH, OpenSearch.
Indexing and Storage	Metadata stored in a centralized CKAN based repository (Elasticsearch for search efficiency)
Search Engine	Enables keyword, spatial, temporal, and thematic searches. May leverage spatial indices (e.g., PostGIS) for geospatial queries.
API Layer	Provides programmatic access via REST Opensearch API; supports discovery and retrieval operations. More information on the usage of that API can be found here: https://catalogue.nextgeoss.eu/opensearch
User Interface	Web portal with map-based and list-based search, filters, metadata preview, and dataset download links.

Authentication & Authorization	Ensures secure access to restricted datasets via federated identity providers (optional).

Key Design Principles

- **Federation:** Data remains with providers, the Catalogue indexes metadata rather than storing full datasets.
- **Standards-First:** Heavy reliance on ISO and OGC standards ensures interoperability with existing EO infrastructures.
- **Scalability & Extensibility:** New data providers can be integrated with minimal effort, APIs allow integration into other platforms.
- **User-Centric Discovery:** Advanced filtering by theme, region, time, and format including map visualization for spatial context.

Underlying Technologies

While the exact tech stack is not fully documented publicly, the architecture hints at common components used in EO catalogues:

- **Backend:** Python based services.
- **Database:** CKAN/Elasticsearch databases for fast searching.
- **APIs:** Restful services, OGS CSW and openseach endpoints for metadata harvesting and discovery.
- **Frontend:** Angular frameworks for interactive maps and faceted search.

How IT Works (Flow):

- Providers expose datasets/services with metadata.
- NextGEOSS Catalogue harvests metadata using standardized protocols.
- Metadata and/or data is indexed and normalized in the catalogue/archive core.
- Users access the catalogue via web portal or API:
  - They search by keywords, regions, time ranges, or data themes.
  - They catalogue return datasest/services with links to the provider.
- Users may download data within the catalogue or access services directly from the original provider.

2.5 Data Interoperability and Data Governance

As different systems become interconnected, IoT ecosystems face significant interoperability challenges before achieving unified datasets and higher-level services. This is also true for ScaleAgData, where diverse technologies across RILs produce heterogeneous datasets in various formats.

Semantic technologies are essential for enabling interoperability, governance, device heterogeneity management, and knowledge discovery. Several IoT reference architectures (e.g., ETSI ISG-oneM2M), open-source initiatives (e.g., FIWARE), and technology providers (e.g., CISCO IoT) are developing solutions for broader use in agriculture. Yet, overcoming the "silo effect"[6] and achieving a common architecture remains difficult. Even so, current efforts are steadily advancing toward a data-driven and interoperable agricultural ecosystem.

[6] C. Brewster, I. Roussaki, N. Kalatzis, K. Doolin and K. Ellis, "IoT in Agriculture: Designing a EuropeWide Large-Scale Pilot," in IEEE Communications Magazine, vol. 55, no. 9, pp. 26-33, Sept. 2017, doi: 10.1109/MCOM.2017.1600528

ScaleAgData will tackle data interoperability challenges by reusing and adapting established best practices in semantic technologies to achieve semantic-level data harmonization. Wherever possible, standardized data models will be adopted and extended only when necessary.

The key objective is to develop data translation mechanisms that enable the harmonization of datasets generated by the RILs, making them suitable for further processing within the RIE. As shown in Figure 1, "High-level architecture of the ScaleAgData system," this will be achieved through the "Data Transformation" components, which act as translators—receiving RIL-generated data, converting it into a common model, and preparing it for subsequent analysis.

This section first provides a brief overview of data interoperability challenges in the agricultural domain, followed by a review of best practices for data harmonization, with a focus on the use of ontologies. The DEMETER Agriculture Information Model (AIM), developed under the H2020 DEMETER project, is identified as the most suitable model for reuse. Examples are then given to illustrate how primary data collected by RIL technologies can be translated using AIM semantics. The section concludes with a concise discussion of the key data governance principles guiding the use of datasets within ScaleAgData.

2.5.1 Data Interoperability

Several initiatives have sought to establish methodologies to address interoperability challenges in IoT systems. The literature identifies multiple classifications or "levels" of interoperability. For instance, the European Interoperability Framework—developed to support pan-European eGovernment services—defines three key levels: technical, semantic, and organizational interoperability.

Another IoT-oriented classification distinguishes four levels of interoperability:

- Technical Interoperability: relates to communication protocols and the supporting infrastructure.
- Syntactic Interoperability: concerns data formats and encodings, such as XML, JSON, and RDF.
- Semantic Interoperability: ensures a shared understanding of the meaning behind exchanged data.
- Organizational Interoperability: involves the capacity of organizations to exchange and interpret information effectively across diverse systems, infrastructures, or regions.

Among these, semantic interoperability remains one of the most complex yet impactful areas, with substantial progress achieved through recent EU-funded projects. There is a growing shift toward semantic standards and ontologies that explicitly define data and models, leverage Uniform Resource Identifiers (URIs), and simplify data integration.

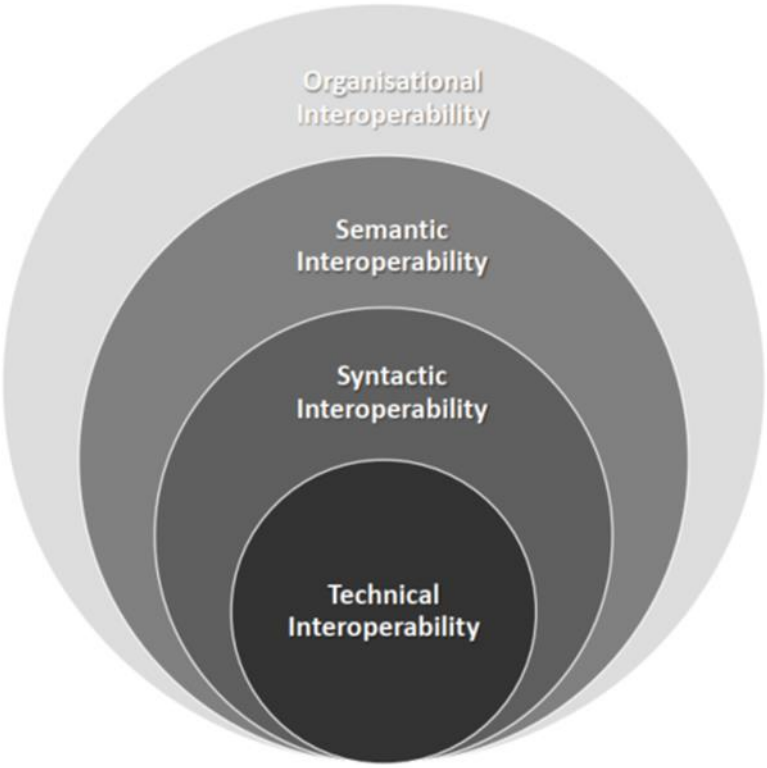


Figure 40: The dimensions of interoperability

2.5.2 DEMETER Agriculture Information Model

As described in the first version of this document (D3.1), the DEMETER Agriculture Information Model (AIM) has been identified as the most suitable approach for addressing the data modelling requirements of ScaleAgData. Our analysis indicates that AIM, being specifically designed for the agricultural sector, already incorporates most of the domain-specific concepts relevant to the project. Furthermore, its extensible structure allows for the seamless integration of any additional concepts that may be required.

Demeter AIM: The DEMETER project developed the Agricultural Information Model (AIM), building in part on the FOODIE ontology. AIM is designed primarily to support data sharing in precision agriculture and smart farming, making these its main areas of application. Unlike many other standards, AIM employs a property graph model rather than the conventional RDF/OWL approach. Its foundation is the NGSI-LD Information Model, originally developed by the telecommunications sector (ETSI) and based on the FIWARE NGSI model, which combines cross-domain concepts with domain-specific modules tailored to agriculture. Key vocabularies reused include NGSI-LD, FOODIE, and SAREF4Agri, resulting in a comprehensive and standardized set of ontologies that stands out from many other sector standardization efforts.

In addition, AIM provides extensive mappings to other relevant ontologies, including FOODON, SAREF4Agri, and AGROVOC, enhancing its interoperability. The AIM ontology is publicly available at: https://agroportal.lirmm.fr/ontologies/DEMETER-AIM/

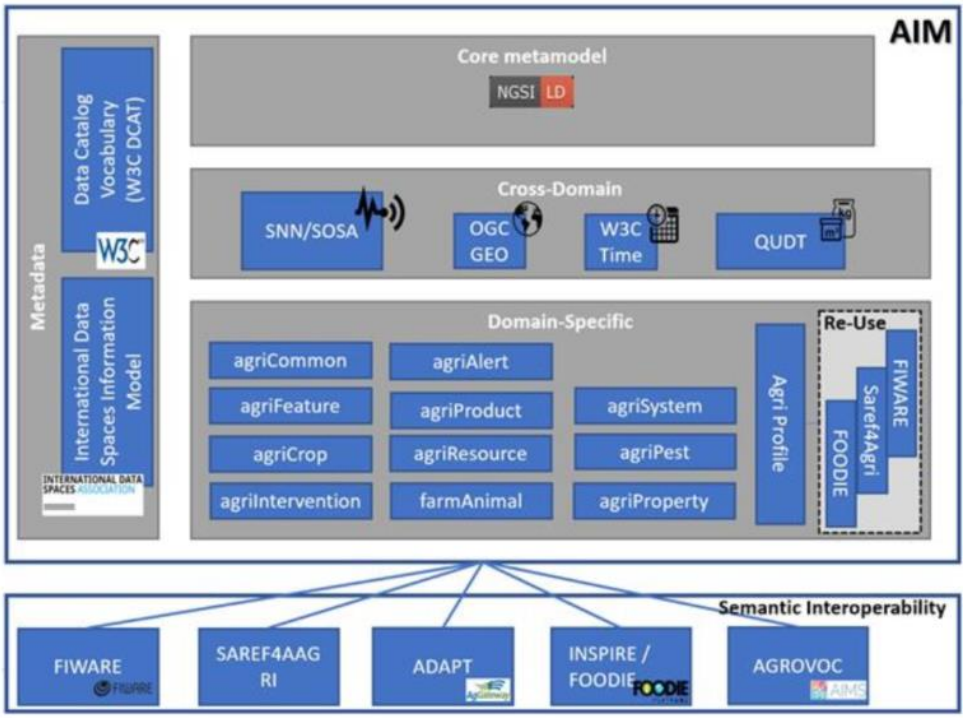


Figure 41: AIM ontology

Figure 41: Illustrates the architecture of AIM and the agricultural application domains it currently supports.

2.5.3 Data governance schemes

Among the dominant data governance principles that ScaleAgData aims to comply with are the Findability, Accessibility, Interoperability, and Reuse of digital assets (FAIR). From a practical perspective data sharing has meant the application of semantic technologies, and to a greater or lesser extent the adoption of the FAIR data principles. The use of ontologies supports such governance schemes:

- The Findability of RIL data is ensured using APIs.
- The Accessibility of data is ensured using standardized protocols integrated with access control.
- Interoperability of data is ensured by the (re)use of widely used ontologies/vocabularies that are accessible online.
- Reusability of data is ensured by using community standards and by ensuring data provenance is a consequence of data ownership and control.

The AIM has already been evaluated against these principles and proved that it is capable to address them to a large extend especially with regards to Interoperability and Reusability. Figure 43 provides the assessment questions outcomes of AIM's FAIRness (https://agroportal.lirmm.fr/ontologies/DEMETER-AIM/%20)

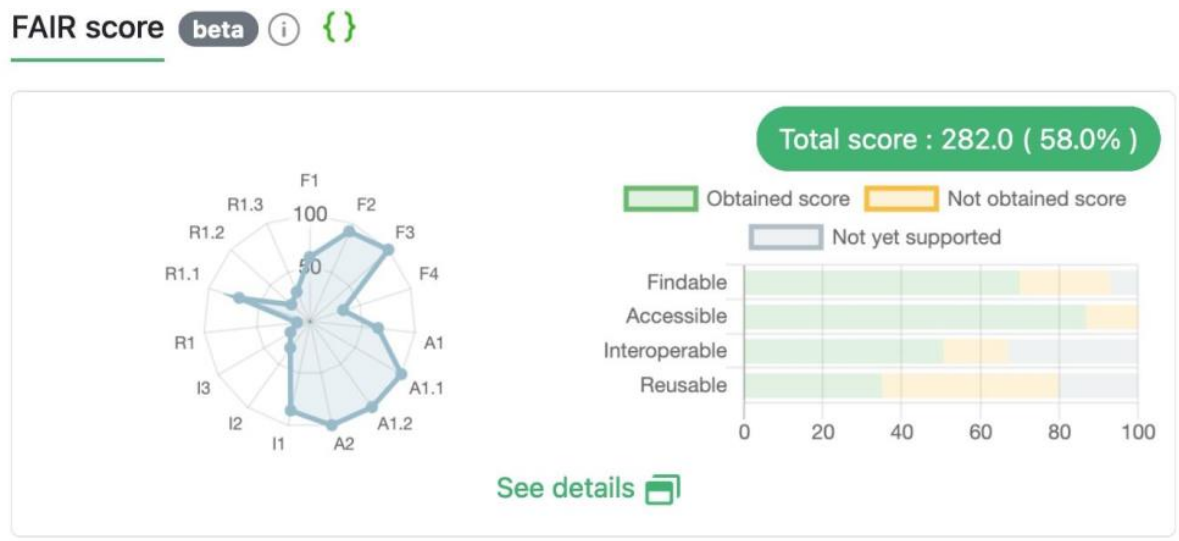


Figure 42: AIM's score against FAIR principles

Overall, the details of the data governance models are specified in collaboration with T2.4 Governance models for the vertical domains of the RILs, considering legal, operational aspects with technical specifications. In this context, T2.4 focuses on guiding and enabling RIL actors in the tailoring and adoption of a coherent governance framework within the second iteration and in the short term after the project. It builds on the DSSC Starter Kit, Glossary and Blueprint, the openDEI design principles, and the outcomes of the AgriDataSpace project to support the project objectives for effective data sharing and innovation within and cross the vertical domains of the RIL.

2.6 External Data Spaces

2.6.1 ScaleAgData Key points

ScaleAgData is engaged in developing data products and services, with a particular focus on diverse data types such as IoT data. The project explores transactions within data ecosystems, which may differ in terms of organization and governance. Its main objectives centre on ensuring fair access to and use of data, specifically aiming to enable data sharing and generate value within the extended defined by the RILs.

Here's a breakdown of the key points mentioned:

- Types of Data: ScaleAgData deals with different data types, including IoT data. This suggests a broad scope regarding the data they handle, including agricultural and rural data.
- Data Products and Services: The primary focus is on the development of data products and services. This could involve creating solutions that leverage data for specific applications, potentially in the agricultural sector.
- Transactions within Data Ecosystems: ScaleAgData is involved in transactions within data ecosystems. This could refer to activities such as data exchanges, collaborations, or partnerships within the broader context of data management.
- Organization and Governance: The data ecosystems may vary in terms of organization and governance. This implies that ScaleAgData may operate in diverse environments with different levels of structure and control over data.

- Fair Access to and Use of Data: Ensuring fair access to and use of data is a key concern. This suggests a commitment to equitable data practices, addressing issues related to data ownership, privacy, and sharing.
- Technical and Non-Technical Building Blocks: ScaleAgData considers both technical and nontechnical building blocks of data spaces. This indicates a holistic approach involving both the technological aspects of data management and the broader organizational and procedural components.
- Facilitating Data Sharing: A key goal is to facilitate data sharing. This aligns with broader industry trends where collaboration and data sharing contribute to innovation and value creation.
- Creation of Value from Data: ScaleAgData aims to enable the creation of value from data. This underscores the belief that data, when properly managed and utilized, can yield significant value, potentially benefiting stakeholders within the defined vertical of Rural Innovation Labs.

Through a series of meetings and discussions, the roles of each partner involved in task T3.4 (Data governance, sharing meta architecture and integration) were clearly defined. It was also established that implementing a full Dataspace falls outside the project's scope. Nonetheless, a roadmap was developed for implementing key building blocks using Dataspace concepts.

User roles and corresponding access rights were defined, and a map outlining data sharing needs was created. Based on deployment scenarios and workshop outcomes, various use cases were analysed, leading to the decision to implement one or more data sharing scenarios.

The first implementation of the Data Transformation building block, leveraging the AIM ontology, has been completed as described in chapter 2.3.15, along with guidance and support for adapting it to selected RILs. Additionally, the compatibility of the context broker—selected for data sharing purposes—with AIM-modelled data was evaluated (chapter 2.4.1), and the use of data connectors was demonstrated through real-world examples.

2.6.2 Common European Data Spaces

The European Strategy for Data envisions a "single European data space" — a unified market where data can flow freely while remaining secure and trustworthy. This space will bring together both personal and non-personal data, including sensitive business information, to give companies easy access to high-quality industrial data that can drive innovation, growth, and value creation. The strategy combines broad, cross-sector actions with the development of dedicated data spaces in key areas such as Agriculture and the Green Deal. Its ultimate goal is to build a safe, well-governed, and interoperable data environment equipped with advanced tools and strong data quality standards, encouraging collaboration and value generation across Europe's sectors.

In practical terms, a data space acts as the backbone that enables data to be exchanged among different participants within a data ecosystem, according to clear governance rules. As outlined by the DSSC, a data space should be designed with enough flexibility to support various use cases — a principle that is fully reflected in the ScaleAgData RILs.

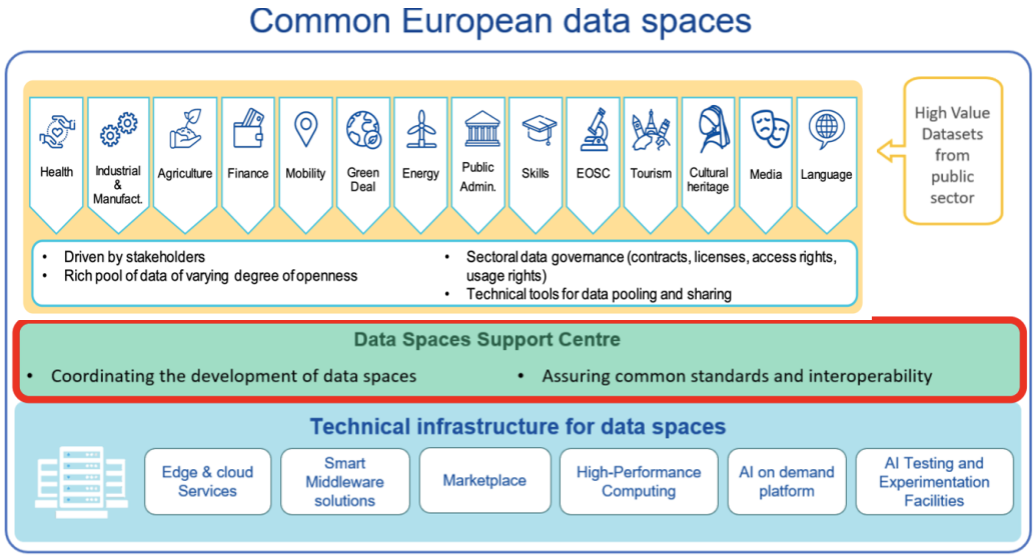


Figure 43: The Common European Data Spaces, updated version.

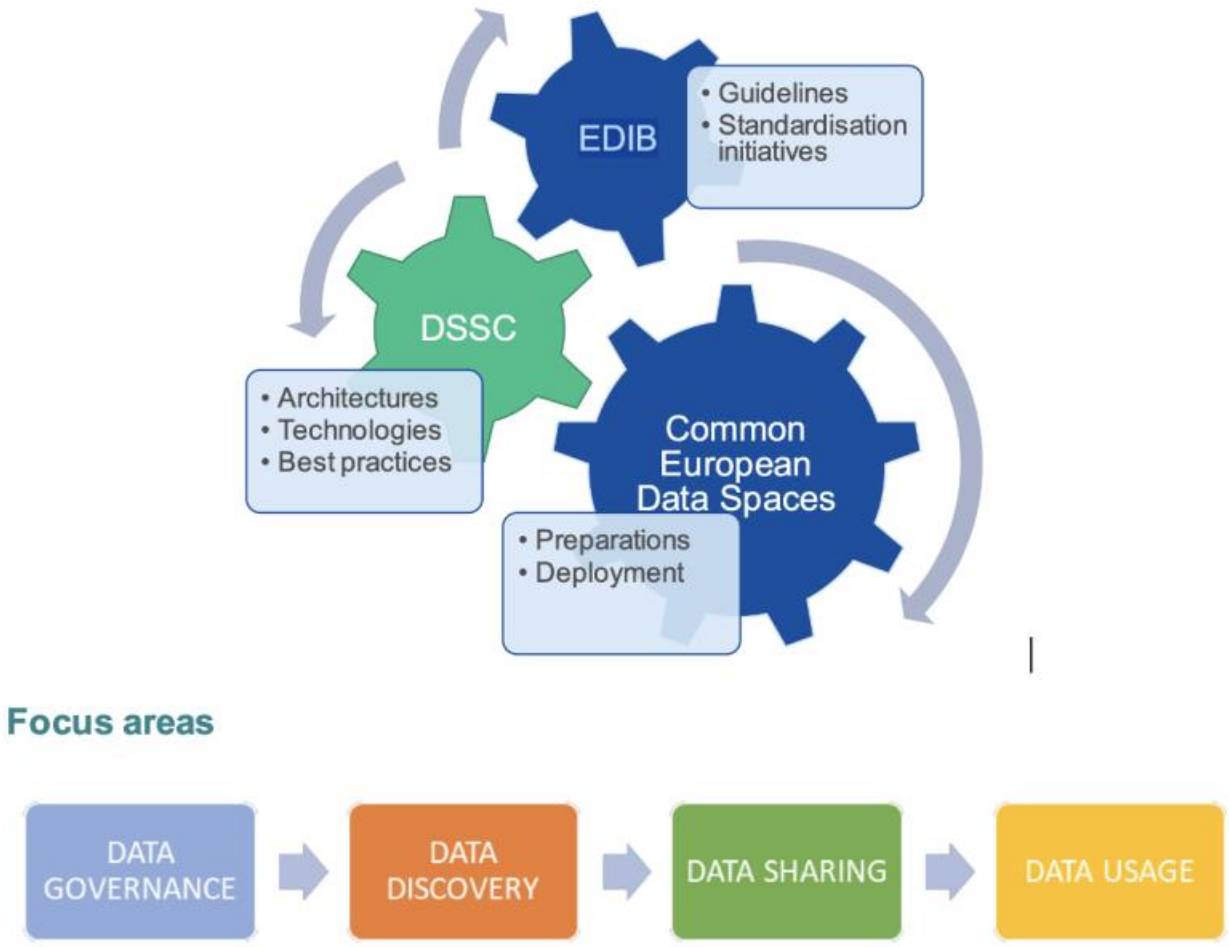


Figure 44: For developing the Common European Data Spaces, the European Data Innovation Board (EDIB), the Data Space Support Center (DSSC) and the preparation and deployment projects like AgDataSpace have complementary roles. The standardization needs to support interoperability, and data governance is a focus area.

2.6.2.1 Agricultural data space and its challenges

According to the Demeter HE project, the key challenges in establishing effective Agriculture Data Spaces are related to the following:

- **Diverse Nature of Agricultural Data**: Agricultural datasets are highly varied, covering areas such as livestock, land, climate, financial information, compliance, and food-related data. Standardized data models are essential to enable effective collection, sharing, and comparison.
- **Building a Trustworthy Environment**: Establishing trust among agri-food sector stakeholders and farmers is critical. Ensuring data sovereignty for farmers and clearly communicating the benefits of data sharing are key to fostering a reliable environment
- **Data Interoperability and Portability**: Limited interoperability and portability between tools in the agri-food ecosystem often force farmers to rely on multiple platforms. Semantic descriptions of data formats can support decentralized, interoperable data management.
- **Access to a Digital Single Market**: A unified market for technologies, services, and data is necessary for SMEs to effectively reach farmers. Agriculture Data Spaces can help reduce market fragmentation and enhance competition.
- **Sustainable Business Models**: European agri-food Data Spaces require clear and sustainable business models. Empowering data owners, incentivizing data sharing, and defining models for intermediation services are all crucial.
- **Data Quality**: High-quality data is fundamental for informed decision-making in agriculture. Data Spaces can help by supporting data quality assessment for both structured and linked data, ensuring it is fit for purpose.
- **Engaging the Entire Food Chain**: Maximizing the benefits of data sharing requires connecting farms with the broader food system. Successful implementation must consider technical, legal, ethical, socio-economic, and business aspects.

2.6.2.2 Data Space Building Blocks

To make data spaces more manageable, the DSSC introduced the concept of building blocks—fundamental units or components that can be implemented individually and combined with others to deliver the full functionality of a data space. It is important to note that these building blocks are not standalone; most of the functionalities required by data space participants emerge from the interaction of multiple building blocks.

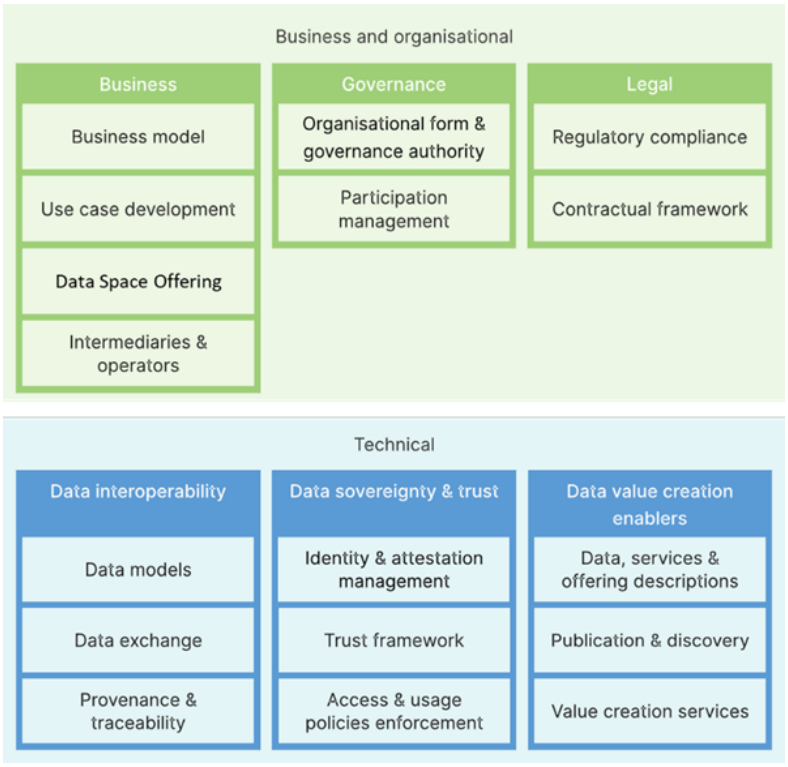


Figure 45: Building Block Taxonomy

Organizational and Business Building Blocks

Business Category: provides the essential concepts necessary in the business modelling of a data space.

Building blocks:

1. Business Model Development:
   - Supports a data space in developing its business model.
   - Identifies key elements and considerations for the governance authority.
2. Use Case Development:
   - A strategic approach to amplify the value of a data space.
   - Fosters the creation, support, and scaling of use cases.
3. Data Product Development:
   - Considers data product templates, governance rules, and network effects.
   - Focuses on enhancing synergy between data providers and users.
4. Data Space Intermediaries:
   - Supports business and governance decisions related to data space intermediaries.

Governance Category: Focuses primarily on data space-level governance. Emphasizes the dynamic nature of governance that needs to adapt as the data space evolves.

Building blocks:

1. Organizational Governance:
   - Guides setting up the data space governance authority.70
   - Identifies key decision points and options for establishing inclusive governance.
2. Participation Management :
   - Outline participant onboarding, offboarding, and role assignment, ensuring smooth and secure collaboration.
   - Aligns participation process with data governance rules on data rights, quality, and compliance.

Legal Category: Provides guidance and resources for data space initiatives to ensure compliance and establish a robust contractual framework.
Building blocks:

1. Regulatory Compliance:
   - Raises awareness of the legal landscape for data space initiatives.
   - Aids in assessing applicable regulatory requirements to ensure compliance and alignment with EU values.
2. Contractual Framework:
   - Supports a data space by establishing clear and enforceable rights and obligations.
   - Provides contractual resources for data space participants to regulate their data transactions.

The Technical Building Blocks
Data Interoperability Category: Capabilities: The exchange of data requires (semantic) models, data formats, and interfaces (APIs). Includes Functionalities for provenance and traceability.
Building blocks:

1. Data Models:
   - Capabilities: Define and use shared semantics in a data space.
   - Purpose: Enhance understanding and consistency in how data is represented and used.
2. Data Exchange:
   - Capabilities: Facilitate the actual exchange and sharing of data.
   - Purpose: Enable seamless sharing and transfer of data within the data space.
3. Provenance and Traceability:
   - Capabilities: Track the process of data sharing.
   - Purpose: Ensure traceability and compliance by making the data-sharing process transparent and auditable

Data Sovereignty and Trust Category: Capabilities: Identification of participants and assets in a data space, establishment of trust, and the ability to define/enforce access and usage control policies.
Building blocks:

1. Access and Usage Policies and Control:
   - Purpose: Enables the specification and enforcement of policies within a data space.
   - Responsibility: Handled by both the data space authority and individual participants.
2. Identity Management:
   - Purpose: Manages identities within a data space.
   - Responsibility: Ensures effective management of participant identities.
3. Trust:
   - Purpose: Verifies that a participant in a data space follows specific rules.
   - Enables: Confidence in the reliability and adherence of participants to established guidelines

Data Value Creation Category: Capabilities: Enable value creation in a data space. Examples: Registering and discovering data offerings or services, providing marketplace functionality, and enabling monetization of data sharing.
Building blocks:

1. Data, Services, and Offering Descriptions:
   - Purpose: Provides tools for data providers to describe data products comprehensively and understandably.
   - Includes: Information on data policies and ways to obtain the data product.
2. Publication and Discovery:

- Purpose: Enables data providers to publish descriptions of their data, services, and offerings.
- Follows: FAIR principles (Findable, Accessible, Interoperable, Reusable) to enhance discoverability by potential users.

3. Marketplace:
   - Purpose: Offers marketplace capabilities for providers and users to engage in relationships.
   - Enables: Access, provision, and use of data products previously published and discovered in the data space.

2.6.2.3 ScaleAgData approach towards the Common European Data Spaces

The RILs are engaged in transactions aimed at developing and scaling data products and services. By adopting the best practices of Data Spaces—covering data sharing, value creation, interoperability, trust, and digital sovereignty—RIL participants can not only carry out these transactions more efficiently but also create the conditions necessary for upscaling.

Establishing effective Agriculture Data Spaces comes with several key challenges. To address these methodologically, ScaleAgData will follow the Building Blocks Taxonomy proposed by the Data Space Support Center (DSSC), focusing on building blocks related to governance, interoperability, data sovereignty, and trust, while considering the applicable legal frameworks.

As outlined in D2.2 "Vision scenarios, Requirements and Innovative Governance Models, v2", ScaleAgData will address:

- Organizational Governance and Regulatory Compliance through Task T2.4, focusing on governance models for the RIL vertical domains.
- Data-sharing Governance primarily within Tasks T2.3/T2.4 and Task T3.4, including data governance, sharing meta-architecture, and integration. This also covers the description of data, services, and offerings, as it relates to catalogue services and metadata organization.
- Data Models and Data Exchange under Task T3.4.
- Access and Usage Policy, along with identity management, within Task T3.4 and T4.3 (Research and Innovation Environment).
- Business Model and Data Product Development in Task T6.4, covering IPR management, business model definition, and policy briefs.

In deliverable D3.1, we've provided the necessary details regarding data models, data exchange, identity management, and access/usage policies, presenting an architectural design that incorporates relevant building blocks to support the establishment and operation of a data space, particularly in terms of data sharing governance.

2.6.3 API specifications for ScaleAgData partners

Two primary approaches can be distinguished for data exchange in an agricultural environment:

1. API architectures (standalone or managed) – direct point-to-point data exchange using standardized API protocols, suitable for straightforward data sharing scenarios between trusted partners. This strategy does not fit in the data space strategy.
2. More complex data space connectors (API-Enhanced) – Sophisticated software components that manage the entire lifecycle of data exchange in a data space, including negotiation, access control, and policy enforcement. Data space connectors are comprehensive software components that rely heavily on and incorporate APIs as a core part of their functionality, using them as interfaces to the underlying data source or as the protocol for the data transfer itself. This approach is necessary when exchanging data within a governed data space ecosystem.

ScaleAgData participants should understand the fundamental differences between basic APIs and data space connectors to choose the appropriate approach for their use cases.

2.6.3.1 API architecture

API architectures provide a mechanism for direct, point-to-point data exchange between systems. When used in a standalone context, they operate without complex intermediary components. However, when used within a Data Space Connector architecture, they serve as the crucial technical interface to the data source.

Technical characteristics (of the underlying API):

- Direct HTTP/HTTPS communication between client and server
- Authentication typically handled via API keys, basic authentication, or OAUth 2.0
- Data payload defined by fixed schemas (JSON, XML, CSV).
- Synchronous request-response pattern (REST) or asynchronous messaging.
- Limited or no built-in contract negotiation capabilities.
- Access control managed at the API level (role-based or attribute-based)

When to use a standalone API:

- Trusted partner relationships with pre-established agreements.
- Internal data exchange within an organization or consortium.
- Public or open data that requires minimal access control
- Real-time data feeds where low latency is critical.
- Simple sensors data collection (IoT devices, weather stations).
- Read-only data access scenarios (public catalogues, reference data).

Advantages:

- Lower implementation complexity and cost.
- Faster development and deployment.
- Direct control over API behaviour and performance.
- Minimal infrastructure requirements.
- Easier debugging and troubleshooting.

Limitation of using APIs in a standalone manner:

- Limited built-in support for dynamic usage policies (requires connector integration).
- Manual management of bilateral agreements (requires connector integration).
- Difficult to enforce complex data sovereignty requirements (requires connector integration).
- No standardized mechanism for usage tracking and billing (requires connectors integration).
- Challenging to maintain consistency across many bilateral connections (requires connector integration).

2.6.3.2 Data Space connectors

Data space connectors are comprehensive software components that manage the entire lifecycle of data exchange in a data space ecosystem. They utilize APIs as the "Data Adapter" to communicate with backend data sources and as the standardized protocol for the Data plane transfer. They implement sophisticated protocols for negotiation policy enforcement and sovereign data exchanges.

Technical characteristics:

- Implementation of data space protocols (IDSA, GAIA-X, Simpl, Ocean protocol or Eclipse Dataspace Components).

- Separation between control plane (agreement negotiation) and data plane (actual data transfer).
- Built-in policy enforcement engines for usage control.
- Identity and trust management integrated with data space governance.
- Support from both in-band (through connector) and out-of-band data flow.
- Standardized contract negotiation and asset catalogue discovery.
- Audit logging and compliance tracking capabilities.

Architecture components:

- Core connector – Manages protocol communication, asset catalogues and policy enforcement.
- Identity provider – Handles authentication, authorization, and trust verification.
- Policy engine – Evaluates and enforces usage policies and access controls.
- Data adapters – Translate between connector protocols and backend data sources, typically by calling or exposing standard APIs (like REST) to retrieve or publish data.
- Contract management – Negotiates and stores data exchange agreements.
- Monitoring & audit – Tracks data usage and generates compliance reports.

When to use data space connectors:

- Multi-party data ecosystems with dynamic membership.
- Commercial data exchange requiring usage metering and billing.
- Scenarios with complex data sovereignty requirements.
- Cross-organizational data sharing with detailed usage policies.
- Data marketplaces with dynamic discovery and negotiation.
- Regulated industries requiring comprehensive audit trails.
- Scenarios requiring automated contract negotiation.

Advantages:

- Automated policy enforcement and compliance.
- Standardized protocols enabled ecosystem scalability.
- Built-in trust and identity management.
- Support for complex business models (pay-per-use, subscription).
- Comprehensive audit trails for regulatory compliance.
- Dynamic discovery of data assets across the data space.

Limitations:

- Higher implementation complexity and cost.
- Additional infrastructure requirements.
- Learning curve for developers.
- Performance overhead from policy evaluation.
- Requires governance framework and trust infrastructure.

Table 4: Comparison matrix between APIs and more complex data space connectors

Aspect	APIs	Data space connectors
Implementation complexity	Low to medium	High
Initial cost	Low	High
Time to deploy	Days to weeks	Weeks to months
Policy enforcement	Manual/API-level	Automated/Built-in

Contract negotiation	Bilateral/Manual	Automated/Dynamic
Trust management	External	Integrated
Scalability	Limited by bilateral connections	Ecosystem-wide
Usage tracking	Custom implementation	Built-in
Suitable scale	Few partners (<10)	Many partners (>10)
Data sovereignty	Limited control	Full control
Audit capabilities	Basic logging	Comprehensive

2.6.3.3 API standardization

API standardization is essential for achieving interoperability within ScaleAgData and across data spaces. Standardization encompasses multiple layers of the data exchange stack, from basic connectivity protocols to domain-specific semantics.

The following aspects are preferably addressed when standardizing APIs for agricultural data exchange:

1. Consistent design patterns
    - Using uniform architectural styles (REST, GraphQL, gRPC, or event-driven architectures).
    - Applying consistent conventions for endpoint naming, URL structures and resource hierarchies.
    - Establishing standard patterns for filtering, pagination, sorting and searching.
    - Defining common, approaches for handling spatial and temporal data queries.
2. Standard data formats
    - Adopting JSON or XML as primary data exchange formats.
    - Using consistent schemas for common agricultural data types (dates, timestamps, geospatial coordinates, measurement units).
    - Implementing standardized representations for agricultural entities (farms, crops, observations).
    - Ensuring alignment with semantic models (AIM, SAREF4AGRI, ADAPT).
3. Authentication and security standards
    - Implementing uniform security protocols across all APIs (OAuth 2.0, OpenID Connect, JWT tokens).
    - Defining API key management strategies for simpler use cases.
    - Establishing certificate-based authentication for machine-to-machine communication.
    - Specifying transport layer security requirements (TLS 1.2+ is mandatory).
4. Error handling and status codes
    - Using standardized HTTP status codes (200, 201, 400, 401, 403, 404, 500, etc.)
    - Implementing consistent error response structures with machine-readable error codes.
    - Providing meaningful error messages for developers and end-users.
    - Including correlation IDs for tracing errors across distributed systems.
5. API documentation standards
    - Adopting OpenAPI Specification (OAS) for RESTful APIs.
    - Providing interactive documentation through Swagger UI or similar tools.
    - Including code examples in multiple programming languages.

- Documenting rate limits, quotas, and service level agreements.

6. Versioning strategies
   - Implementing semantic versioning for API releases.
   - Supporting multiple API versions simultaneously during transition periods.
   - Providing clear deprecation notices and migration paths.
   - Using URL-based versioning (e.g., /v1/, /v2/) or header-based versioning.

Related to these points, the following can be used as a basis for the API implementation:

- REST API design principles for agricultural data
  When implementing REST APIs for agricultural data exchange, the following principles can be followed:
  1. Resource-oriented design
     - Model agricultural entities as resources (farms, fields, crops, observations)
     - Use nouns for endpoints, not verbs (/farms, not /getFarms)
     - Implement hierarchical relationships in URLs (/farms/{farmId}/fields/{fieldId})
     - Use plural nouns for collections (/observations, not /observation)
  2. HTTP method semantics
     - GET - Retrieve resources (idempotent, cacheable)
     - POST - Create new resources or trigger actions
     - PUT - Replace entire resource (idempotent)
     - PATCH - Partial update of resource
     - DELETE - Remove resource
     - HEAD - Retrieve metadata without body
     - OPTIONS - Discover supported operations
  3. Query parameters for data retrieval, e.g.:
     - Filtering: ?cropType=wheat&season=2024
     - Pagination: ?limit=50&offset=100 or ?page=3&pageSize=50
     - Sorting: ?sort=date&order=desc
     - Field selection: ?fields=name,location,area
     - Spatial queries: ?bbox=50.8,4.3,51.2,4.9 (geographic bounding box)
     - Temporal queries: ?from=2024-01-01&to=2024-12-31
  4. Response formats
     - Default to JSON with UTF-8 encoding
     - Support content negotiation via Accept header
     - Include metadata in responses (timestamp, version, pagination links)
     - Use GeoJSON for spatial data
     - Support JSON-LD for semantic enrichment

- NGSI-LD for context-aware agricultural data
  NGSI-LD is recommended for agricultural applications requiring semantic interoperability and property graphs. Key features include: entity-based model with properties, relationships, and context, temporal representation of entity evolution, geospatial query capabilities (near, within, intersects), subscription mechanism for real-time notifications, multi-tenancy support for data isolation, integration with linked data vocabularies (AIM, SAREF4AGRI).

- AsyncAPI for event-driven architectures
  - For streaming data and event-driven scenarios (sensor networks, real-time monitoring), AsyncAPI specifications should be used: define message brokers and protocols (MQTT, AMQP, Kafka, WebSockets), specify message schemas and

payloads, document publish/subscribe patterns, include security schemes for event streams, support for IoT protocols common in agriculture
- Use cases for event-driven patterns: real-time sensor data streaming (soil moisture, weather), alerts and notifications (pest detection, irrigation needs), machine-to-machine coordination (autonomous equipment), operational monitoring (equipment status, location tracking)

- Authentication and authorization standards
  It is advisable ScaleAgData APIs implement appropriate security mechanisms:
  For simple APIs:
  - OAuth 2.0 / OpenID Connect - Standard for delegated authorization
  - API Keys - Suitable for server-to-server communication
  - Mutual TLS - Certificate-based authentication for high-security scenarios
  - JWT Tokens - Stateless authentication with embedded claims

  For Data Space connectors:
  - Decentralized Identity (DID) - Self-sovereign identity for participants
  - Verifiable Credentials - Attestations about participant capabilities
  - Identity Providers - Trusted third-party authentication (Keycloak, A-AD)
  - Dynamic Client Registration - Automated connector onboarding

  Authorization patterns:
  - Role-Based Access Control (RBAC) - Permissions based on user roles
  - Attribute-Based Access Control (ABAC) - Fine-grained policies based on attributes
  - Usage Policies - Rules governing data usage after transfer (IDSA Usage Control)

- API documentation and developer experience
  Comprehensive API documentation is essential for adoption and can contain the following elements:
  - Relevant documentation elements: *Getting Started Guide* - Onboarding path for new developers, *Authentication Setup* - Step-by-step credential configuration, *Endpoint Reference* - Complete API operation documentation, *Data Model Schemas* - Detailed schema definitions with examples, *Code Examples* - Working samples in multiple languages (Python, Java, JavaScript), *Error Reference* - All possible error codes with resolution guidance, *Rate Limits & Quotas* - Usage restrictions and throttling policies, *Changelog* - Version history and migration guides, *SDKs and client Libraries* - Ready-to-use integration packages
  - Interactive documentation: *Swagger UI* - Browser-based API testing, *Postman collections* - Importable request collections, *GraphQL Playground* - Interactive GraphQL schema exploration, *API Sandbox* - Test environment with sample data
  - Developer portal features: self-service API key generation, usage analytics and dashboards, support ticket system, community forums, notification of API changes

- API Lifecycle Management
  Managing APIs throughout their lifecycle ensures long-term success and generally follows the successive phases:
  - Design phase: stakeholder requirements gathering, API contract definition (OpenAPI, AsyncAPI), security and privacy review, performance requirements specification
  - Development phase: contract-first development approach, automated code generation from specifications, unit and integration testing, security testing.
  - Deployment phase: API gateway configuration, rate limiting and throttling setup, monitoring and observability instrumentation, documentation publication

- Operations phase: performance monitoring (latency, throughput, error rates), security monitoring (authentication failures, abuse patterns), usage analytics (popular endpoints, consumer patterns), incident response procedures
- Evolution phase: versioning strategy implementation, backward compatibility maintenance, deprecation notices (e.g. minimum 6 months), migration support for consumers
- Retirement phase: formal deprecation announcement, alternative API recommendations, grace period for migration, data archival and clean-up

- Interoperability Testing and Validation
  To ensure APIs conform to standards the following actions can be followed:
  - Conformance testing: OpenAPI specification validation, schema compliance checking, HTTP protocol compliance (RFC 7231), security standards verification (TLS, OAuth)
  - Interoperability testing: cross-vendor compatibility tests, round-trip data integrity validation, semantic compatibility verification, performance benchmarking
  - Continuous validation: automated regression testing, contract testing between providers and consumers, semantic model validation against vocabularies, breaking change detection

API standardization is essential for achieving interoperability within ScaleAgData and across data spaces. Standardization encompasses multiple layers of the data exchange stack, from basic connectivity protocols to domain-specific semantics. Crucially, these standards also form the technical base that allows Data Space Connectors to automatically govern, enforce policy, and audit data transactions.

The following European initiatives on data exchange can be a source of inspiration for setting up agricultural data exchange solutions:

1. Pontus-X, Ocean Protocol and DeltaDAO
   Pontus-X represents a European data ecosystem initiative that leverages Ocean Protocol technology to enable secure and sovereign data sharing. Ocean Protocol is a decentralized data exchange protocol that facilitates the creation of data marketplaces while maintaining data sovereignty and control.
   Key characteristics:
   - Provides a protocol and network infrastructure for building data marketplaces and exchanges
   - Enables token-based data economy with support for data assets as tradeable tokens
   - Supports decentralized data sharing without requiring data to leave the provider's infrastructure
   - Implements compute-to-data approaches where algorithms travel to data rather than data moving to algorithms
   - Facilitates discovery and access to data through decentralized metadata catalogues
   - DeltaDAO contributes to the Gaia-X and European data space ecosystem development

   ScaleAgData relevance: Ocean Protocol can be considered as an alternative data space connector implementation for scenarios requiring:
   - Tokenized data assets and blockchain-based transactions
   - Decentralized marketplace functionality
   - Strong emphasis on data monetization models
   - Integration with Web3 infrastructure

2. SIMPL middleware (EC-sponsored)

SIMPL-Open (formerly SIMPL) is an EC-sponsored, open-source, multi-vendor, modular middleware for building European data spaces. It addresses the need for federated cloud infrastructure across the European Union.

Core capabilities:

- Multi-vendor, large-scale, modular and interoperable middleware architecture
- Federation of data, applications, and infrastructure across cloud-to-edge environments
- Support for both public sector and business stakeholders
- Marketplace functionality for EU resource sharing
- Integration with Gaia-X compliance and European data space standards

Key components:

- Identity and access management (IAM) including Keycloak, authentication providers, and tier-based gateway architecture
- EDC (Eclipse Dataspace Components) connector adapter for data space connectivity
- Catalogue services with FC-Service (federated catalogue) and query mapper
- Contract management and billing services
- Self-description creation wizard and validation tools
- Monitoring stack with Elastic, Kibana, and Filebeat

Actor types supported:

- Governance authority
- Data providers
- Infrastructure providers
- Application providers
- Consumers

ScaleAgData can leverage SIMPL middleware for:

- Establishing governance authority infrastructure
- Implementing federated catalogue services
- Managing participant identities and access control
- Ensuring compliance with European data space standards

3. Eclipse Dataspace Components (EDC)

EDC framework provides a set of open-source components for building global and scalable standards-based data sharing services based on the dataspace concept.

Core architecture:

- Modular extension system allowing customization without modifying core code.
- Service Provider Interface (SPI) layer defining all the necessary interfaces
- Core module containing essential components (TransferProcessManager, DataFlowManager, policy engine)
- Extensions for technology-specific implementations (cloud providers, databases, protocols)
- Launchers as run-able connector packages

Key capabilities

- Identity service for organizational credential management using DIDs and W3C verifiable credentials.
- Catalogue service for publishing and securing shareable assets.
- Control plane for automated data usage agreement creating and processing.
- Data plane services for transfer using HTTP, Kafka, cloud object storage, or custom protocols

- Standard-based implementation of Data Space Protocol Specification and Decentralized Claims Protocol

Critical EDC characteristics

- Not a pre-packaged system but a toolbox for building customized distributions.
- Does not ship with installable distribution – downstream projects customized for specific needs
- Does not contain use-case specific features – added through modularity and extension
- Does not provide data storage/processing infrastructure – integrates with third-party data planes

ScaleAgData RILs can utilize EDC for:

- Implementing sovereign data connectors with policy enforcement
- Establishing control plane for contract negotiation
- Managing data transfer protocols across heterogeneous systems
- Ensuring compliance with dataspace standards and protocols

4. DjustConnect - API requirements (provider specifications)

DjustConnect is a Belgian agricultural data platform that enables secure data sharing between agricultural data providers and consumers. For ScaleAgData participants acting as data providers to DjustConnect, specific API requirements must be met.

Provider API requirements:

- Only HTTP GET requests are currently supported
- Authentication methods:
  - Mutual SSL (client certificate validation) - preferred method
  - OAuth2 authentication as alternative
  - Optional: additional HTTP header parameters for extra security
- Server SSL certificate requirement (Let'sEncrypt or commercial certificate)
- Validation of incoming DjustConnect SSL client certificate

Documentation requirements:

- OpenAPI spec (Swagger file) must be provided
- Farm-id endpoint must be implemented as additional endpoint
- Each endpoint must include farm identification in request parameter or response body
- Documentation published to DjustConnect developer portal

Optional partner API:

- DjustConnect-Subscription-Key header for authenticated calls
- SSL certificate configuration
- Access to partner details and configuration

Push notification mechanism (events):

- Azure EventGrid-based push notifications
- JSON message format with metadata
- Event type registration and endpoint configuration
- Subject-based routing (e.g., farm numbers)

ScaleAgData considerations:

- DjustConnect represents a domain-specific, existing platform that RIL participants may need to integrate with
- Provides example of pragmatic API-based data sharing without full data space connector complexity
- Demonstrates importance of clear API documentation and authentication requirements

- Can serve as reference for defining API requirements for simple data sharing scenarios

5. VSDS, LDES and Athumi

The Flemish Smart Data Space (VSDS) project, led by Athumi (the Flemish data utility company), provides infrastructure for publishing and consuming linked data event streams (LDES).

LDES (linked data event stream) core concepts:

- Immutable, append-only collection of versioned objects
- Enables efficient synchronization and replication of datasets
- Supports fragmentation strategies for scalable data distribution
- Time-based ordering of data members

LDES server capabilities:

- Configurable component for ingesting, storing, and (re-)publishing event streams
- Retention policies: time-based, point-in-time, and version-based
- Fragmentation strategies:
  - Partitioning (pagination) - linear fragmentation based on arrival order
  - Geospatial fragmentation - based on geographic location
  - Time-based fragmentation - temporal organization
  - Substring fragmentation - based on property values
- DCAT metadata support for dataset and view descriptions
- Integration with identity and access management

Publishing pipeline architecture:

- Adapter component: transforms data into linked data
- Transformer component: converts geometry to WKT format
- HTTP-out component: transmits to LDES server
- LDES server operates as separate VSDS building block

Integration with FIWARE:

- FIWARE-Orion context broker (OCB) can publish to LDES
- VSDS NIFI solution translates NGSI-LD context data into LDES events
- Pipeline includes: HTTP listener → update attributes → OSLO converter → LdesConverter → HTTP output
- Supports real-time IoT sensor data streaming to linked data format

Integration options:

- Apache Kafka to LDES (for streaming data topics)
- MQTT to LDES (for IoT sensor networks)
- FIWARE OCB to LDES (for context-aware applications)

ScaleAgData application scenarios:

- Publishing agricultural sensor data as event streams
- Synchronizing parcel and farm data across RILs
- Enabling historical data access with temporal versioning
- Distributing geospatially-organized agricultural observations
- Providing public catalogues and reference data through LDES views

6. IDSA rulebook and ISO standardization

The International Data Spaces Association (IDSA) develops the IDSA rulebook and related standards that are being formalized through ISO standardization processes.

IDSA reference architecture model provides:

- Multi-layer architecture framework:
  - Business layer: roles, business processes, and interactions
  - Functional layer: features and capabilities

- Process layer: workflow and orchestration
- Information layer: data models and vocabularies
- System layer: technical components and their interactions

- Separation of control plane (agreement negotiation) and data plane (data transfer)
- Identity and trust management framework
- Usage control and policy enforcement mechanisms

IDSA rulebook components:

- Organizational governance requirements
- Technical specifications for connectors
- Certification criteria for components and participants
- Security and trust requirements
- Compliance and operational rules

ISO standardization efforts:

- Work towards ISO normalization of data space concepts
- Alignment with international standards for interoperability
- Retention policy standards (ISO 8601 duration format for time-based policies)
- Data governance frameworks

Relevance for ScaleAgData:

- Provides structured approach to data space architecture across all layers
- Ensures alignment with emerging international standards
- IDSA RAM system layer guides technical component selection (connectors, identity management, catalogues)
- Rulebook provides governance framework
- Certification paths for establishing trust among RIL participants

7. FIWARE connector and context broker

FIWARE provides open-source components for building smart applications, with the Orion context broker being the core component for context information management.

FIWARE OCB capabilities:

- Real-time context information management
- Receives updates from IoT devices, sensors, and other sources
- Stores context data in centralized or federated manner
- Publish/subscribe notification mechanisms
- RESTful API for context data access
- Native NGSI-LD support

Integration with data spaces:

- OCB can function as data provider for VSDS LDES streams
- Pipeline: OCB → update attributes → OSLO converter → LdesConverter → LDES server.
- Translates real-time context updates into immutable event streams.
- Enables integration of IoT sensor networks with linked data infrastructure.

Example use case: internet of water (VMM) data:

- NIFI or Orion components subscribe to IoT data sources
- MQTT protocol receives sensor measurements
- JdesConverter processes into OSLO format
- HTTP protocol publishes to LDES
- Results in time-series of water quality observations, device data, zones

Agricultural applications:

- Weather station data aggregation
- Soil sensor network integration
- Precision agriculture equipment telemetry

- Greenhouse climate monitoring
- Water management system integration

ScaleAgData integration paths:

- Use FIWARE OCB for real-time agricultural sensor data management
- Connect precision agriculture equipment through NGSI-LD interfaces
- Publish context data to LDES for historical access and replication
- Enable cross-RIL sensor data discovery through federated context brokers

8. NGSI-LD for context-aware agricultural data

NGSI-LD (next generation service interface - linked data) is an information model and API specification developed by ETSI and adopted by FIWARE. It is particularly relevant for context-aware agricultural applications requiring semantic interoperability.

Core NGSI-LD characteristics:

- Entity-based model with properties, relationships, and context
- Temporal representation of entity evolution (supports time-series data)
- Geospatial query capabilities (near, within, intersects operations)
- Subscription mechanism for real-time notifications
- Multi-tenancy support for data isolation
- Integration with linked data vocabularies (such as AIM, SAREF4AGRI)

Key NGSI-LD API operations:

- Entity creation, retrieval, update, and deletion (CRUD)
- Batch operations for multiple entities
- Query API with filtering, geospatial and temporal queries
- Subscription/notification for event-driven architectures
- Context sources for federation
- Temporal queries for historical data access

NGSI-LD response formats:

- Default JSON with UTF-8 encoding
- Content negotiation via Accept header
- GeoJSON support for spatial data
- JSON-LD for semantic enrichment and linked data
- Metadata inclusion (timestamps, versions, pagination links)

Integration with Demeter-AIM:

- Demeter agricultural information model (AIM) is based on NGSI-LD
- Property graph model rather than traditional RDF/OWL approach
- Cross-domain concepts combined with agricultural domain modules
- Reuses FOODIE, SAREF4Agri ontologies
- Comprehensive mappings to FOODON, AGROVOC, and other standards

Practical NGSI-LD implementation considerations:

- Choose NGSI-LD for scenarios requiring:
  - Real-time context data from IoT sensors
  - Geospatial queries (farm boundaries, equipment locations)
  - Temporal analysis (crop growth, weather patterns)
  - Event-driven subscriptions (threshold alerts, anomaly notifications)
  - Federation across multiple data sources}

ScaleAgData NGSI-LD application scenarios:

- Publishing sensor observations from RIL precision agriculture pilots
- Federated queries across multiple farm management information systems
- Real-time monitoring dashboards with subscription-based updates
- Integration with FIWARE OCB for context broker functionality
- Semantic interoperability across RILs using AIM vocabulary

2.7 High-level architecture per RIL

The subsequent sub-chapters detail the current high-level architecture of each RIL, incorporating the functional modules described in Chapter 2.3.

2.7.1 High-level architecture of RIL 1 – Water Productivity

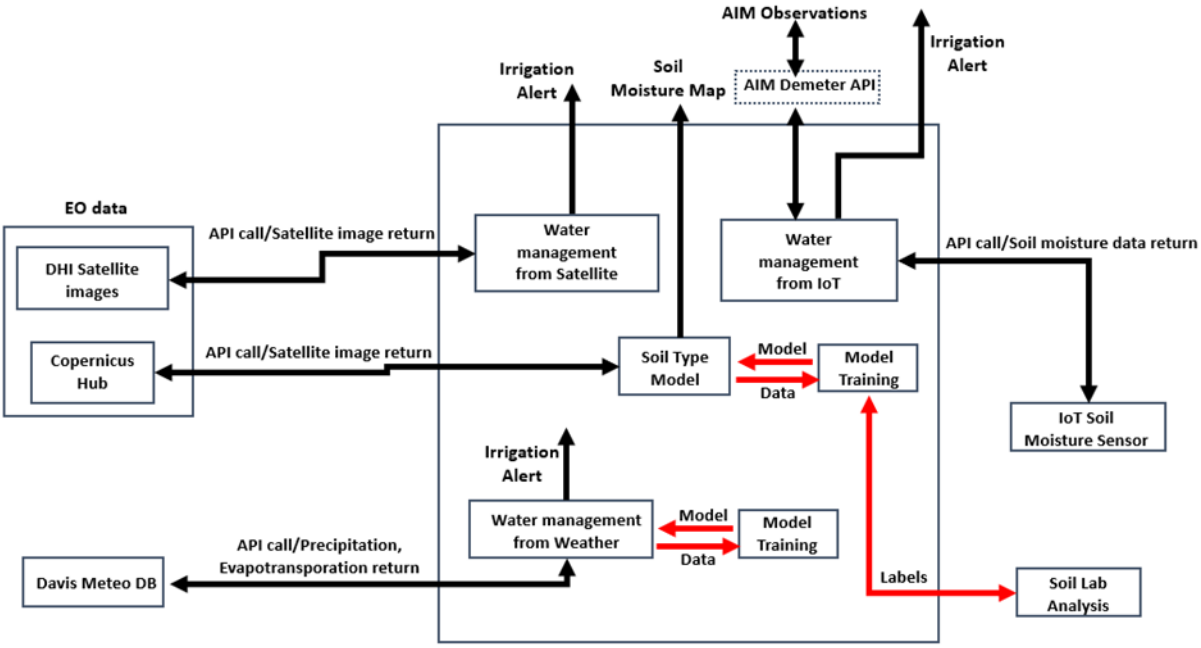


Figure 46: Water Productivity RIL high level architecture

2.7.2 High-level architecture of RIL 2 – Crop Management

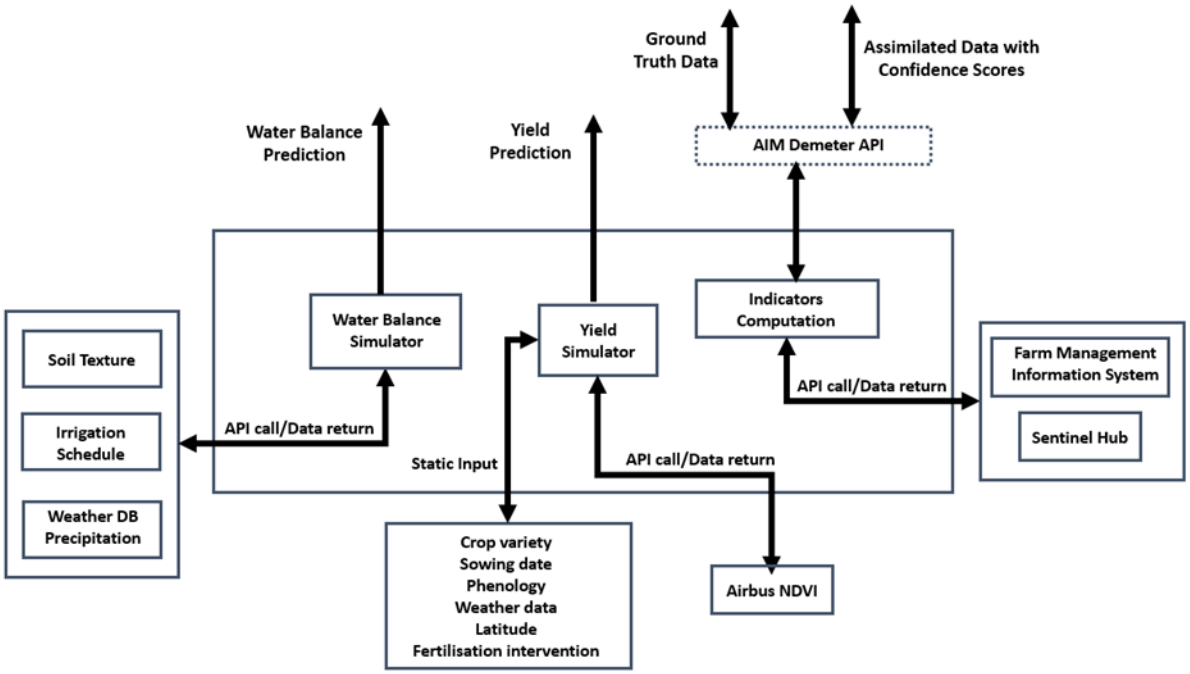


Figure 47: Crop management high level architecture

2.7.3 High-level architecture of RIL 3 – Yield Monitoring

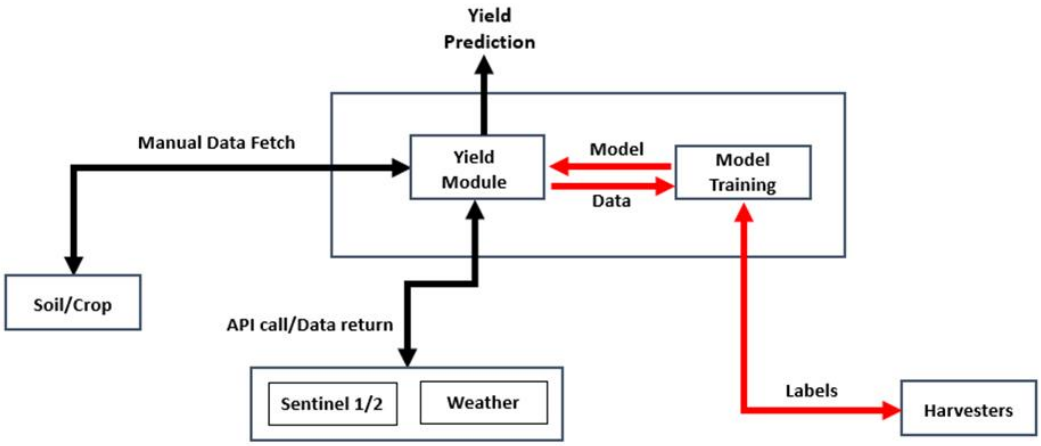


Figure 48: Yield monitoring RIL high level architecture

2.7.4 High-level architecture of RIL 4 – Soil Health

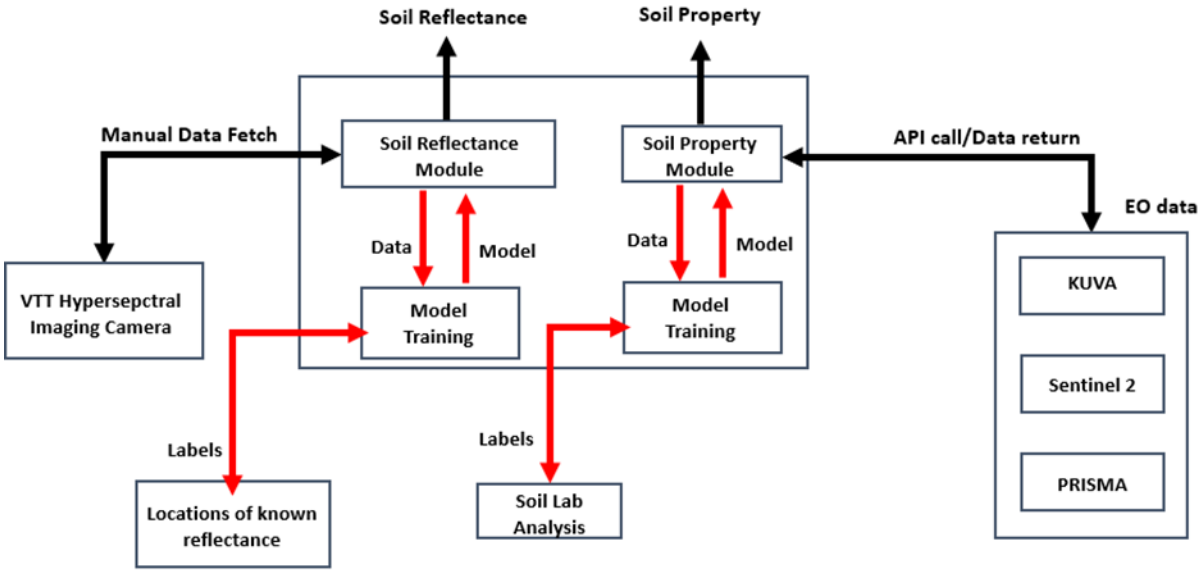


Figure 49: Soil health RIL high level architecture

2.7.5 High-level architecture of RIL 5 – Grassland

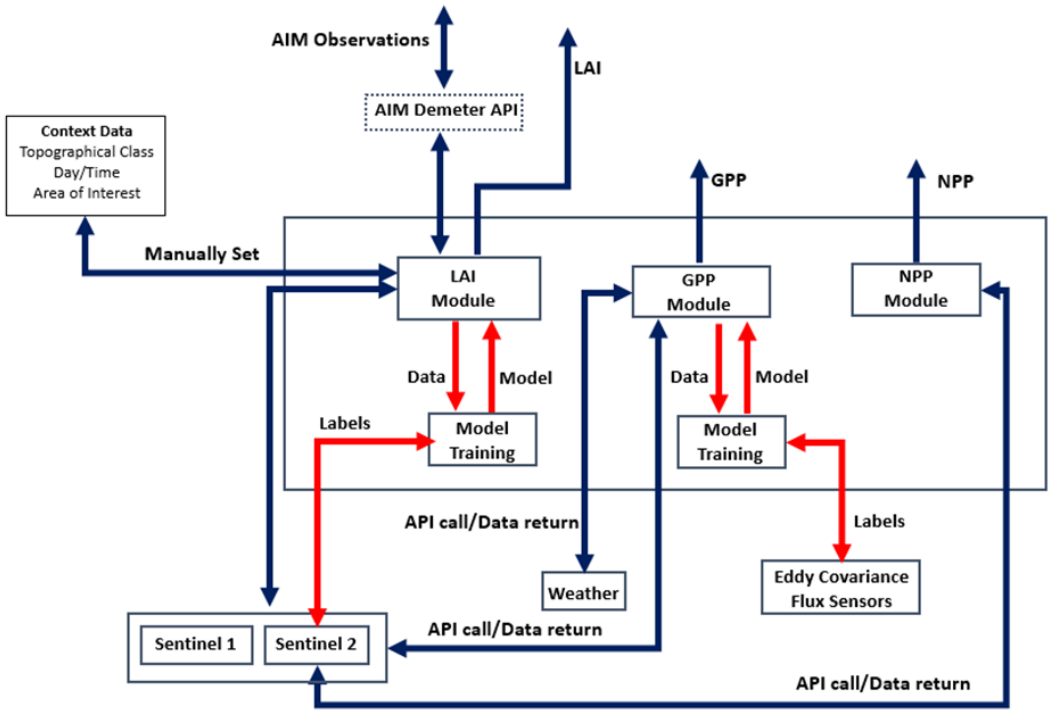


Figure 50: Grassland RIL high level architecture

2.7.6 High-level architecture of RIL 6 – Sustain Dairy

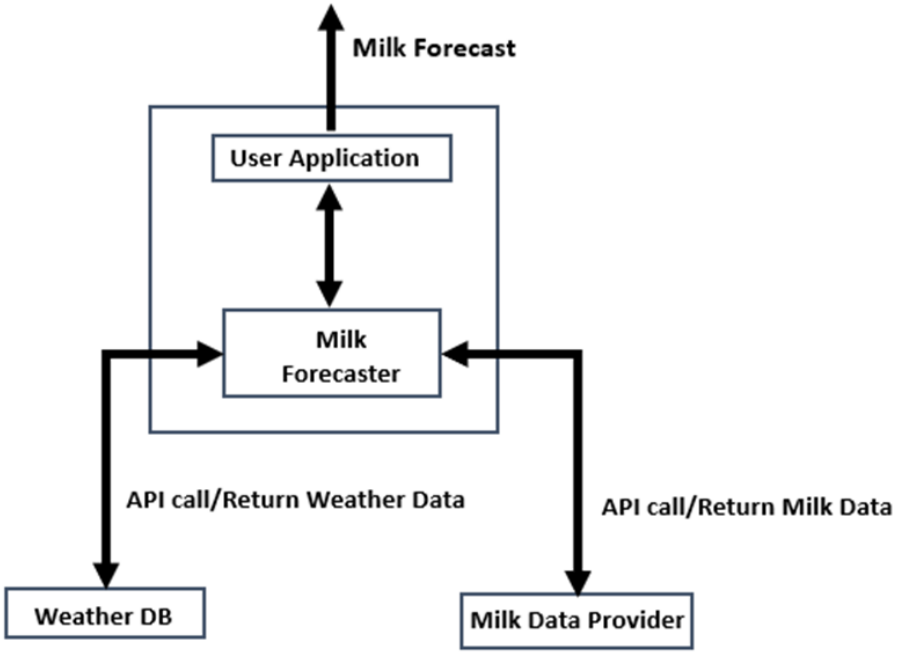


Figure 51: Sustain dairy RIL high level architecture

2.8 Information View

Static information view

The following information view summaries present the structured description of the datasets collected and managed within each RIL of the ScaleAgData project. Each table provides a concise overview of the sensor, satellite, and experimental datasets that form the foundation for the development, validation, and integration of the project's data-driven tools and services.

Each RIL is responsible for maintaining and updating its own catalogue sheet throughout the project's duration. New data collected from successive campaigns or growing seasons is entered as a separate dataset entry. Each RIL Data Catalogue Summary table provides an overview of the data. The columns are harmonized across all RILs and typically include:

- Availability: defines the access level (open / restricted / proprietary).
- Data Set Description: short technical description of the dataset contents and purpose.
- Data Category / No. of Sensors: classification and scale of data acquisition.
- Data Volume: approximate data size.
- Data Format: storage or exchange format (CSV, JSON, GeoTIFF, etc.).

Table 5: RIL Water Productivity

Availability	Data Set Description	Data Category / No. of Sensors	Data Volume	Data Format

Open	Irrigation quantity (m^3 day^{-1} ha^{-1})	Raw / —	< 1 MB / ha	xlsx
Open	Meteorological data (air T, RH, precipitation, solar radiation, soil T, ET)	Sensor / 1	~ 10 MB / ha	csv
Open	Soil moisture (SM)	Sensor / 12	~ 10 MB / ha	csv
—	Meteorological data (air T, RH, precipitation, wind speed & direction, solar radiation)	Sensor / 1	—	—
—	Soil moisture and temperature at 10 cm depth	Sensor / 4	—	—

Table 6: RIL Crop Management

Availability	Data Set Description	Data Set Category / No of Sensors	Data Volume	Data Format
Restricted	IoT agrometeorological station time series: air temperature, RH, rainfall, wind speed/direction, solar radiation, pressure, leaf wetness. Hourly sampling; ingested to NP cloud via REST API.	In-situ / IoT – 18 pilot deployments	MB size (daily aggregations)	JSON (API); csv/xlsx exports
Restricted	Spray classification from nanoparticle gas-sensing array at edge; detects no spray / water / pesticide spray; includes ambient T, RH, and sensor signals.	In-situ / IoT (derived) – 1	KB size	JSON (REST payloads); csv exports
Restricted	Farm records ("farm calendars"): irrigation, pesticide spraying, harvest events used as contextual labels.	Farm management logs – several per farm	MB size	JSON-LD (AIM); csv
Open	EO imagery for crop classification: Sentinel-2 L2A bands (B2–B6), NDVI, spectral features; used in SVM classifier.	EO (Satellite)	GB size	GeoTIFF (L2A, NDVI rasters)
Open	Ancillary vectors: parcel polygons, crop type labels,	Vector GIS (labels/masks)	MB size	Shapefile / GeoPackage / GeoJSON

	LAU boundaries, land-use mask.			
Open	Real-time meteorological data from 540 stations across Poland; per-minute measurements (T, RH, wind, precipitation).	Sensor network / 540	Hundreds of KB per year per station	JSON
Restricted	Weather parameters (WV, RR, AT, LW, RH, SR).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (AT, RH, LW, RR).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (SR, RR, RH, WV, AT, LW).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (AT, RR, LW, RH, WV).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (AT, RR, RH, LW).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (RR, AT, RH, LW).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (RH, RR, LW, WV, AT).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (AT, WV, WD, RH, RR, LW).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (RR, SR, RH, AT, WV, LW).	Sensor / 1	≤ 500 KB per year	csv
Restricted	Weather parameters (AT, RR, LW, RH).	Sensor / 1	≤ 500 KB per year	csv

Table 7: RIL Yield Monitoring

Availability	Data Set Description	Data Category No. of Sensors	Data Volume	Data Format

Proprietary	Yield data for potatoes per square meter collected from field harvesters; integrated with AVR Connect platform.	Raw / 10 sensors	~10 MB per field	JSON, csv
Proprietary	Wheat yield (kg/ha) data acquired by onboard harvester sensors.	Sensor / 4	~1 MB per field	GeoJSON, CN1
Proprietary	Augment index derived from onboard camera analysis.	Sensor / 1	~1 MB per ha	GeoJSON
Proprietary	Electrical conductivity (EC) at four soil depths for soil profiling.	Sensor / 1	~100 KB per ha	GeoJSON
Proprietary	Meteorological data: air temperature, soil temperature, rainfall, relative humidity.	Sensor / 10	—	—
Restricted	Soil moisture measurements collected manually (12 per field).	Sensor / 1	< 1 MB per ha	CSV
Restricted	RGB images for crop status and canopy cover assessment (10–15 per field).	Imaging / 1	< 100 MB per field	JPG
Restricted	Radiation (PAR) measurements for light-use efficiency analysis.	Sensor / 1	< 1 MB per ha	csv
Restricted	Chlorophyll content readings on crop leaves.	Sensor / 1	< 1 MB per ha	csv
Restricted	Leaf quality index readings (chlorophyll and flavonol content).	Sensor / 1	< 1 MB per ha	csv
Proprietary	RGB camera installed on potato harvester for visual yield estimation.	Imaging / 1	< 5 GB per field	JPG

Table 8: RIL Soil Health

Availability	Data Set Description	Data Category No. of Sensors	Data Volume	Data Format
Proprietary	Soil samples – topsoil organic carbon (SOC) collected and analyzed in laboratory.	Lab Analysis / —	64 KB	xlsx
Open	Hyperspectral images of agricultural fields captured using drone-mounted cameras for soil assessment.	Sensor Data / 1	81.2 GB	NPZ

Open	Copernicus Sentinel-2 Level-2A Earth Observation data used for soil property mapping and validation.	EO (Satellite) / —	—	TIFF
Proprietary	Soil samples – topsoil organic carbon (SOC) laboratory measurements.	Lab Analysis / —	—	—
Proprietary	PRISMA hyperspectral imagery for soil health and vegetation analysis.	Sensor Data / 1	—	—
Open	Drone-based hyperspectral images of agricultural plots.	Sensor Data / —	—	TIFF
Open	LUCAS 2018 TOPSOIL dataset (European soil organic carbon benchmark).	Lab Analysis / —	3.6 MB	xslx

Table 9: RIL Grassland

Availability	Data Set Description	Data Category No. of Sensors	Data Volume	Data Format
Restricted (to be open after publication)	Bio-physical parameters – Leaf Area Index (LAI) measurements over multiple agricultural parcels.	Raw / 4 sensors	TBD	csv
Restricted (to be open after publication)	Bio-physical parameters – Photosynthetically Active Radiation (PAR) for grassland plots.	Raw / 4 sensors	TBD	csv
Restricted (to be open after publication)	Bio-physical parameters – Chlorophyll content (CC) measurements.	Raw / 4 sensors	TBD	csv
Restricted (to be open after publication)	Bio-physical parameters – Soil moisture measurements over S2 sampling parcels.	Raw / 4 sensors	TBD	csv
Restricted (to be open after publication)	Carbon and energy fluxes (eddy covariance data) for grassland productivity and flux studies.	Sensor Data / 4 sensors	TBD	csv
Open	Meteorological variables: air temperature, humidity, rainfall.	Sensor Data / 3–4 sensors	TBD	csv
Restricted	Soil moisture (3 depths) in flux tower footprints.	Sensor Data / 5 sensors	TBD	csv
Restricted	Field radiometry over 10 monitored sites.	Sensor Data / 1 sensor (10 sites)	TBD	csv
Restricted	Bio-physical parameters – Leaf Area Index (LAI) for canopy structure monitoring.	Sensor Data / 1 sensor (10 sites)	TBD	csv

Restricted	Bio-physical parameters – fraction of Photosynthetically Active Radiation (fPAR).	Sensor Data / 1 sensor (10 sites)	TBD	csv
Restricted	Biomass data collected at 10 grassland sites.	Sensor Data / 2 sensors (10 sites)	TBD	csv

Table 10: RIL Sustain Dairy

Availability	Data Set Description	Data Category No. of Sensors	Data Volume	Data Format
Proprietary	Milk quality and quantity data (fat, protein, volume) from ~1,000 farms in near-real-time and ~10,000 historical records.	Lab Analysis / Milk Quality Forecasting	~13 GB	JSON
Proprietary	Farm metadata including location, farm type, and management attributes associated with milk records.	Metadata / —	Included in above	JSON
Open	Vegetation indices derived from Sentinel-2 satellite imagery for feed production assessment.	Satellite Imagery / —	~100 MB per date	GeoTIFF
Open	Meteorological data (temperature, precipitation) from ECMWF ERA5 reanalysis for environmental correlation.	Meteo-Reanalysis / Data Cube	~20 MB	Zarr
Restricted	Biomass and dry matter production (kg/ha/day) derived from VITO models combining Sentinel-1, Sentinel-2, and meteorological data.	Model Output / —	~1 MB	csv
Proprietary	Harvester data including yield per area and dry matter percentage for feed production tracking.	Sensor / —	~140 MB	csv
Open	Weather forecast data (NRT and historical) from ECMWF IFS model including temperature and precipitation.	Weather Forecast / Data Cube	~20 MB	Zarr

Information Flow

The Information Flow within ScaleAgData shows the communication pathways that facilitate the data collection and sharing. As shown by the previous analysis there are various components and enough diversity in the technological foundations and functionalities that are integrated, so the Information Flow will be a high level one trying mostly to show some generalized common data flows focusing on data sharing. Not all security requests are shown as it may depend on the specific protocols used internally by each RIL, which may depend for example on the sensor supported protocols, and their security practices may vary.

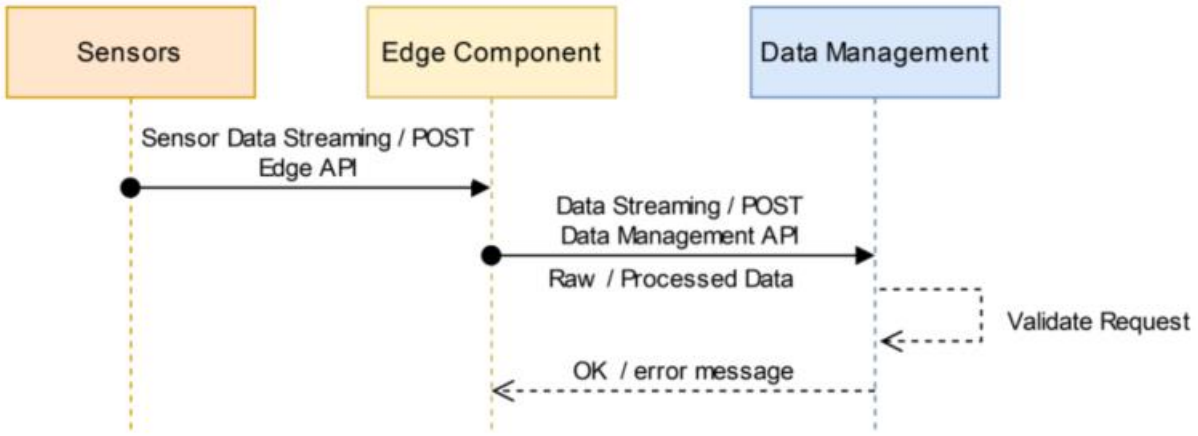


Figure 52: Sensor to Data Management Information Flow

The sequence diagram shown in the Figure 52 shows a simple flow from the sensors to the Data Management layer. The sensors may send data first to the Edge Component using their supported transfer protocol. The Edge Component afterwards, may transmit the data directly to the data management layer or do a local processing and transmit processed Data depending on the use case. Security best practices should be followed during this data transfer that are not shown, as it is internal to each RIL, but a validation of any request, or data streaming interface must be applied.

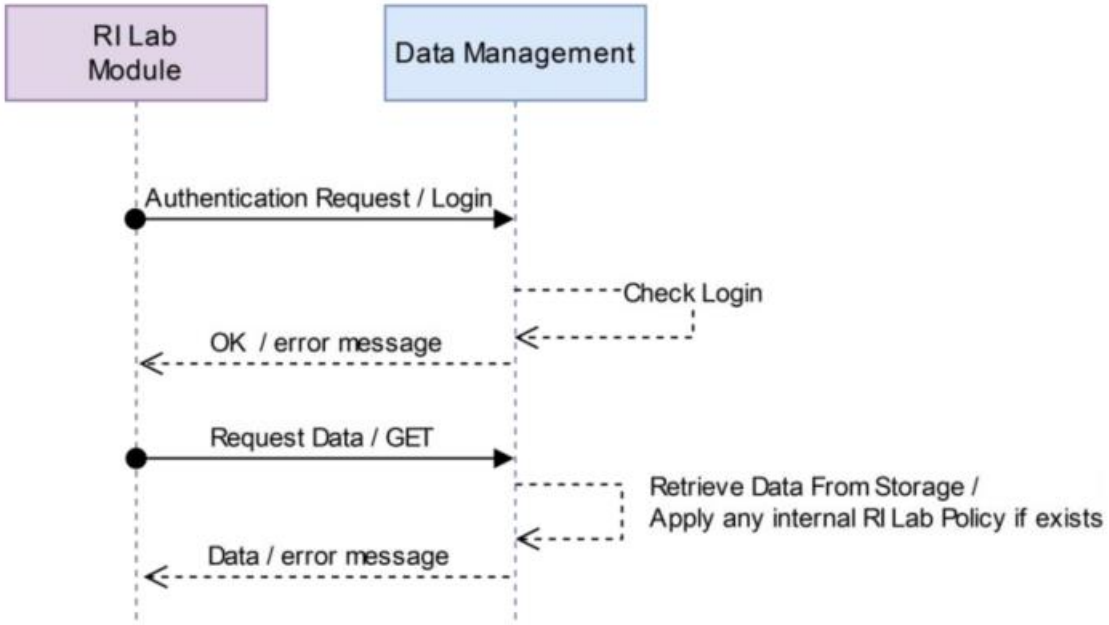


Figure 53: Data Management to RIL Module Information Flow

Each RIL develops some internal modules guided by research interests. These modules can get data from the Data Management platform to perform their analysis. The sequence flow is shown in Figure 53. The internal module must make a login request to gain access to the platform and then request the data that it needs. Any internal to the RIL Authorization policy upon the data may be applied before retrieving and sending the data back to the internal module.

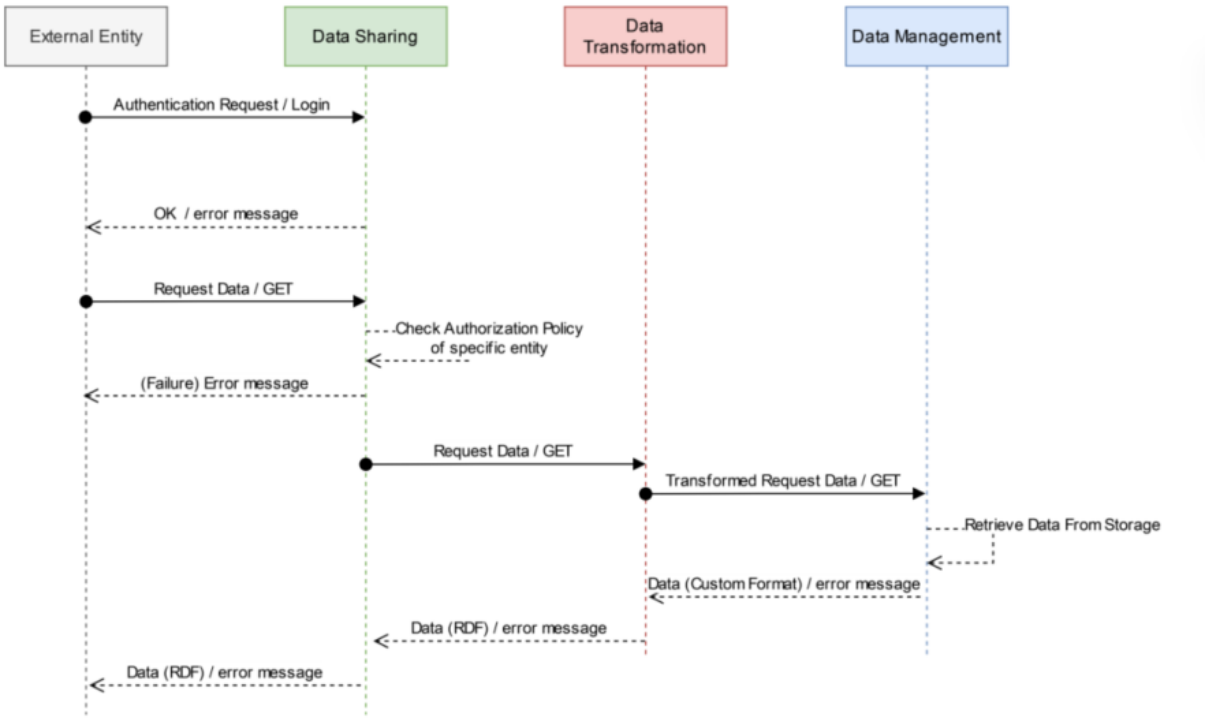


Figure 54: Data Sharing Sequence Flow

Figure 54 shows the data sharing sequence flow. Any external entity, for example another RIL, the RIE, can request data using the public Common API exposed by each RIL. An authentication request must be validated first to identify the external entity. Afterward the external entity can request data that it may be interested in retrieving from the RIL. Afterwards any policy is checked to verify whether this external entity is authorized to access the requested data. If there is such policy then the data are retrieved from the data management layer, are transformed to the common data model based on RDF vocabularies and then responded back to the external entity.

3. References

1.Woods, E. (2005). Software Architecture Using ViewPoints and Perspectives. SET2005. Zurich.

2.Michael A. Ogush, D. C. (2000). A Template for Documenting Software and Firmware Architectures.

3.Nick Rozanski, E. W. (2005). Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives.

4.Magerkurth, C. (2012). IoT-A Deliverable D1.4 Converged architectural reference model for the IoT v2.0. IoT-A Consortium.

5.Wood, E & Rozanski, N (2005). Understanding Architectural Perspectives.

6.C. Brewster, I. Roussaki, N. Kalatzis, K. Doolin and K. Ellis, "IoT in Agriculture: Designing a EuropeWide Large-Scale Pilot," in IEEE Communications Magazine, vol. 55, no. 9, pp. 26-33, Sept. 2017, doi: 10.1109/MCOM.2017.1600528

7."D4.2: Data interoperability for the agri-food sector", June 2021. https://ploutos-h2020.eu/wp content/uploads/2022/11/Ploutos_D4.2-Data-interoperability-for-the-agri-food-sector_v1.0_2021 06-11.pdf