D2.4 DEMETER Data and Knowledge extraction tools – Release 2

Dissemination level: Public
Submission date: 31.05.2021

Contents

This document is issued within the frame and for the purpose of the DEMETER project. This project has received funding from the European Union's Horizon2020 research and innovation programme under Grant Agreement No. 857202. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

List of figures

List of tables

1 Executive Summary

DEMETER aims to lead the Digital Transformation of the European Agri-food sector based on the rapid adoption of advanced technologies. Such technologies are Internet of Things, Artificial Intelligence, Big Data, Decision Support, Benchmarking, Earth Observation, and so on. Also, Demeter aims to increase performance in multiple aspects of farming operations, as well as to assure the viability and sustainability of the sector in the long term. Accordingly, DEMETER will put these digital technologies at the service of farmers using a human-in-the-loop approach that constantly focuses on mixing human knowledge and expertise with digital information. DEMETER focuses on interoperability as the main digital enabler, extending the coverage of interoperability across data, platforms, services, applications and online intelligence, as well as human knowledge, and the implementation of interoperability by connecting farmers and advisors with providers of ICT solutions and machinery.

To enable the achievement of the aforementioned objectives, DEMETER has already delivered a Reference Architecture (RA) that is suitable to address these challenges in the agri-food domain. DEMETER RA aims to facilitate the collection, processing and usage of the data used by DEMETER enabled pilots and to provide an integrated view over different and heterogenous datasets that can support the discovery and extraction of new knowledge. The last-mentioned requirements are handled by the *Data & Knowledge (DK) enablers* which have already been discussed in D2.2 and are adjusted in this deliverable.

One of the main elements framing the implementation of the different DK enablers is the *DEMETER Agricultural Information Model (AIM)*, which provides the basis to enable the semantic interoperability between different systems and data models. AIM, described in detail in D2.1, is providing the common vocabulary that is used to exchange data between different components, and to provide the integrated view over different and heteronomous data sources.

The DK enablers are part of the set of advanced DEMETER enablers. In particular, the DK enablers include facilities for data collection & preparation to collect, curate and prepare the data, data integration & linking to provide an integrated data view over heterogeneous sources, data fusion to fuse the data collected, data management to support the users' stated preferences, and data analytics & knowledge extraction for further processing of the fused and integrated data. These facilities are mapped into three enablers which are presented in this deliverable: first, *Data Preparation & Integration*, second, *Data Management*, and third, *Data Analytics & Knowledge Extraction*. Moreover, this deliverable also covers the *data security protection facilities*, split among the Core and the Advanced Enablers, which aim for example to ensure secure transfer of sensitive data or to prevent access to unauthorized entities.

The design and implementation of the different enablers is based on an exhaustive analysis of the state of the art, which has been presented in D2.2. In the first part of this document we present some updates of the state of the art. Then, the deliverable includes a further discussion framing the DK facilities and enablers within DEMETER RA, and continues with a detailed description of each of the enablers. For each of these sections, a description of the approach is provided, followed by a presentation of the design, including UML diagrams, some include even examples how the communication via AIM looks like. Finally, the document concludes possible challenges and obstacles that may occur in the second round of the pilots. In the Annex, the ongoing implementation work can be found. Also, the requirements driving the implementation of these enablers are presented in the Annex, namely in five different categories:

- Data Integration including semantic Interoperability and integration requirements,
- Data Management, including CRUD, data storage, synchronization, translation to/from various data access methods and query languages, data discovery, data aggregation, etc.,
- Data Quality & Fusion,
- Data Analytics & Machine Learning,
- Data Security & Privacy.

Note that this document is an extension of the initial release of the data and knowledge extraction tools presented in D2.2. A final deliverable is planned for October 2022.

2 Acronyms

AA	Attribute Authority
ABE	Attribute-Based Encryption
ACS	Access control server
AGRIS	International System for Agricultural Science and Technology
AIM	Agriculture Information Model
AIS	Agricultural Interoperability Space
API	Application Program Interface
BDE	Big Data Europe
BDE2020	Big Data Europe 2020
BSE	Brokerage Service Environment
CLI	Command Line Interface
CM	Capability Manager
CSV	Comma-separated values
CT	Capability Token
DCapBAC	Distributed Capability-Based Access Control
DEE	Demeter Enhanced Entity
DEH	DEH Enabler HUB
DEMETER RA	DEMETER Reference Architecture
DES	Data Encryption Standard
DK	Data & Knowledge
DPI	Data Preparation and Integration
DQ	Data Quality Assessment
DQA	Data Quality Assessment
DSS	Decision Support System
D2.1	DEMETER Deliverable D2.1
D2.2	DEMETER Deliverable D2.2
ETL	Extract, transform, load
FADN	Farm Accountancy Data Network
FOODIE	Farm-oriented Open Data In Europe
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
ha	hectare
HDFS	Hadoop Distributed File System
HTTP	HyperText Transfer Protocol
IdM	Identity Management
IoT	Internet of Things
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
LPIS	Land Parcel Information System
MAC	Mandatory Access Control
ML	Machine Learning
MLOps	Machine Learning Operations
OAuth	Open Authorisation
OIDC	OpenID Connect
OMG	Object Management Group
PAP	Policy Administration Point
PDP	Policy Decision Point

PEP	Performance Enhancing Proxy
R2RML	RDB to RDF Mapping Language
RA	Reference Architecture
RDF	Resource Description Framework
REST	Representational State Transfer
RI	Risk Index
RML	RDF Mapping Language
RMSE	Root Mean Square Error
RRM	Resource Registry Management
SANSA	Semantic Analytics Stack
SOAP	Simple Object Access Protocol
SOCS	Stakeholders Open Collaboration Space
SPARQL	Simple Protocol and Rdf Query Language
SQL	Structured Query Language
SQuaRE	Software Product Quality Requirements and Evaluation
UAV	Unpiloted Aerial Vehicle
UML	Unified Modeling Language
url	Uniform Resource Locator
UUID	Universally Unique Identifier
W3C	World Wide Web Consortium
WP2	DEMETER work package 2
WP3	DEMETER work package 3
WP4	DEMETER work package 4
WP5	DEMETER work package 5
XACML	Extensible Access Control Markup Language
XML	Extensible Markup Language

3 List of Authors

Company	Author
Fraunhofer IESE	Christof Schroth (Lead)
Fraunhofer IESE	Henning Barthel
Fraunhofer IESE	Adam Trendowicz
Fraunhofer IESE	Anna Maria Vollmer
ATOS	Tomás Pariente Lobo
ATOS	Sergio Salmerón
DNET Labs	Senka Gajinov
DNET Labs	Nenad Gligoric
DNET Labs	Srdjan Krco
Fraunhofer FIT	Till Döhmen
ICCS	Ioanna Roussaki
ROT	Lorenzo Bortoloni
ROT	Nadia Caterina Zullo Lasala
OdinS	Juan Andrés Sánchez
OdinS	Juan Antonio Martínez
UMU	Antonio Skármeta
UMU	Manuel Mora
TECNALIA	Belén Martínez
TECNALIA	Sonia Bilbao
m2Xpert	Hans-Peter Grothaus
m2Xpert	Victoria Helmich
Vicom	Raúl Orduna
Vicom	José Álvaro
Vicom	Xabier Etxeberria

4 List of reviewers

Company	Reviewer
ATOS	Diego Esteban (format only)
AGRICOLUS	Andrea Chiancone
TRAGSA	Eladio Rego

5 Document History

Version	Date	Change editors	Changes
0.1	08/03/2021	Anna Maria Vollmer (Fraunhofer IESE)	Created document with initial organisational content and ToC.
0.2	05/05/2021	Christof Schroth (Fraunhofer IESE)	Integrating input from partners until 4th May (taken from NextCloud), editing typos, commenting
0.3	07/05/2021	Christof Schroth (Fraunhofer IESE)	Integrating input from partners until 7th May (taken from NextCloud), editing typos, commenting
0.4	10/05/2021	Christof Schroth (Fraunhofer IESE)	Update of Section 10.1
0.5	12/05/2021	Christof Schroth (Fraunhofer IESE)	Integration of Section 1 & Section 5
0.6	13/05/2021	Manuel Mora (UMU)	UML diagrams for sections 10.2.6 & 10.2.7.
0.7	17/05/2021	Christof Schroth (IESE)	Adding section 12, Updating Section 2
0.8	18/05/2021	Nenad Gligoric (DNET)	Update of section 10.2.5
0.9	18/05/2021	Christof Schroth (IESE)	Update of section 12.3 & 12.4; integrate input from ICCS
1.0	18/05/2021	Christof Schroth (IESE)	Final version to be submitted to peer review
1.1	21/05/2021	Christof Schroth	Revised final version to be submitted to review lead
2.0	27/05/2021	Diego Esteban (ATOS)	Final format quality check and final version for submission

6 Introduction

This deliverable presents the second release of the "DEMETER Data and Knowledge extraction tools". The work presented in this document is the output of task 2.2 (Data Management and Integration), task 2.3 (Targeted data fusion, analytics and knowledge extraction) and task 2.4 (Data Protection, Privacy, Traceability and Governance Management) of Work Package 2 (WP2) of the project. The results presented in this document realize the implementation of the Data & Knowledge (DK) enablers (which are part of the advanced enablers in DEMETER), and also which enablers use (or rely on, respectively) the DK repository. This repository carries any data or extracted knowledge that may eventually be stored by DEMETER locally. It is based on the DEMETER Agriculture Information Model (AIM) which is described in detail in D2.1. Also, this document includes the Data Security enablers, which provide services to both core and advanced enablers, where the formers are mandatory for creating any DEMETER application, while the latter are optional as described in detail in deliverable D3.3.

This document is an updated version of D2.2. After the end of round one we checked, if the DK are still appropriate. In particular, we checked if the requirements still need to be addressed, if they need to be adjusted or if they are out of scope for DEMETER. Recall that D2.2 contains a lot of background information like discussions of the design/approaches, details about the implementation and so on. Hence, this deliverable only lists updates that were done in terms of implementation. The rest of the document is structured as follows:

Section 7 provides updates on the state of the art (and state of the practice) on relevant methods and existing technological support related to the different topics covered by the DK enablers. In this case, only one subsection is added to the corresponding chapter in D2.2. Section 0 provides an overview of the changes in the technical requirements extracted by Task 2.2, Task 2.3 and Task 2.4. Section 0 presents a synopsis of the data and knowledge handing infrastructure provided by the DEMETER Reference Architecture (RA). This section discusses the place, role and relationship of the DK enablers in the framework of the general DEMETER architecture. Section 0 presents the data management components and the data preparation & integration components. This deliverable only contains updates of the components of D2.2., i.e. the on-going work in the implementation of the enablers. Section 0 summarizes updates of the data quality components, the targeted data analysis & data fusion components and the machine learning tools components. Section 0 summarizes the data security components, including the security architecture overview, the approach of the authentication, authorization, traceability and confidentiality, and the on-going work in the implementation of each of these components. Section 0 concludes the document presenting possible problems or obstacles that may future work, in particular for round two in the pilots.

Additionally, in the annex there are the updated technical requirements (Annex A.1) and a short summary of the status of the components in tabular form, where also for each component the related requirement(s) are listed (Annex 0).

The enablers presented in this deliverable complement the deliverable D3.3 DEMETER Reference Architecture.

7 Additions to Related Work

In this chapter we briefly summarize the updates of the related work section in D2.2. As there are no major changes in the requirements and as it looks like there seems to be no massive obstacles in the implementation work there is only one new section added in this chapter. The interested reader may refer to D2.2.

7.1 *Data Management and Integration*

No changes, please refer to D2.2.

7.2 *Analytics and knowledge extraction*

In this section we add one subsection to the original subsections in D2.2:

7.2.1 Model Management for Machine Learning

Model Management is an essential component of a modern, production Machine Learning infrastructure. It is sometimes also referred to as Experiment Tracking. Model Management solutions are concerned with storing and versioning Machine Learning experiments in a multi-user setting. As opposed to version control systems for software code (such as Git or SVN), Model Management solutions are optimized for storing large binary data (machine learning models) and machine learning-specific metadata, such as evaluation metrics. Most model management solutions offer a Command Line Interface (CLI) or API (Application Program Interface) for registering models and metadata in the repository. Also, they usually offer and a web-based graphical user interface to review registered models, evaluation metrics and more.

Model management solutions are part of the ecosystem of all modern machine learning platforms, such as Microsoft Azure [1], Google Cloud Platform [2], Amazon Web Services [3] and Databricks [4]. All these platforms embrace MLOps principles (see Figure 1), enabling the operation of productive Machine Learning systems in a managed and automated manner.

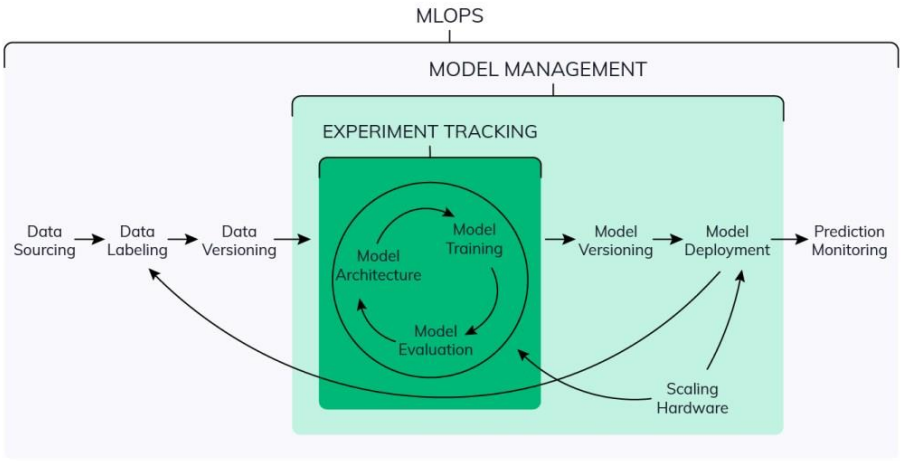

Figure 1: The role of Model Management in MLOps.[5]

[1] https://docs.microsoft.com/en-us/azure/machine-learning/concept-model-management-and-deployment
[2] https://cloud.google.com/ai-platform/prediction/docs/managing-models-jobs
[3] https://docs.aws.amazon.com/machine-learning/latest/dg/managing_objects.html
[4] https://databricks.com/de/product/managed-mlflow
[5] Source: https://neptune.ai/blog/machine-learning-model-management

Commercial products for model management are offered by the mentioned cloud platforms and other managed service providers, such as neptune.ai, and studio.ml, and polyaxon.com.

For the use in DEMETER, however, we considered available open-source solutions:

DVC[6] is a lightweight version control system for datasets and machine learning models. It is command-lines based and does not offer a graphical user interface[7].

ClearML[8] is an open-source MLOps-centric platform, which is also offered as commercial managed service by AllegroAI. The platform consists of three main components: Model Management, ML-Ops, and Data-Management. The model management component offers a Python API which provides data scientist an easy way to register models and experiment results. Major machine learning frameworks such as scikit-learn, xgboost, TensorFlow and PyTorch are supported by the API. There is currently no API for other programming languages such as R or Java. Via a Web-UI, users can view and manage registered models.

MLflow[9] is another open-source end-to-end platform for managing the machine learning lifecycle. This includes experiment tracking, model management as well as the deployment of models. Data scientists can use the MLflow Python/R/Java API to interact with the platform, which provides support for major machine learning frameworks such as scikit-learn, xgboost, TensorFlow and PyTorch and many more. Recently, support for the explainable-AI method SHAP were added to the API[10].

All presented open-source solutions offer extensive user documentation and installation instructions.

Other open-source projects such as Aim[11], steppy[12], Apache Marvin[13], and modelstore[14] were not considered due to comparatively low popularity and adoption (based on GitHub Stars, Forks, and PRs) at the time of writing.

7.3 *Data Protection, Privacy and Traceability*

No changes, please refer to D2.2.

[6] https://github.com/iterative/dvc
[7] https://github.com/iterative/dvc/issues/1074
[8] https://github.com/allegroai/clearml
[9] https://github.com/mlflow/mlflow
[10] https://mlflow.org/docs/latest/python_api/mlflow.shap.html
[11] https://github.com/aimhubio/aim
[12] https://github.com/minerva-ml/steppy
[13] https://github.com/apache/incubator-marvin
[14] https://github.com/operatorai/modelstore

8 Revised Technical Requirements

The technical requirements are extracted by Task 2.2, 2.3 & 2.4 (the technical requirements of Task 2.1 can be found in D2.3). They are organized according to the following topics:

- **DK3. Data integration:** Semantic Interoperability/integration Requirements
- **DK4. Data Management:** Including CRUD, data storage, synchronization, translation to/from various data access methods and query languages, data discovery, data aggregation, etc.
- **DK5. Data Quality & Fusion**
- **DK6. Data Analytics & Machine Learning**
- **DK7. Data Security & Privacy**

These five separate classes of requirements are an updated version of D2.2. In Annex A.1 there will be a complete list regarding the capabilities/functionalities that DEMETER must or should deliver with respect to data integration, including semantic interoperability, data management, data quality & fusion, data analytics, machine learning, data security & privacy.

The updates in this document are based on an ongoing exchange with WP3, WP4 & WP5. The original requirement tables from D2.2 (see Section 6 there) has been extended with one more field capturing the enabler(s)/module(s) addressing the respective requirement.

Altogether, the requirements did not change much since D2.2. Only a small number of them were updated in negligible points. The only significant change was done on DK5.12 which has been merged with DK5.11 because DK5.12 contained points which are out of scope for the project. A detailed description of the revised/extended technical requirements is provided in Annex A.1. Moreover, for reasons of completeness, the requirement fields that remain unchanged are also provided there.

9 Additions to Data & knowledge handling in DEMETER architecture

In this section, we present an updated synopsis of the data and knowledge handing infrastructure provided by the DEMETER Reference Architecture (RA), as it has been revised in D3.3. To this end, we mostly focus on the changes to the RA since D3.1, but also give some information, where appropriate, that has been present in the original architecture of D3.1 and has not changed in the update of the RA.

Now, the DEMETER RA aims to facilitate the collection, processing and usage of the data used by DEMETER enabled apps using a modular architecture that allows the creation of these apps from DEMETER enabled entities and various DEMETER enablers. In order for the various enablers to communicate with each other DEMETER provides the AIM model which allows semantic interoperability between DEMETER enablers and also, with the appropriate data wrappers to/from AIM, the interoperability of DEMETER enablers with existing systems, AKIS, and ontologies. Thanks to this modular approach the DEMETER RA is also compatible (or in line) with recent initiatives which promote the federation of data and services, see e.g. GAIA-X and the European data strategy.

Therefore, the modular approach of the DEMETER RA has not changed, in general, and it integrates heterogeneous technologies, platforms and systems, while supporting fluid data exchange across the entire agri-food chain, addressing scalability and governance of ownership. It is based on the ability of different enablers and systems (provided by different sources/entities potentially through the DEMETER Enabler Hub) to interoperate and to exchange data between them in order to form complete DEMETER enabled apps from all these offered systems and components. This is facilitated by the aforementioned common data model (the DEMETER AIM) based on which it is possible to create wrappers and enablers that transform the data from whatever data model is being used by the existing systems and sensors etc. that we want to incorporate into AIM data. For more information about the common data model and semantics, refer to deliverable D2.1 and the extensions presented in D2.3.

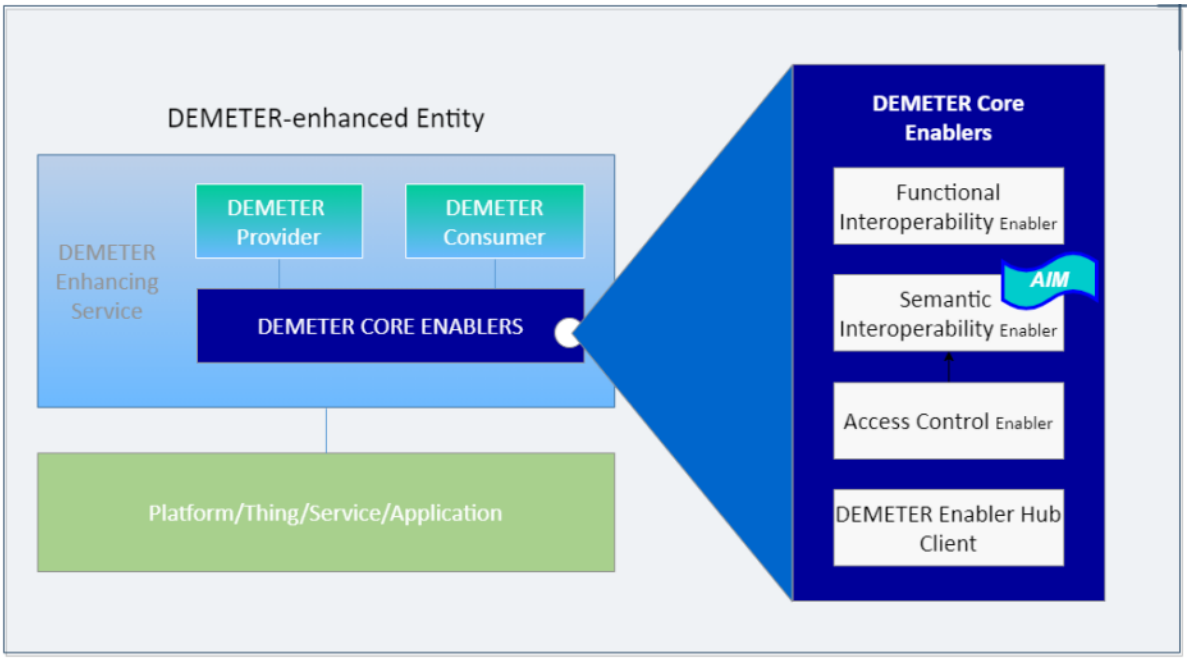

Figure 2: Updated Core Enablers offered by DEMETER (Figure 29 in D3.3)

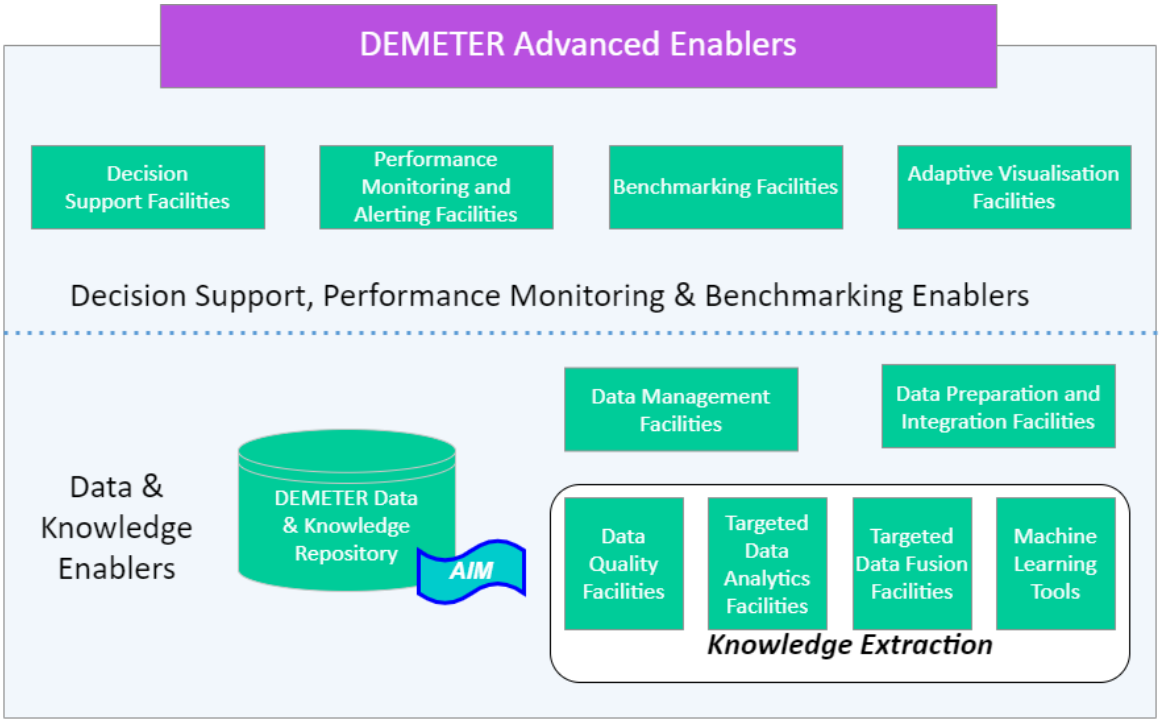

Figure 3: Updated Advanced Enablers offered by DEMETER (Figure 30 in D3.3)

DEMETER still offers a set of Core Enablers needed for creating any DEMETER applications, which are mandatory for DEMETER enabled apps, and also offers another type, i.e. Advanced Enablers, that are optional and are discoverable and accessible through the DEMETER Enabler Hub. The list of both core and advanced enablers has been updated in D3.3 and they are depicted respectively in the two figures above. Regarding the core enablers of DEMETER, these have been updated and the most important changes (not an exhaustive list) are the following: the Access Control enabler now provides the security and privacy functionality, while the DEH client enabler offers the runtime facilities needed. Regarding the advanced enablers, the updated Data & Knowledge Enablers presented in this deliverable (and in D2.2) are positioned at the lower layer and are responsible for Collecting and Curating data from the various sources that the DEMETER developers and stakeholders have been registered for. A number of Knowledge Extraction exist, which offer Data Quality Facilities, Targeted Data Fusion and Analytics and also other tools based on Machine Learning techniques. In this delivarable, we elaborate on the updates performed to all these aforementioned enablers since D2.2.

In addition, while the general concepts of DEMETER have not changed, some updates have been performed as depicted in the following Figure 4:

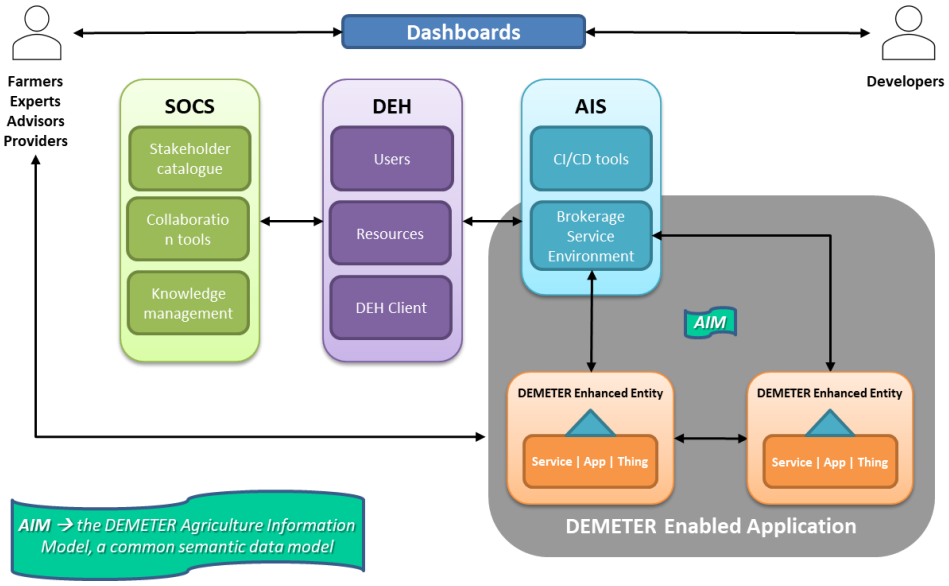

Figure 4: The updated main DEMETER concepts (Figure 25 in D3.3)

The DEMETER hub (DEH) is accessed via the appropriate dashboards and the DEMETER stakeholders still access it either via the DEMETER Stakeholders Open Collaboration Space (SOCS), which focuses on resolving the needs of the farmers, or via the DEMETER Agricultural Interoperability Space (AIS), which focuses on delivering a full set of interoperability mechanisms to develop, validate and then deploy the DEMETER enablers and even complete DEMETER enabled apps. Some updates have been performed to these components, as several tools have been updated. For example, the Brokerage Service Environment (BSE) now offers the deployment facilities for DEMETER apps, while the DEH client offers runtime facilities. For more details on these updates refer to D3.3.

Furthermore, the hub itself has been updated. While the main logic behind it remains, and it is used in order to register DEMETER enabler which can then be discovered in order to create DEMETER enabled apps, it now also includes components such as the Resource Consumption Monitoring which can be used in conjunction with the DEH client tool in order to provide the necessary runtime functionality needed by the final apps and systems deployed through DEMETER. These changes are depicted in the following figure:

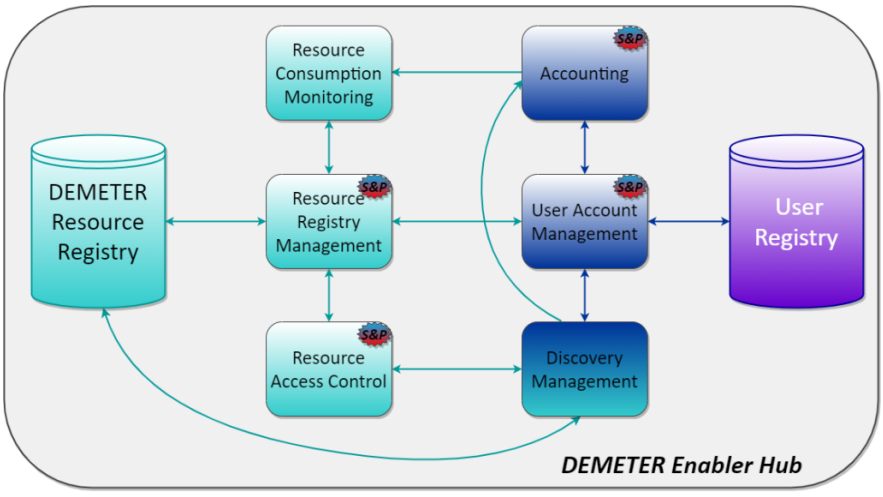

Figure 5: Updated Components and Registries of the DEMETER Enabler Hub (Figure 31 in D3.3)

All components of the RA have as a main driving force to allow interoperability between internal and external enabler and processes, as well as making information accessible at all architectural levels (e.g., between applications and services, user and application, users and services) up to the data visualization for the end-users through the DEMETER Dashboards. The RA is designed to enable the appropriate data storage and archival architecture, data retrieval, processing (and subsequent storage of the processed results) and security management. This is achieved via the appropriate kind of data spaces in the RA: the **DEMETER Data & Resource Repository**, the **DEMETER User Registry** and the **DEMETER BSE Registry**. All these are described in the figure that follows which depicts the updated (main) data flows between the various components in the DEMETER RA that facilitate all these procedures, while also highlighting the stakeholders involved:

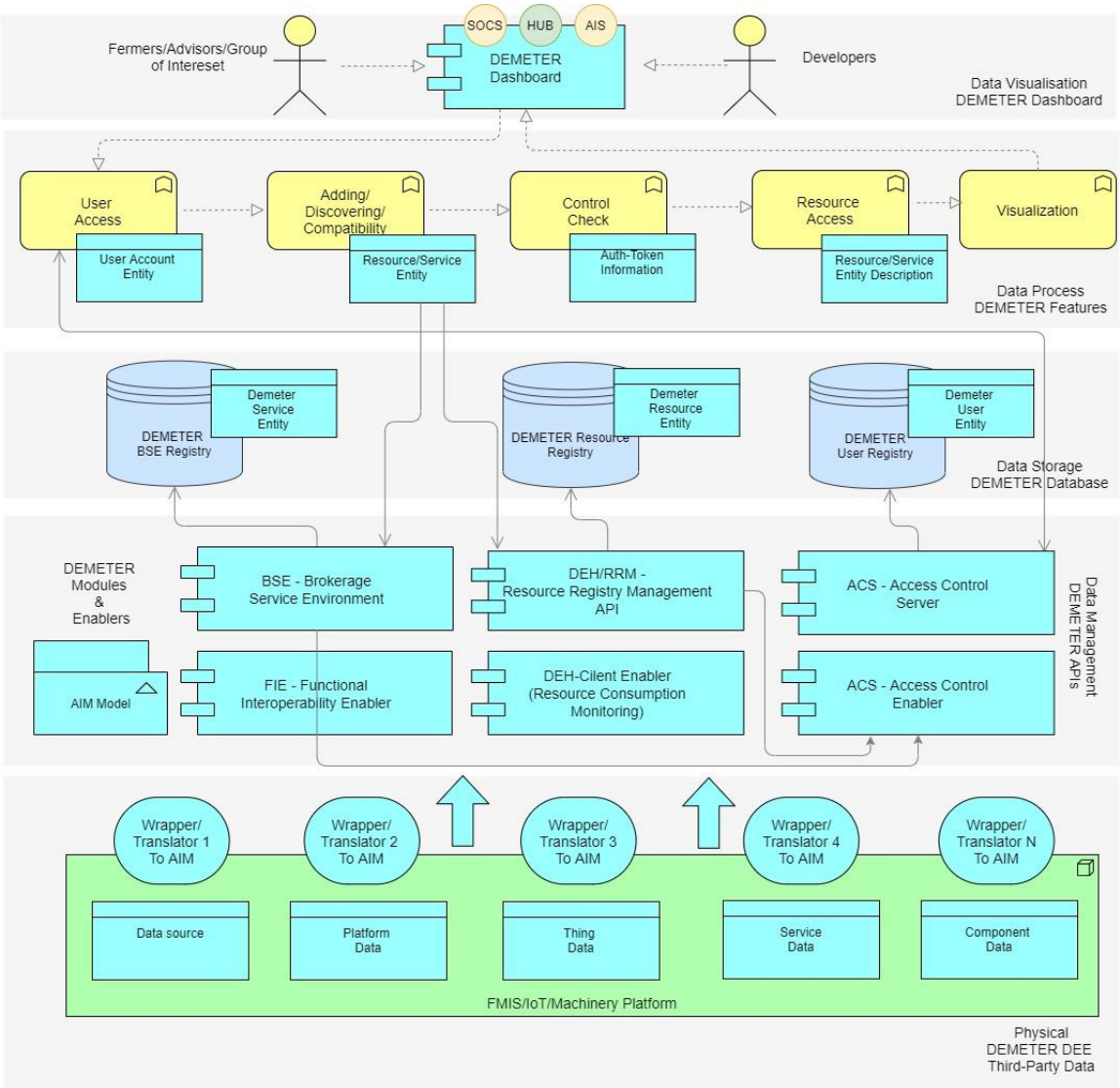

Figure 6: DEMETER Main Data Flows (Figure 36 in D3.3)

In more detail, the **DEMETER Resource Repository** (or **DEH Resource Repository)** addresses the problem of the physical representation of the DEMETER resource information model, normalizing the information flow coming from the DEMETER data providers using a formal definition of the DEMETER resources. It is based on the DEMETER AIM described in detail in D2.1 and updated in D2.3. The use

of appropriate data exchange models, knowledge representation languages and rule languages, which allow for semantic querying of data, have been applied in its design process. On the other hand, the **DEMETER User Registry** (or **ACS User Repository**) supports GDPR compliant storing and securing users' personal data, credentials, user permissions and other aspects of personal data, while the **DEMETER BSE Registry** records all resources, things, services or even entire platforms that are registered to DEMETER by their owners. Any interested data producer and anyone that can offer a device or a service (e.g. analytics or DS enablers) is required to register their services in this store, offering the application logic of their services to third parties or DEMETER consumers from an as-a-service perspective.

As depicted in Figure 6, the registered third-party resources (e.g., Thing, Platform, Service) can feed DEMETER with data that can be processed by the system and made available via the DEMETER Enabler Hub (DEH). Specific wrappers/translators are in place to allow for translation of foreign standardized or dominant data formats to AIM and vice versa, thus allowing for data interoperability in DEMETER, and this is required in order to become DEMETER compliant and be registered as a DEMETER enabler or a DEMETER enabled app. In this way, all these enablers (service and data and apps) can be discovered (via the appropriate DEMETER Dashboard) and consumed by other DEMETER apps, provided that they are interoperable with the AIM format.

In addition to the AIM compliance, central to the data handling facilities of the DEMETER RA are specific tools and components that have been developed enabling the data preparation, integration, fusion and analytics; also tools based on machine learning analysis of data. All these have been presented in D2.2 and an updated report of all these is presented in the following sections of this deliverable.

10 Data Management, data preparation and Integration components

This section aims to report the updated and final version of D2.2 regarding the DEMETER data management and integration enablers. It illustrates the main information related to the changes provided by the technical partners to align these software modules with DEMETER Reference architecture compared to D2.2.

10.1 *Data management*

Basically, the data management module is about data storage, data retrieval process and information security management. The section highlights the main changes related to the technological structure of this module made during the development phase, such as archiving and processing DEMETER resources, users and services. The data management system has been composed by a series of APIs frameworks, heterogeneous among them and consolidated technologies capable of supporting and manage the needs of DEMETER Pilot. This work involved both WP2 and WP3 technical partners. *However, compared to the first version (D2.2), there were no major deviations from the criteria for defining the technological methodology that inspired the design of this module at the beginning of the project*. However, during the implementation phase some aspects have been improved and enriched such as the technologies identification to support the solution and make the module consistent and suitable for the Pilot developers, consolidating the structure of services, processes and data flow. For planning and structuring the technical work carried out during the development and assembly phase of the entire module, the technical team has done a hard work of integration.

A more mature awareness on a technical level has ensured the technological module structure and the processes well defined by renewing underneath the positioning of this module within the DEMETER enablers architecture; in essence, the technical team worked to contextualize the set of APIs making up the module to the real project context engineer, having achieved greater technological awareness in this second iteration of the deliverable or D2.4.

The data management module block module consists of three main software sub-modules:

- ACS - Access control server
- DEH - DEMETER Enabler HUB
- BSE – Brokerage Service Environment

Each of these modules exposes standard APIs that depending on the case, perform specific tasks in the data management process. Below is a holistic figure (Figure 7), already used for Deliverable D3.3, which shows the whole flow of data coming from DEMETER data sources (Data Provider) and how these data are managed by the software sub-module of the data management system:

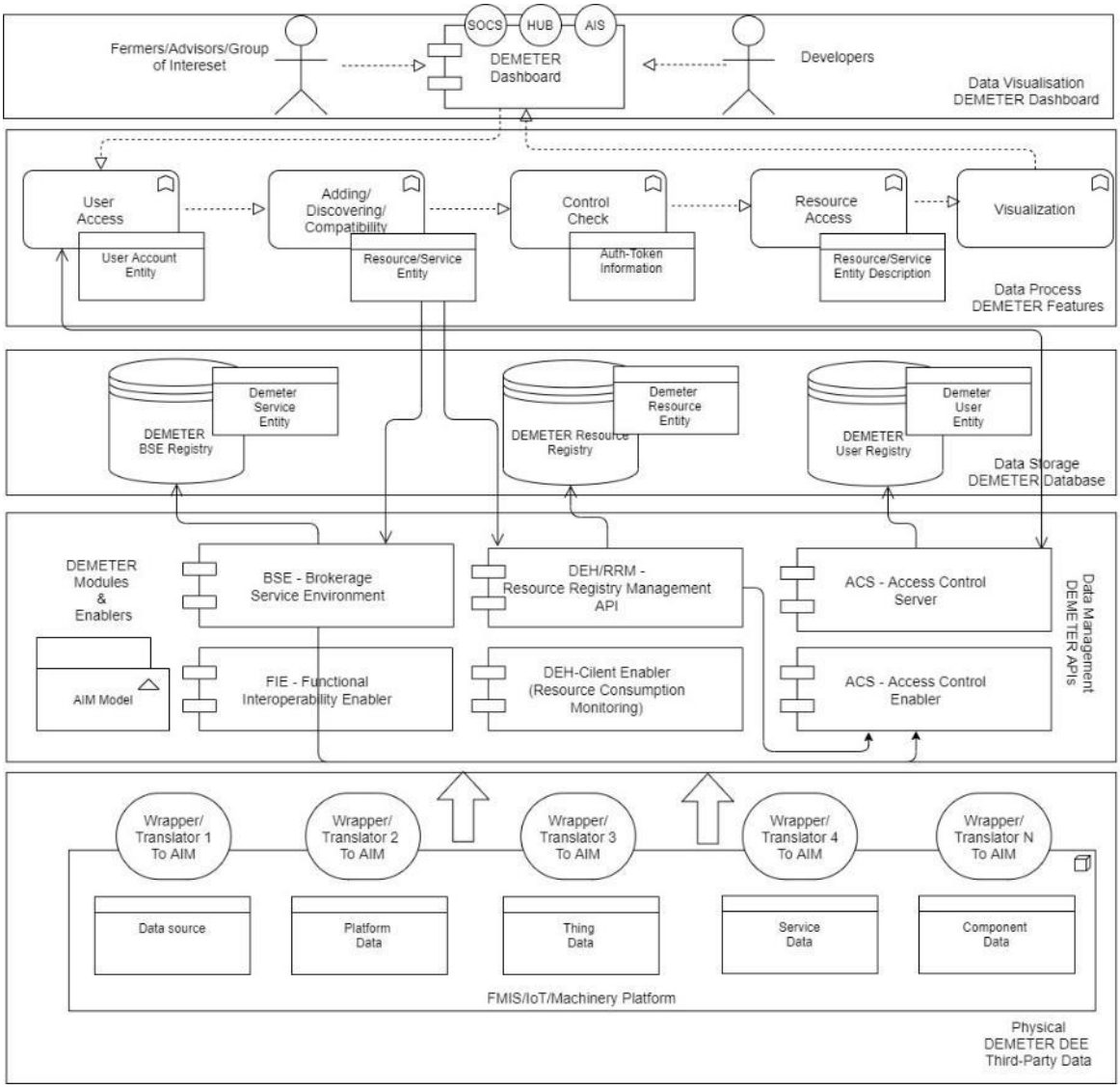

Figure 7: DEMETER Data Management Module

As can be seen from the figure, DEMETER data source that wants to share its own data and services interact with the data management module APIs. The data source, which in some cases translate the data into the AIM (Agricultural Information Model) data format, pushes the data to DEMETER management APIs. The entities are represented by users, resources and services; these latter entities are non-static entity that can expose specific endpoints or APIs. These APIs can be used within the DEMETER context from a Software-as-a-Service perspective by the Pilots who intend to use them for their own needs, in order to build their business scenarios. The data, acquired by the data management module APIs such as ACS (Access Control Server), DEH RRM (Resource Registry Management) and BSE (Brokerage Service Environment) are then stored in specific databases. As can be seen from the figure, the databases in which these data are saved are:

- DEMETER User Registry (where the user entities are stored)
- DEMETER Resource Registry (where DEMETER DEEs are stored)
- DEMETER BSE Registry (where the non-static DEMETER services are enabled which can be invoked at runtime and assembled together to other services to build business scenarios within the Pilots).

For more information about the technological structure of these databases, refer to deliverable D3.3. The information or entities, after being saved in the DEMETER database, are ready to be discovered by business processes and therefore always made available via APIs at the higher level of the architecture or at the presentation layer. The data can be used by DEMETER Consumer and Developers in full security manner guaranteed by ACS module.

All the changes compared to those defined in the first version of this document or D2.4 related to the data management module are reported in some UML diagrams, shown below in Figure 8, starting from the first version of these diagrams. Also, in this case, an internal high-level structure of this module is depicted, while the next figure (Figure 9) contains the main interfaces exposed by his own APIs:

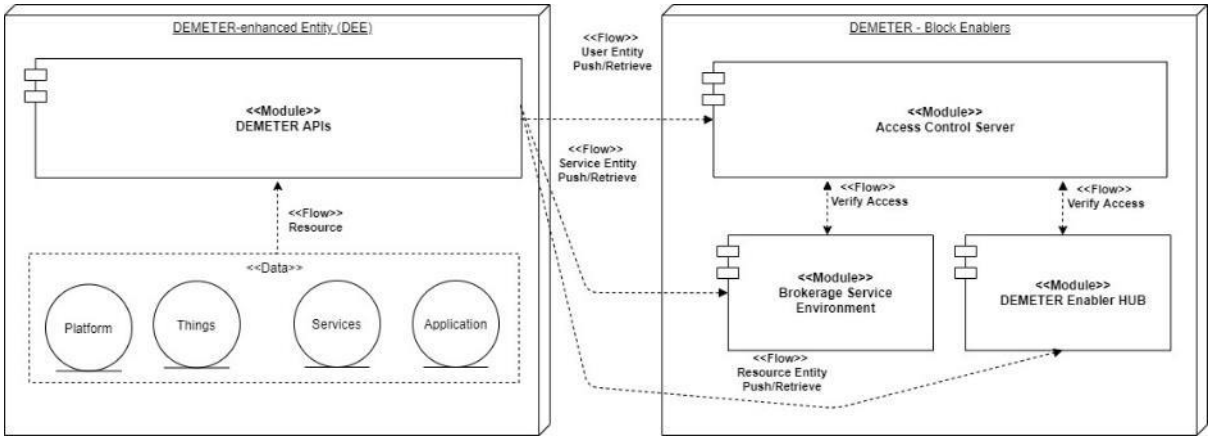

Figure 8: Enablers block for data management in DEMETER Project – revisited version compared to D2.2

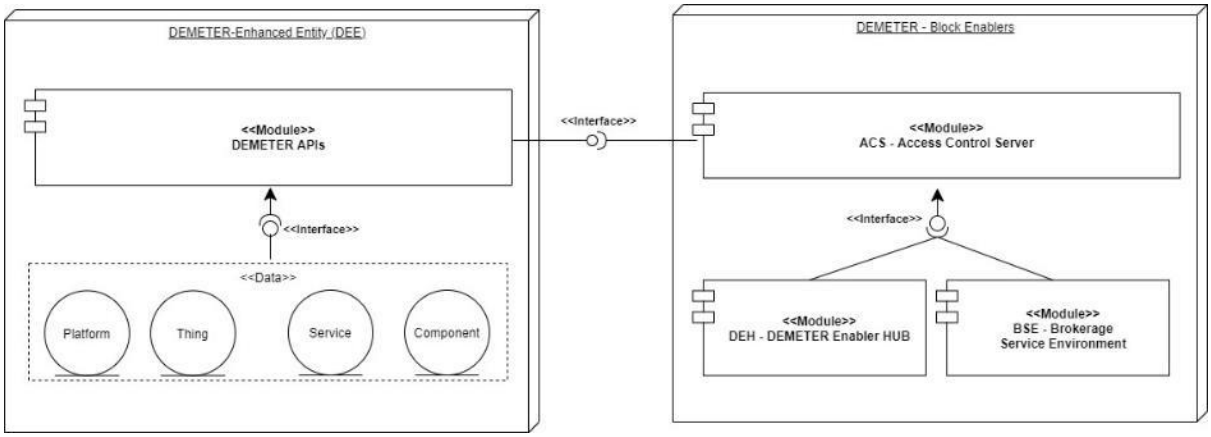

Figure 9: Main interfaces of Enablers block for DEMETER data management – revisited version compared to D2.2

Furthermore, in this iteration, D2.4 provides a view of the other's entire sub-module, with a more mature and comprehensive awareness of development and validation within WP5. The UML diagram presents a highs level view of this module: the data generated by different sources are transferred through DEMETER APIs to the DEMETER stores. Each operation step or request to DEMETER APIs must be authorized by the ACS enabler that guarantees the security among all the internal processes of the data management module. Only registered and authorized accounts can interact with DEMETER enablers and consequently store or retrieve data. Finally, the interfaces are represented by REST APIs

that exchange data in JSON format, in complete uniformity with the most common data interchange format on the Web.

10.2 *Data Preparation and Integration*

The Data Preparation and Integration Enabler provides a simple yet powerful interface to the functionalities of data preparation & integration that are based on the use of linked data as a federated layer, as described in D2.2. The enabler exposes a RESTful API that facilitates the exploitation of the underlying components via a homogenous layer, enabling other DEMETER enablers or enhanced entities to launch the whole Linked Data pipeline, or individual steps. The goal of the enabler is to abstract the different types of interfaces and implementation details through a simple to use interface. Additionally, the enabler facilitates access to the integrated data, represented according to DEMETER AIM, available in the Data & Knowledge repository. Accordingly, access to the data is possible in two different ways: i) via SPARQL queries directly submitted to the repository's SPARQL endpoint; ii) via the API that exposes pre-defined queries as simple access methods, which may be more convenient and require less effort from developers and client applications. The second way not only allows to execute pre-defined methods, leveraging AIM terms, but also allows users and developers to define their own queries that are converted on the fly to API methods.

The enabler was implemented as a CLI tool and a Web Service.

The former is an ETL tool written in Python implementing Linked Data Pipelines. It takes care of fetching, extracting, preprocessing, transforming, post-processing, and loading linked data into the triplestore. The interaction with the user is performed through CLI (Command Line Interface) and configuration files. Users can choose from a set of specific pipelines, as well as the generic pipeline. Generic pipeline enables single and flexible operations on a different type of data. The tool re-uses other existing tools, such as Geotriples, RMLmapper, and others, for particular tasks, providing a unified interface over them and connecting them transparently in sequences to implement full pipelines.

The project is distributed with a Dockerfile which helps immensely with setting up the whole environment in a stable and reproducible way. It is recommended to use this tool inside the docker container, although if one takes care of all dependencies it is also possible to set it up locally. The docker file and installation instructions, as well as the usage instructions are clearly documented in the README file. The project is distributed using MIT license, and is available via: https://git.man.poznan.pl/stash/projects/DEM/repos/pipelines/browse

The Web Service wraps the CLI tool and exposes its functionalities through a Restful API. The service is deployed in a PaaS environment, and is connected with other containers to provide storage, security, queue management and other functionalities. The architecture of the service is depicted in Figure 10 below.

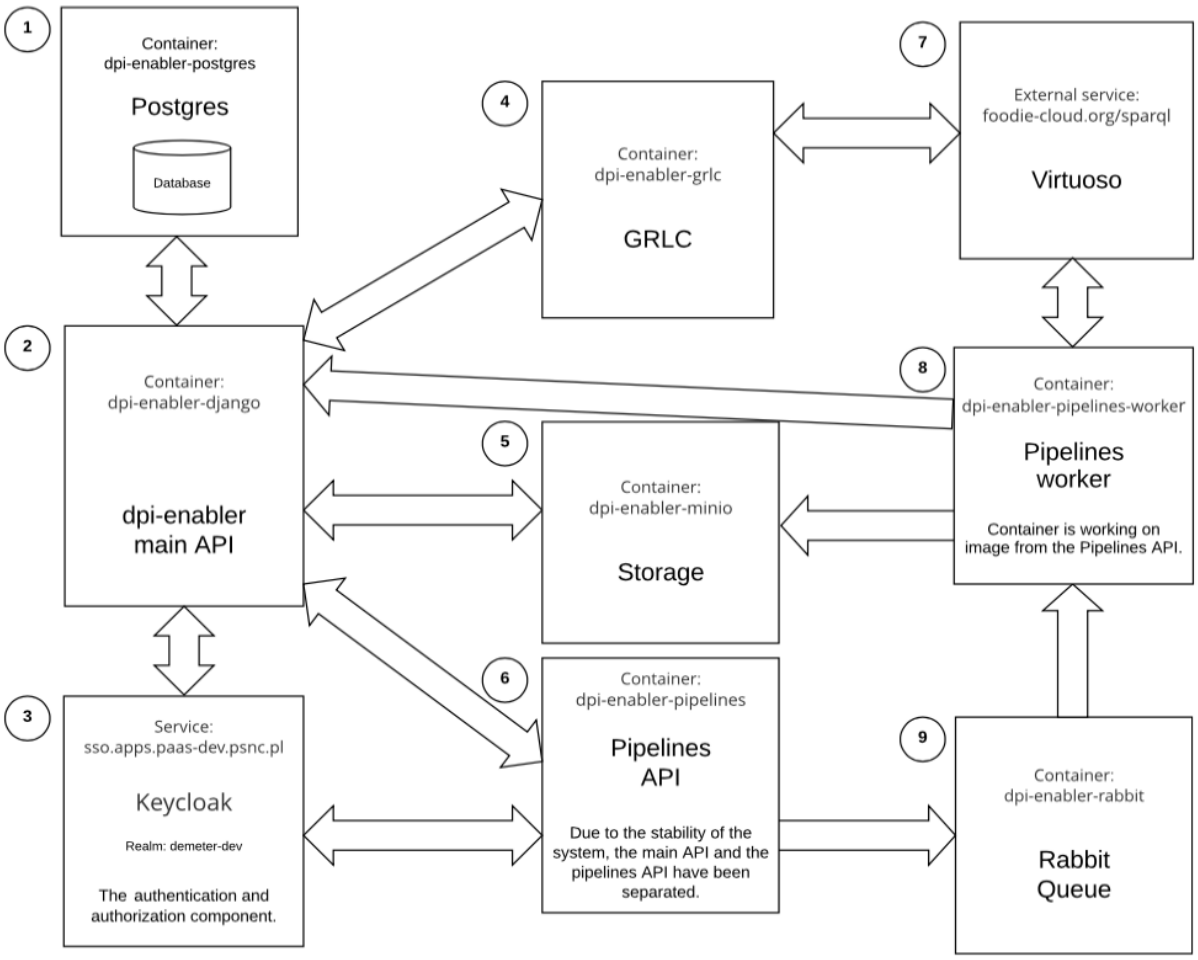

Figure 10: Data Preparation and Integration Enabler Architecture

In the following we provide a short description of each component:

10.2.1 **Postgres**

Postgresql is used to store internal information about the service configuration and instance, such as list of pre-defined queries, users' permissions, logs configuration, etc. The database for the project is called: "postgres", and includes the tables depicted in Figure 11 below.

```
          List of relations
           Name            | Type
---------------------------+-------
api_accessauthorinputset   | table
api_accessinputset         | table
api_accessinputset_queries | table
api_accessqueryurl         | table
api_scheduledjob           | table
auth_group                 | table
auth_group_permissions     | table
auth_permission            | table
auth_user                  | table
auth_user_groups           | table
auth_user_user_permissions | table
django_admin_log           | table
django_content_type        | table
django_migrations          | table
django_session             | table
testapp_note               | table
```

Figure 11: DPI enabler postgresql tables

10.2.2 **Main API for "Data Preparation and Integration"**

This component is based on Django and Django-REST framework, and it is the container exposing the service API. The service endpoint is: https://dpi-enabler-demeter.apps.paas-dev.psnc.pl/, which resolves to a simple API view provided by Django-REST framework, as depicted in Figure: 12 below, which can be used to try the API methods.

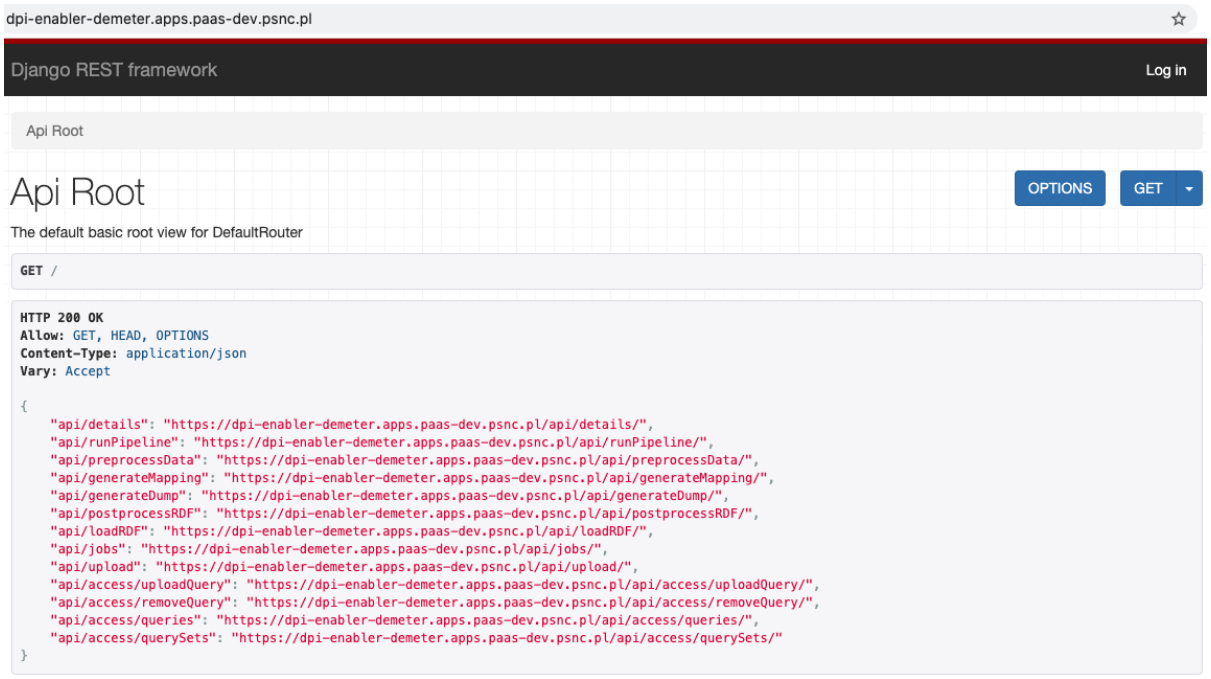

Figure: 12 DPI enabler API default view

The API methods with short description are described below:

1. api/details – returns a basic information about the DPI enabler service
2. api/runPipeline - method to execute a whole pipeline, from the preparation until its loading into triple store.

3. api/preprocessData - method to execute pre-process data tasks. For the generic pipeline, the user has possibility to specify specific tasks, otherwise, pipeline will run pre-configured task.
4. api/generateMapping - method to generate a (RML/R2RML) mapping. In the case of LPIS pipeline, predefined countries can be specified. Additionally, optionally a custom mapping configuration file url can be provided to generate the mapping file.
5. api/generateDump - method to generate RDF dump(s). In the case of LPIS pipeline, a predefined country can be specified and optionally a custom mapping configuration file url.
6. api/postprocessRDF - method to postprocess RDF dump(s). In the case of LPIS pipeline, country should be specified and optionally a custom mapping config file url.
7. api/loadRDF - method to load RDF dump(s) in triple store. For FADN and LPIS pipelines, this method is the same as the method runPipeline.
8. api/jobs - method to shows the status of the tasks.
9. api/upload - method to load a file into the file server.
10. api/access/uploadQuery - method to load a file with query definition into the server to create new method to access data.
11. api/access/removeQuery - method to remove query file from storage.
12. api/access/queries - method to list all the query files from storage.
13. api/access/querySets - method to i) upload a QuerySet file specifying the set of queries accessible via the API, or ii) to list all available QuerySets.

The API methods can also be tried and tested more easily via the swagger interface available via: https://dpi-enabler-demeter.apps.paas-dev.psnc.pl/api/swagger/, as depicted partially in Figure 13.

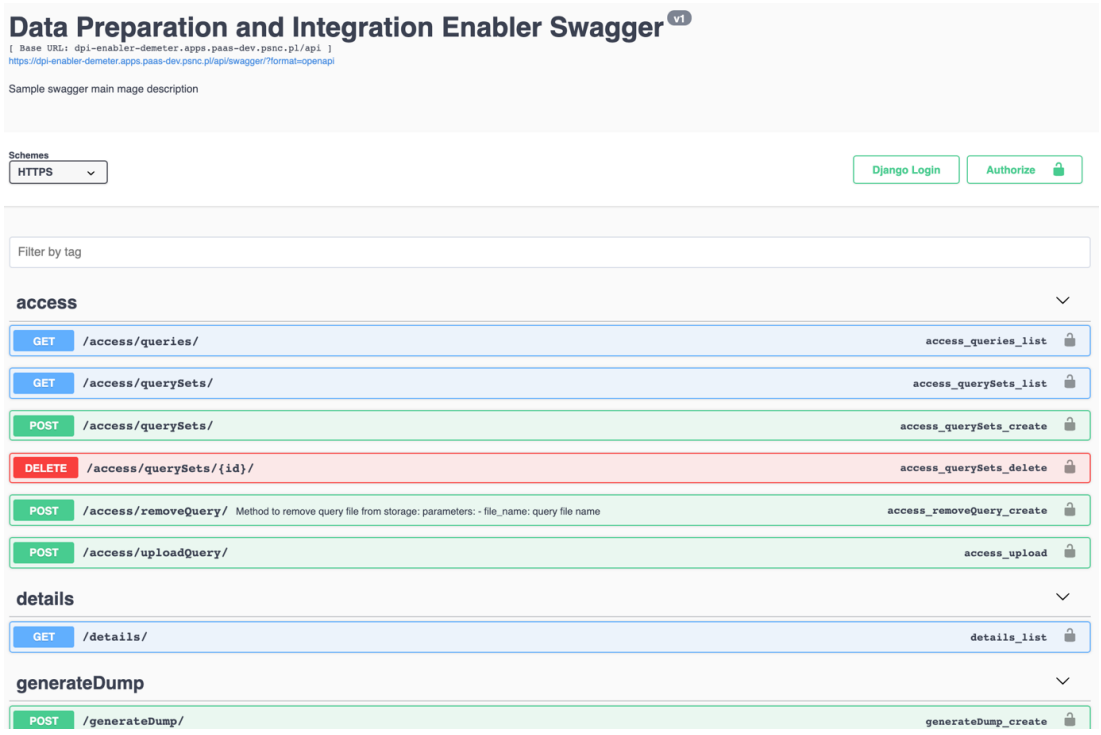

Figure 13: DPI enabler API swagger view

10.2.3 **Keycloak**

The authentication and authorisation component, which enables authorized usage of the API. At the moment, there are some pre-defined accounts to use the DPI service.

The keycloak realm for the project is available via: https://sso.apps.paas-dev.psnc.pl/auth/realms/demeter-dev

10.2.4 **GRLC**

This component enables access to the DEMETER Data & Knowledge repository that stores integrated data via simple Restful APIs methods. The component is based on the GRLC tool[15], which converts SPARQL queries into API methods on the fly. The component is deployed at: https://grlc-dpi-enabler-demeter.apps.paas-dev.psnc.pl/

The component exposes by default an API with a predefined set of queries, accessible via: https://grlc-dpi-enabler-demeter.apps.paas-dev.psnc.pl/api-local/

However, the component is very flexible and allows users to generate the API from their own specification file, specified as a YAML configuration file. This file, known as QuerySet in the DPI enabler, specifies the individual queries that will be exposed in the API.

The individual queries referenced by the QuerySet are uploaded into the DPI enabler via the access method: api/access/uploadQuery. The available queries can be listed via the access method: api/access/queries.

An example query file can be found here: https://dpi-enabler-demeter.apps.paas-dev.psnc.pl/api/access/query/allDatasets.json

Similarly, the QuerySet file is uploaded into the DPI enabler via the access method: api/access/querySets. The available QuerySets can be listed via the same access method: api/access/querySets.

An example query set configuration file is: https://dpi-enabler-demeter.apps.paas-dev.psnc.pl/api/access/grlcInput/urls_5.yml

Then, in order to load the API from the custom QuerySet file, we reference it from the component endpoint, for instance:

https://grlc-dpi-enabler-demeter.apps.paas-dev.psnc.pl/api-url/?specUrl=https%3A%2F%2Fdpi-enabler-demeter.apps.paas-dev.psnc.pl%2Fapi%2Faccess%2FgrlcInput%2Furls_5.yml which loads the following swagger interface as shown in Figure 14:

[15] https://github.com/CLARIAH/grlc

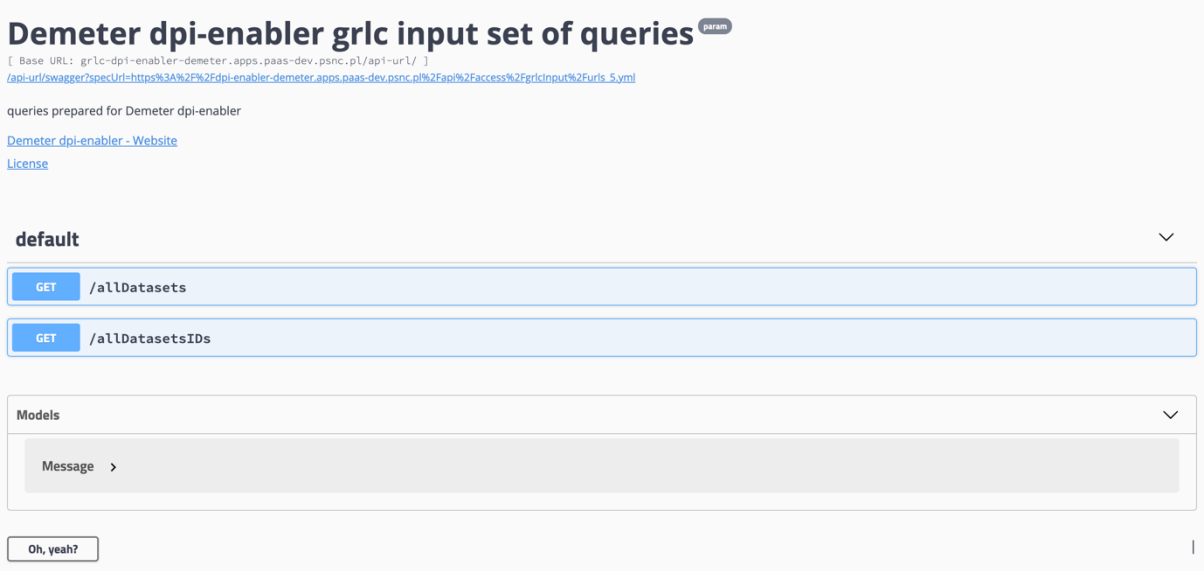

Figure 14: DPI enabler custom access methods example

10.2.5 **Storage**

This component is the main storage for the service. Files stored in this component include, among others: query files and query set files, outcomes of the pipelines methods execution, and logs of the executions. Additionally, users can upload any other type of files via the method: api/upload.

Example of files returned by pipelines execution, e.g. for runPipelines: https://dpi-enabler-demeter.apps.paas-dev.psnc.pl/minio/bca51ef6-72e3-4258-af43-a928dfdfa9bc/

The storage is available via: https://dpi-enabler-demeter.apps.paas-dev.psnc.pl/minio/login

The component is based on https://min.io/. MinIO Storage is a performant and scalable object store for hybrid cloud strategies. S3 compatible from inception, MinIO has more than 7.7M instances running in AWS, Azure and GCP today - more than the rest of the private cloud combined.

10.2.6 **Pipelines API**

This component wraps the CLI tool described above, allowing its execution from the main API component. This component provides a micro-API for the CLI tool to facilitate the execution of the provided functionalities. The container separates pipelines and main API by micro-API based on Flask: https://flask.palletsprojects.com/en/1.1.x/.

The image built from this component is used in the pipeline's worker container where the actual pipelines are executed by celery worker.

10.2.7 **Virtuoso**

Virtuoso is the semantic repository, also known as the data & knowledge repository, where the AIM compliant (integrated) data generated by the pipelines execution is stored by default. The default virtuoso instance used in DEMETER is accessible via http://virtuoso.foodie-cloud.org/, which exposes a SPARQL endpoint and a faceted search GUI. This instance includes other linked data, related or relevant to the agri-food domain, generated by other projects too.

10.2.8 **Pipeline's worker**

The component is based on Celery: https://docs.celeryproject.org/en/stable/.

The worker runs the pipelines API image by celery work.

There are three main processes on this container:

- CLI tool executions
- Submission of the output of pipelines execution to the minio storage
- all updates of job information https://dpi-enabler-demeter.apps.paas-dev.psnc.pl/api/jobs/

10.2.9 **Rabbit Queue**

RabbitMQ is a message broker for pipelines worker.

11 Analytics and knowledge extraction components

11.1 *Data Quality*

Data Quality plays an important role in the data analytics pipeline. Before data is fed into downstream analytics and decision support tasks it has to be ensured that incoming data meets general quality criteria. In this chapter, we summarize the current development and improvements in the Demeter project. More background about approaches how to model data quality can be found in D2.2.

Data Quality Assessment – General View

Data quality (DQ) is an essential characteristic that determines the reliability of decision-making. In this chapter, the Data Quality Assessment (DQA) component is described, realizing data quality analytical functionality to be used within the DEMETER platform. This component analyses *structured* data according to a given measurement specification (*Measurement-Spec*). The Measurement-Spec describes the data quality measures that should be performed by the component. After all measures has been computed, the component returns the measurement results (*DQA-Results*) which can then be used by other DEMETER components to deal with data quality issues that need to be addressed.

From a high-level architectural view, the component has two inputs and one output (see Figure 15):

- **Input**: (1) the data to be analysed and (2) the specification of data quality measures to be used for the analysis (*Measurement-Spec*)
- **Output**: information about measurement results (*Measurement-Results*).

The component is implemented using Python 3 libraries and offers an API to analyse structured data according to implemented measures. The Measurement-Spec is a JSON-String describing the measures to be performed for the data quality analysis. After performing the measure computations, the DQA component provides the data quality analysis results (i.e. the measurement results) using a JSON notation.

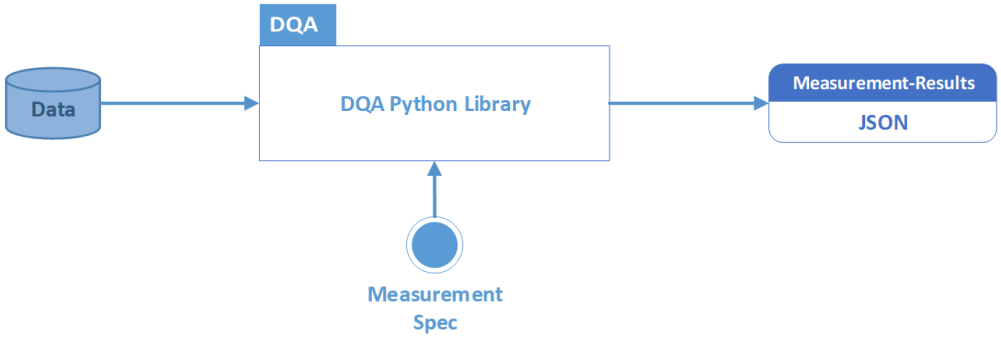

Figure 15: General view of the DQA component.

In the following two sections, we will describe two types of data quality assessment components, namely the DQA component for data quality assessment of structured data (named here as DQA.S) an the DQA component for data quality assessment of linked data (named here as DQA.L). Both need to be integrated in one single DQA service to the DEMETER platform. Figure 16 shows how this can be achieved. Both components, DQA.S as well as DQA.L, uses each a JSON specification (Data-Spec) for

the data source from which to load data, and a JSON specification specifying the measures to be performed on the data (Measurement-Spec).

Both specifications differ in certain aspects (e.g., to load structured data other parameters might be used as when to load linked data). In order to create an integrated version of the DQA (named here as DQA.I), both specification types will be merged and a small routing component (***router***) will take over to delegate the DQA request to the needed component (DQA.S or DQA.L). The details about how that merging will be realized is ongoing work.

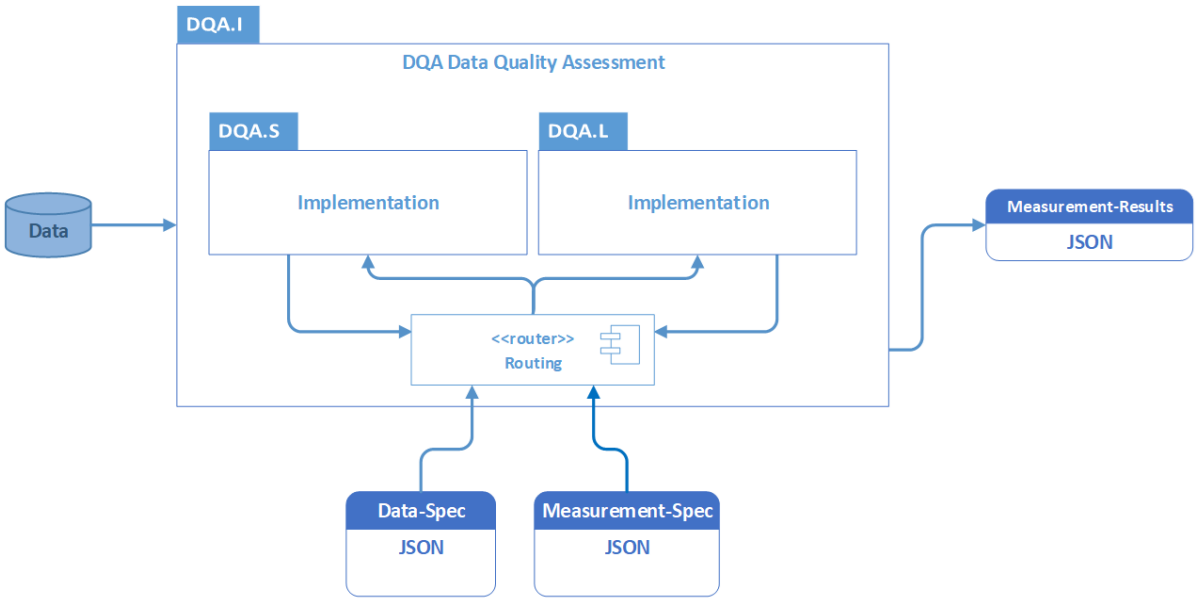

Figure 16: Integrated DQA component.

All implementation of the data quality components described in this section can be found at the WP2 Data Quality folder in DEMETER GitLab: https://gitlab.com/demeterproject/wp2/dataquality

11.1.1 Data Quality Assessment Service of Structured Data

In the context of the Demeter project we deal with machinery data coming from sensors. In order to analyze these data, it must be ensured that the data are of high quality. In the following, data quality is the “degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions” (ISO - ISO/IEC 25012:2008 - Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model 2020).

The ISO-25012 norm distinguishes between *inherent* data quality characteristics and *system dependent* data quality characteristics (note that some characteristics can cover both). After some workshops with engineers from the technical agriculture domain it turned out that the important characteristics in the pilots are the inherent characteristics, namely Accuracy, Credibility, Completeness, Consistency, Currentness, Precision and Understandability. We specified several measurements from the ISO-25024 norm to meet the needs in the pilots. We implement the measurements (sometimes also called metrics) that can be found in Table 1 to evaluate the quality of machinery data. Also, there will be statistical properties provided for numerical values, for example mean, standard deviation, minimum, 25% quantile, median, 75% quantile, maximum. Further information of the procedure how we obtained these measurements can be found in D2.2.

Table 1 Measurements to be implemented for the DQA for tabular data service

Data Quality Characteristic	Measurement title	Formula
Accuracy	Data accuracy range	X=A/B, where A=number of records in a specific range provided by user B=number of data points
Accuracy	Risk of data set inaccuracy	A=the number of anomalies
Completeness	Empty records in a data file	X=1-A/B, where A=number of data points containing nan values B=number of data points
Completeness	Number of time gaps	N=the number of time gaps
Completeness	Length of time gaps	T=Sum of time gaps intervals
Completeness	Metadata completeness	X=A/B, where A=number of attributes with description in metadata B=number of attributes
Consistency	Data values consistency coverage	X= A/B, where A=number of duplicated rows B=number of rows
Consistency	Data values consistency coverage	X=b, where b=1, if the timestamps are sorted b=0, otherwise
Credibility	Values credibility A	X=A/B, where A=number of records in a specific range provided by the user B=number of data points
Credibility	Values credibility B[16]	X=1-A/B, where A=number of geo-points outside the field boundary B=number of geo-points
Precision	Precision of data values	X=A/B, where A=# numerical data points with T ϵ N (or more) expected decimal places B=number of numerical data points
Understand-ability	Master data understandability	X=b, where b=1, if metadata exists b=0, otherwise

[16] This measurement will be ignored in case there is no field boundary in the application, e.g. pilot 2.1.

Internal structure of the DQA Service of Tabular Data Service

Figure 17 shows the internal structure of the DQA component. Its main parts are

- The ***loader*** component responsible to load data from a data source,
- The ***measures*** component realizing the data quality measure functionality and
- The ***analyser*** component responsible to perform measurements by calling measure functions provided by the *measure's* component.

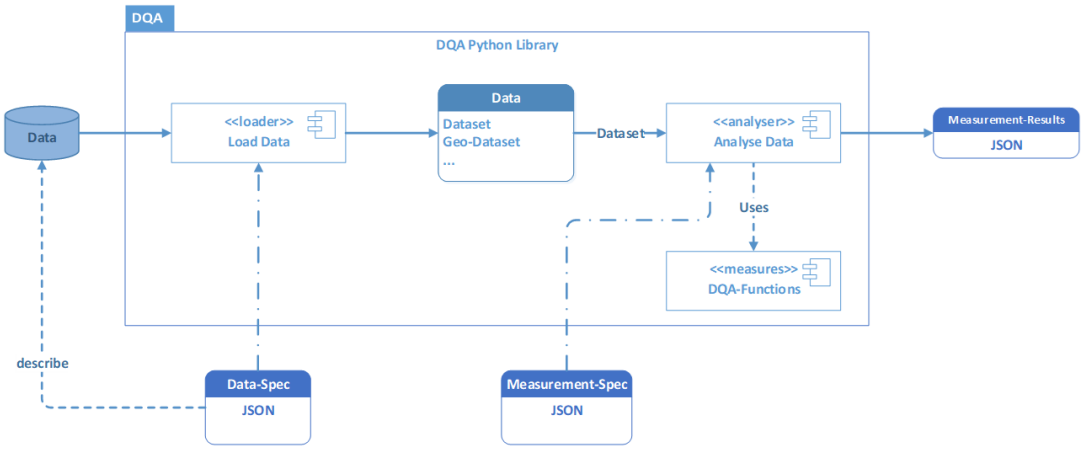

Figure 17: Internal structure of the DQA component for structured data quality analysis.

As mentioned in the previous section, by using the API of the components, the python functionality can be used directly within code. In order to be able to integrate the DQA component into existing software systems or platforms like DEMETER, the API provides methods to use JSON-coded strings to steer data quality measurements. There are two JSON-specifications that can be used to steer the data quality assessment:

- The ***DATA-Spec*** describing the data source from which to load the data that has to be analysed. Besides the URL of the data source other parameters can be specified, like the type of the data source (CSV, Shapefiles, etc.), which column to use for timestamps, whether all columns or only a selection of columns should be loaded, and so on.
- The ***Measurement-Spec*** describing the measures to be used by the DQA component. For each measure, the Measurement-Spec also includes all necessary as well as optional parameters needed to perform the measures.

To be able to make use of the measurement results of the DQA component directly within source code as well as within DEMETER, measurement results are provided using a JSON notation (**Measurement-**

Results). This JSON-String contains information about the measures performed, its parameters, and the findings of the measurements.

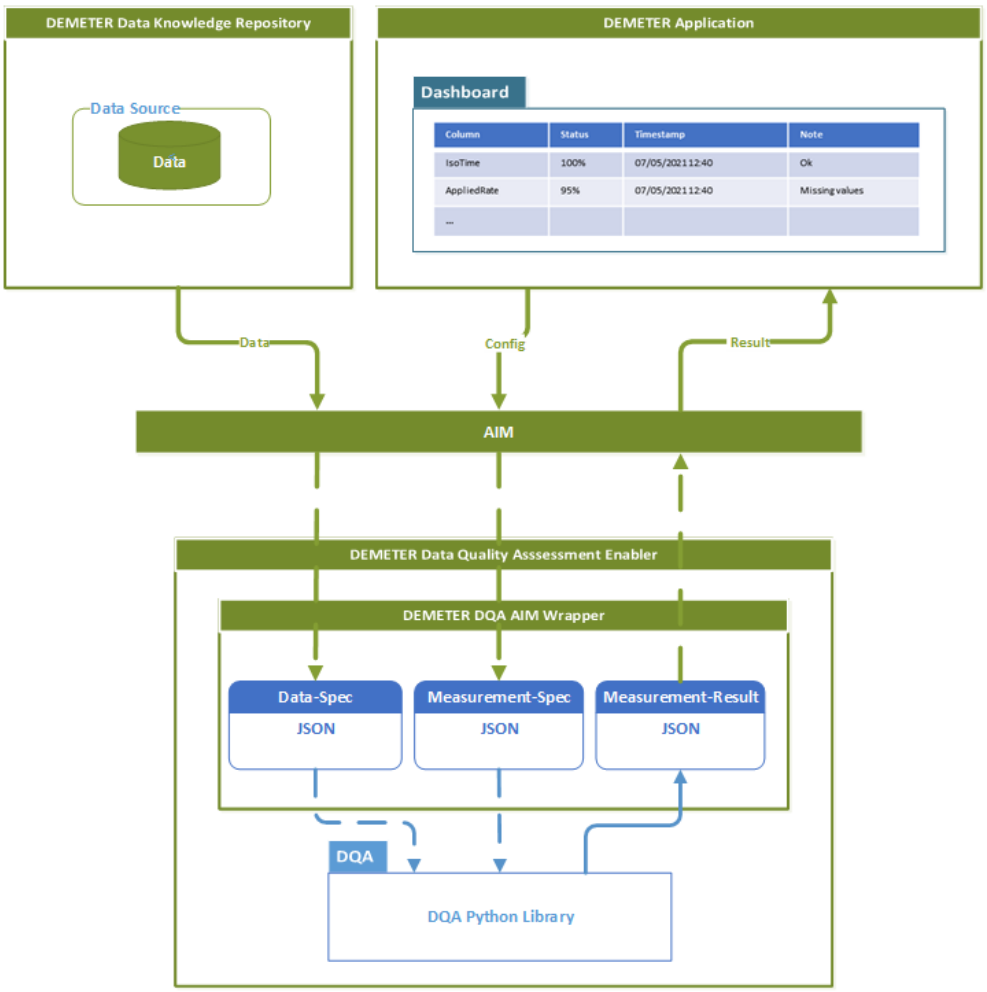

Figure 18: Integration into the DEMETER platform.

Figure 18 shows how the DQA component can be integrated into the DEMETER platform. The DQA component – after its integration into the DEMETER platform as a service (through a DEH) – can provide its functionality to other DEMETER services by making use of the JSON specifications with which the DQA service can be steered. A service like a Dashboard first creates JSON specifications for the data and the measurements (Data-Spec and Measurement-Spec). These specifications are translated to AIM using an AIM-Wrapper and then forwarded to the DEMETER DQA service. The received AIM representations of the two specifications are translated back to JSON and forwarded to the API of the DQA component.

After the component finishes its measurements, the JSON-notation of the measurements results (Measurement-Result) are translated by the AIM-Wrapper to AIM and send to the requesting service (e.g. the Dashboard). This service uses the AIM-Wrapper to translate the received AIM-Message to the JSON-representation describing the measurement results. The service can then react to these results accordingly (e.g., within a Dashboard by visualizing the data quality, or within a data analytics service by triggering needed data preparation algorithms).

11.1.2 Data Quality Assessment Service of Linked Data

The Quality Assessment component for Linked Data is based on the SANSA-Stack[17]. SANSA supports reading most of the common RDF formats, such as RDF/XML and JSON-LD. And it offers a scalable solution for data quality assessment of linked data, suitable for ISO 25012-compliant quality assessment, covering the quality metrics presented in Table 2:

Table 2: SANSA quality criteria

Availability:	**Completeness:**	**Conciseness:**	**Interlinking:**	**Licensing:**
Dereferenceable URIs	Interlinking Completeness	Extensional Conciseness	External SameAs Links	Human Readable License
	Property Completeness			Machine Readable License
	Schema Completeness			
Performance:	**Relevancy:**	**Conciseness:**	**Syntactic Validity:**	**Understandability:**
No Hash Uris	Amount of Triples	Query Param Free URIs	Literal Numeric Range Checker	Labeled Resources
	Coverage Detail	Short URIs	XSD Datatype Compatible Literals	
	Coverage Scope			

Implementation

Figure 19 shows the architecture of the SANSA-based Data Quality assessment component. As this library was created outside of DEMETER, we refer to the paper of Sejdiu et al.[18] The original source code of the SANSA stack is available here: https://github.com/SANSA-Stack/SANSA-RDF.

SANSA is Spark-based and hence a Spark environment is required for running the data quality analysis. We created dockerized Spark environment, based on BDE2020[19] (Big Data Europe 2020). BDE2020 is a general-purpose Spark deployment, funded by an initiative of the European Union[20]. To provide an easy way to execute data quality assessments in this environment, Apache Zeppelin was added to the environment as well. Apache Zeppelin[21] in an open-source computational notebook similar so Jupyter

17 http://sansa-stack.net/

18 Sejdiu, Gezim, et al. "A scalable framework for quality assessment of RDF datasets." International Semantic Web Conference. Springer, Cham, 2019

19 https://hub.docker.com/u/bde2020

20 https://www.big-data-europe.eu/

21 https://zeppelin.apache.org/

Notebook but is targeted towards Spark applications. Docker-compose was used to orchestrate the startup of the BDE Spark environment, the SANSA Stack and Apache Zeppelin.

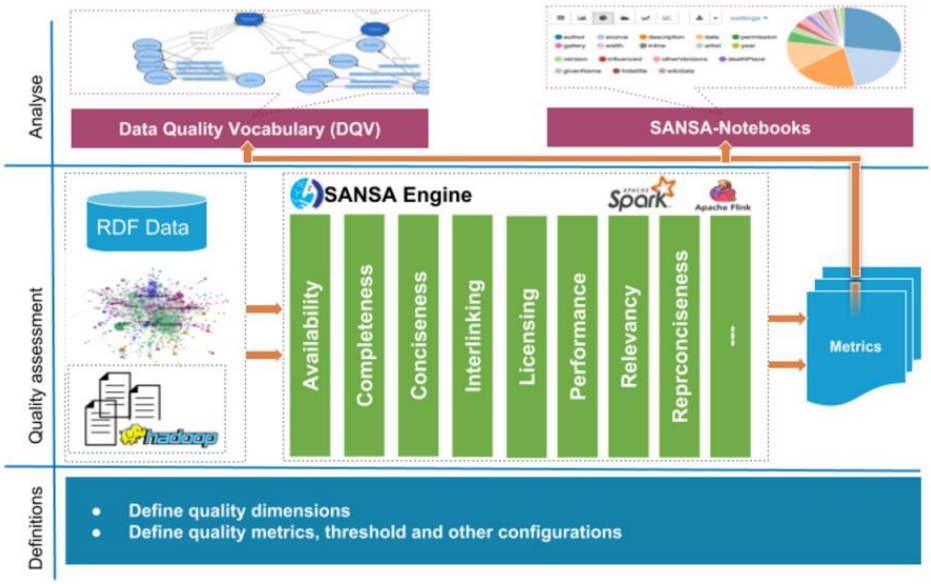

Figure 19: SANSA-based quality assessment[22]

Finally, we provide a Make script, which automatically creates the required Docker network (spark-net) and executes a docker-compose.yaml file, which creates the following processes:

- Spark namenode
- Spark datanode
- Spark master
- Spark worker
- Zeppelin notebook (with SANSA Stack)

The running Zeppelin notebook can then be accessed via *localhost:8080* and be used to execute SANSA-based data quality checks in a self-service fashion.

The described deployment, including Zeppelin example notebooks, is available in the DEMETER GitLab:

https://gitlab.com/demeterproject/wp2/dataquality/data-quality-notebooks/-/tree/master/

An integration of the presented Linked Data Quality Assessment into the Data Quality Assessment Service is still work in progress. In general, the Linked Data Quality Assessment component should follow the same architecture as the Quality Assessment for Tabular Data. Main differentiators to the Data Quality Assessment for Structured Data are the expected data input format (JSON-LD/RDF), the underlying library used for quality assessment (SANSA) as well as the supported data quality metrics. The interface for starting a quality assessment, for defining the quality criteria, and for reporting the quality assessment results are planned to be aligned in the future. The motivation for a unified quality assessment API and, in particular, for a unified way of representing quality assessment results are

[22] http://sansa-stack.net/distqualityassessment/

based on requirements 5.13 (Support for interoperability and interchangeability of data quality assessment) and 5.29 (Easily understandable and machine-executable data quality metrics).

How to Use

We provide an example Zeppelin Notebook, which demonstrates how to use the SANSA Stack for linked data quality analysis. It is part of the deployment described in the previous section and will be available once the make-script was executed, see here: https://gitlab.com/demeterproject/wp2/dataquality/data-quality-notebooks/-/blob/master/notebook/Data%20Quality_2FDUAT44K.zpln

The code for an exemplary data quality validation, in which the schema complements, interlinking completeness, property completeness, and data type completeness are being assessed, looks as follows:

```
import net.sansa_stack.rdf.spark.io._
import net.sansa_stack.rdf.spark.qualityassessment._
import org.apache.jena.riot.Lang
val input = "hdfs://..."

val triples = spark.rdf(Lang.NTRIPLES)(input)

// compute  quality assessment
val completeness_schema = triples.assessSchemaCompleteness()
val completeness_interlinking = triples.assessInterlinkingCompleteness()
val completeness_property = triples.assessPropertyCompleteness()
val syntacticvalidity_XSDDatatypeCompatibleLiterals =
triples.assessXSDDatatypeCompatibleLiterals()
```

11.1.3 Embeddable Data Quality Checks

Embeddable Data Quality Checks is an extension to the Data Quality Service, which is targeted towards Data Analytics and Data Fusion Enablers specifically. It offers a large set of quality metrics for structured data (requirement 5.17), is highly efficient and integrates well with Python-based machine learning pipelines. This allows DEMETER developers who use the AIM-based Model Serving component, to integrate data quality checks into the model serving process. This shall prevent data quality issues within a Targeted Data Analytics/Fusion Enabler effectively (requirement 5.21, 6.15).

The library for embeddable data quality checks (DuckDQ) allows validating tabular data in different data sources, based on user-defined constraints. It is also suitable for equipping scikit-learn pipelines with data quality checks. DuckDQ can check data that passes through the pipeline for compliance with user-defined constraints (e.g. value ranges, column types, completeness) and will stop the pipeline or generate a warning when such constraints are not met. It can be easily integrated into any existing scikit-learn training or serving pipelines (see Figure 20).

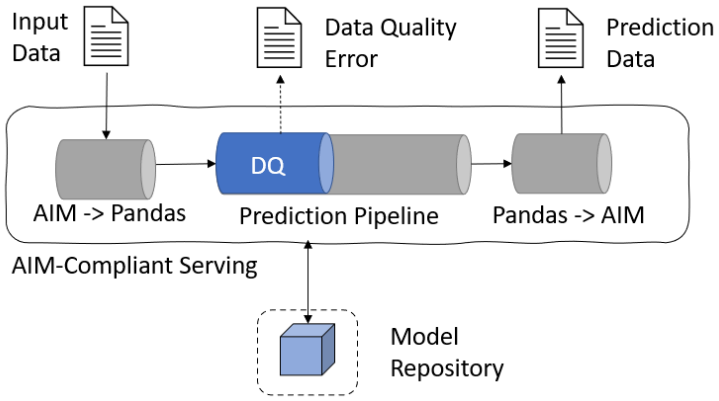

Figure 20: Embeddable Data Quality Checks (DQ) in Context of the AIM-Compliant Serving Enabler

Implementation

The library was implemented in **Python 3.7** and uses **DuckDB**[23] as computational backend. It offers a data validation API similar to deequ[24] and additionally a tight integration with the **scikit-learn** machine learning library. The architecture is shown in Figure 21. The Verification Suite handle the business logic of the data quality validation. It takes incoming quality check definitions from either the general-purpose verification API (s. subsection *Verification API*) or from the scikit-learn pipeline integration (s. subsection *Scikit-Learn Pipeline Checks*) and translates them to an execution plan. DuckDQ offers an execution engine for Pandas DataFrames and SQL Databases, supported by SQLAlchemy[25].

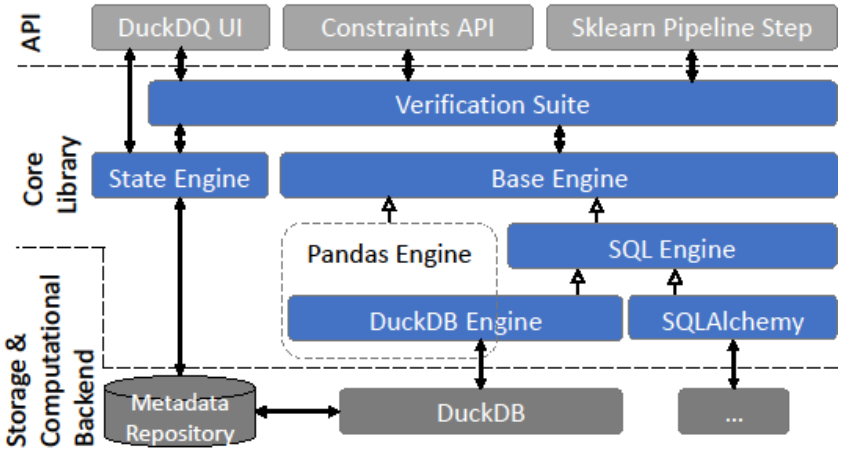

Figure 21: Embeddable Data Quality Checks (DQ) Architecture

Further implementation details and code documentation are available on GitHub: https://github.com/tdoehmen/duckdq

DuckDQ can be installed locally, using the following commands:

[23] https://duckdb.org/
[24] https://github.com/awslabs/deequ
[25] https://www.sqlalchemy.org/

```
git clone https://github.com/tdoehmen/duckdq
cd duckdq
python setup.py install
```

The DEMETER GitLab provides further usage examples and installation instructions.

Verification API Example:

```
import pandas as pd
from duckdq.checks import Check, CheckLevel
from duckdq.verification_suite import VerificationSuite

df = pd.read_csv("data/train.csv")

verification_result = (
    VerificationSuite()
        .on_data(df, dataset_id="data10")
        .add_check(
        Check(CheckLevel.EXCEPTION, "Basic Check")
            .is_complete("Name")
            .is_contained_in("Pclass",(1,2,3))
            .is_contained_in("Sex",("male","female"))
            .is_contained_in("SibSp",[1, 0, 3, 4, 2, 5, 8])
            .is_contained_in("Embarked",("S","C","Q"))
            .has_min("Age", lambda mn: mn > 0)
            .has_max("Age", lambda mx: mx < 60)
            .has_min("Fare", lambda mn: mn >= 0)
            .has_max("Fare", lambda mx: mx < 999)
    )
        .run()
)
print(verification_result)
```

In total, 42 different quality checks are available. The following lists contains the most important checks, ordered by data type:

Numeric Values	String Values	Both
hasMin	hasPattern	isComplete
hasMax	containsEmail	isUnique
hasMean	containsURL	hasNumberOfDistinctValues
hasSum	hasMinLength	hasSize
hasApproxQuantile	hasMaxLength	hasDataType
hasStandardDeviation	satisfies	isContainedIn
hasCorrelation		
isNonNegative		
isPositive		
isLessThan		
isLessThanOrEqualTo		

Scikit-learn Pipeline Checks Example:

```
from duckdq.sklearn import DataAssertion, Check, CheckLevel

dataquality = DataAssertion(
    Check(CheckLevel.WARNING, description="Basic Check")
        .is_complete("Name")
        .is_contained_in("Pclass",(1,2,3))
        .is_contained_in("Sex",("male","female"))
        .is_contained_in("SibSp",[1, 0, 3, 4, 2, 5, 8])
        .is_contained_in("Embarked",("S","C","Q"))
        .has_min("Age", lambda mn: mn > 0)
        .has_max("Age", lambda mx: mx < 60)
        .has_min("Fare", lambda mn: mn >= 0)
        .has_max("Fare", lambda mx: mx < 999)
        .is_unique("PassengerId")
        .is_unique("Name")
        )
from sklearn.linear_model import LogisticRegression

pipe = Pipeline([('dataquality', dataquality),
                 ('preprocessor', preprocessor),
                 ('clf', LogisticRegression(solver='liblinear')),
                 ])

pipe.fit(X_train, y_train)
```

11.2 *Targeted Data Analytics and data fusion*

11.2.1 Pattern Extraction for Optimal Fertilizer Usage

We have developed a module that provides insights on the Nitrogen status in rice and maize crops, based on indices extracted by processing UAV or satellite imagery. Specifically, imagery is logged properly and several spectral indices are being computed in the process. The data is annotated as Vegetation Indices, metrics sensitive to plant biomass and chlorophyll. These indices need to be produced so that we can depict reflectance and other similar traits of the plant. All data are fed to the model in the form of a comma-separated file containing all the aforementioned indices. The model deploys machine learning algorithms to estimate the nitrogen level in the crop. Output of the component is a list of the nitrogen level in each plot included in the analysis. Specifically, there are three levels of nitrogen status, those being a) “urgent action needed”, b) “needs fertilization soon” and c) “no fertilization needed”. A simple visualization of the component’s output is presented in Figure 22 below.

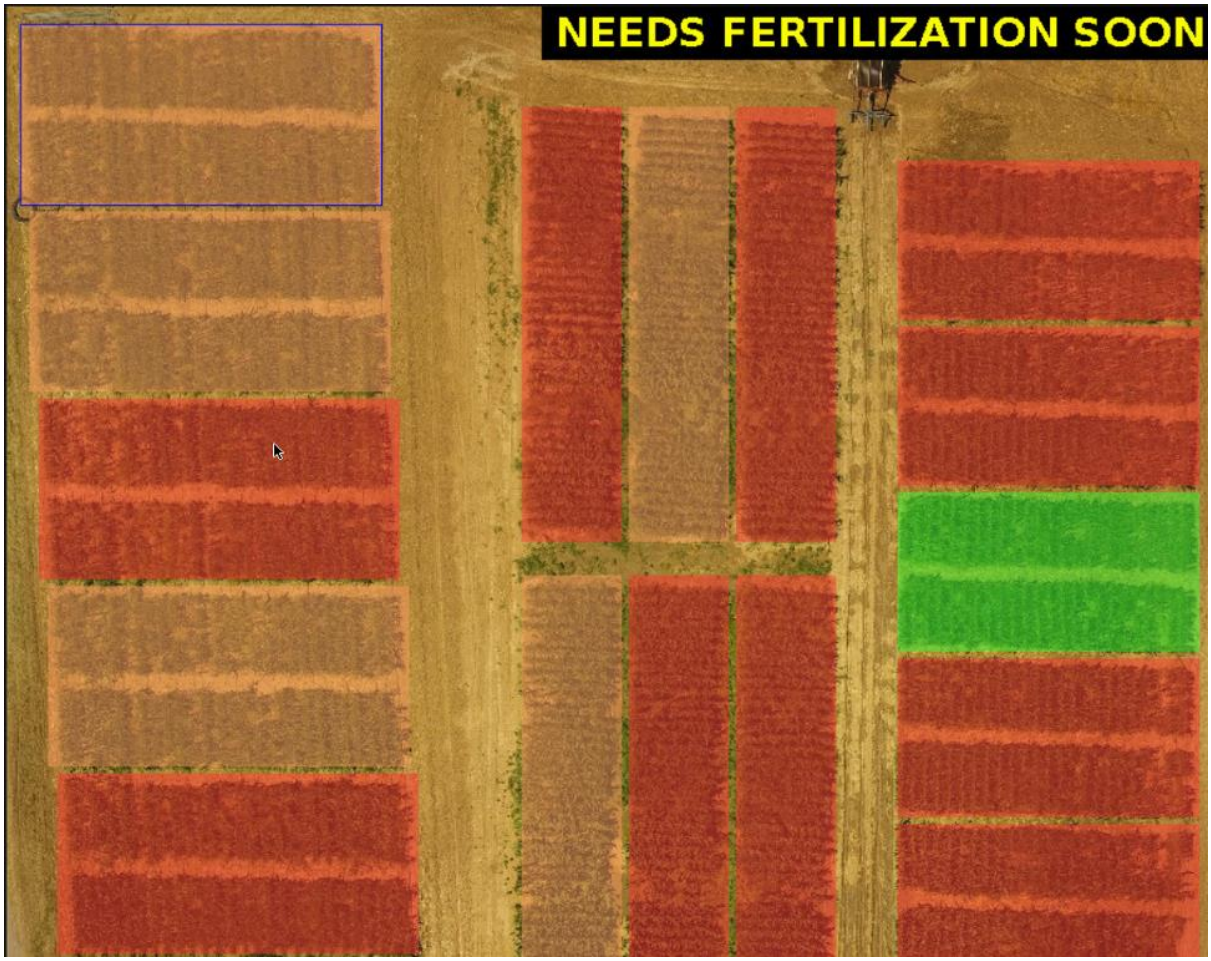

Figure 22: Visualization of Optimal Fertilizer Usage enabler

The motivation behind this module is to optimize the fertiliser usage. Nitrogen is by far the most used ingredient as fertilizer and actually is the only one used before sowing. This stage is the one that currently depends solely on human experience and requires decision support. Advanced analytics can assist this task by estimating the level of nitrogen and consulting the farmer on whether an intervention should take place. This module is destined to be a component of a broader decision support system that consults farmers about the appropriate fertiliser usage and maybe also on better scheduling of the aforementioned treatment. Efficient fertiliser use is one of the biggest challenges that farmers face and this decision support system is more than necessary to facilitate in-farm operations.

The module deploys a randomized technique to choose the few most appropriate among all indices and then feed a model that estimates the aforementioned Nitrogen quantities. For now, two algorithms run independently to provide the desired output, those being regression and random forests. The first one estimates the desired output as a number, while the latter quantizes the output into same range classes. The model is evaluated with cross-validation algorithm, as there is currently a shortage in ground truth data and the model is not adequately trained following the standard procedure of dividing the dataset into train (and validation) and test slices.

An idea that is being recently implemented is to divide the range of the output in a number of classes and transform the problem to a classification instance that allows lower mean error and subsequently

higher accuracy. This last part is ongoing work. The first model concerns the initial approach and is actually deprecated.

The current model is based on random forest ML algorithm and subsequently, solves a classification problem. The component is implemented in Python programming language using the pandas and scikit learn libraries. As mentioned earlier we use random forest algorithm to classify the plots into one of the three aforementioned levels. Fine-tuning is used to select a few of all the indices that we use as input and data argumentation tackles the problem of few ground truth data that follows the model through all those steps.

As described, the component is meant to be integrated with other components to develop a decision support system that aims to facilitate decisions about in-farm operations. Thus, details about visualisation and how the module is going to be used are not yet determined.

11.2.2 **Data Analysis for Irrigation**

We have built an application in the python programming language and using many libraries such as tensorflow, PyQt5 and others, that displays a UAV image that the user has chosen, and also gets informed about ground sensors' measurements (wetness, temperature, relative humidity) for that specific image. It has also the capability of supporting the farmer take a decision on when to irrigate maize based on weather forecasting. Weather forecasting is been implemented by the APP by making a good use of wetness, temperature, relative humidity historical data. So, the maize irrigation APP is consisted of the following functional buttons:

- Load UAV image button, where the user is navigated inside folders and choose the UAV .tif image he wants to see (1).
- Load time durations excel file, where the user can choose the excel file that contains all the UAV .tif images and the related timestamps that show when the image capturing started and when the capturing ended (2).
- Load sensors' measurements excel file, where the user can choose the excel file that contains ground sensors' measurements (wetness, temperature, relative humidity) and the related timestamps (3).
- Home button, that the user chooses in order to navigated to the initial state (4).
- Left arrow, that the user chooses in order to navigate to the previous screen state (5).
- Right arrow, that the user chooses if he wants to navigate to the next screen state (6).
- Cross button, for moving Earth, North, South, West inside the .tif image (7).
- Zoom button, to zoom in/out in the UAV image (8).
- Save button, to store the current state of the UAV image, in other words what the user examines/views at that specific time (9).

On the figure we depicted also the .tif image viewing panel (10), the information box (15) that shows all the information of the sensors' measurements (14).

There are also 3 more buttons related to weather forecasting. "Predict Wetness" (11), "Predict temperature" (12) and "Predict relative humidity" (13). The rationale behind the last 3 buttons is to make forecasting in wetness, temperature and relative humidity and help the farmer decide whether to delay a scheduled irrigation, if there is a forecasting for rain in the near future, or hasten an irrigation if there is prediction for warm weather during the next days. An illustration can be found in Figure 23.

Figure 23: Maize irrigation APP overview

Below we depict the graphs for the prediction in wetness, temperature, relative humidity. The dataset contains about 15000+ values and we split the dataset to 67% used for training and 33% for testing. We use RNN-LSTM neural network and 50 epochs for training. When running the code for predicting **wetness**, we trained our RNN-LSTM model with 50 epochs, with train Score: 9.85 RMSE, test Score: 8,94 RMSE. When running the code for predicting **temperature**, we trained our model with 50 epochs, with train Score: 0,81 RMSE, test Score: 0,93 RMSE. When running the code for predicting **relative humidity**, we trained our model with 50 epochs, with train Score: 2,93 RMSE, test Score: 2,60 RMSE. In blue colour we depict the original dataset, the predictions are coloured green, and the known data with orange. It is obvious that there is high accuracy and small error. An example is illustrated in Figure 24.

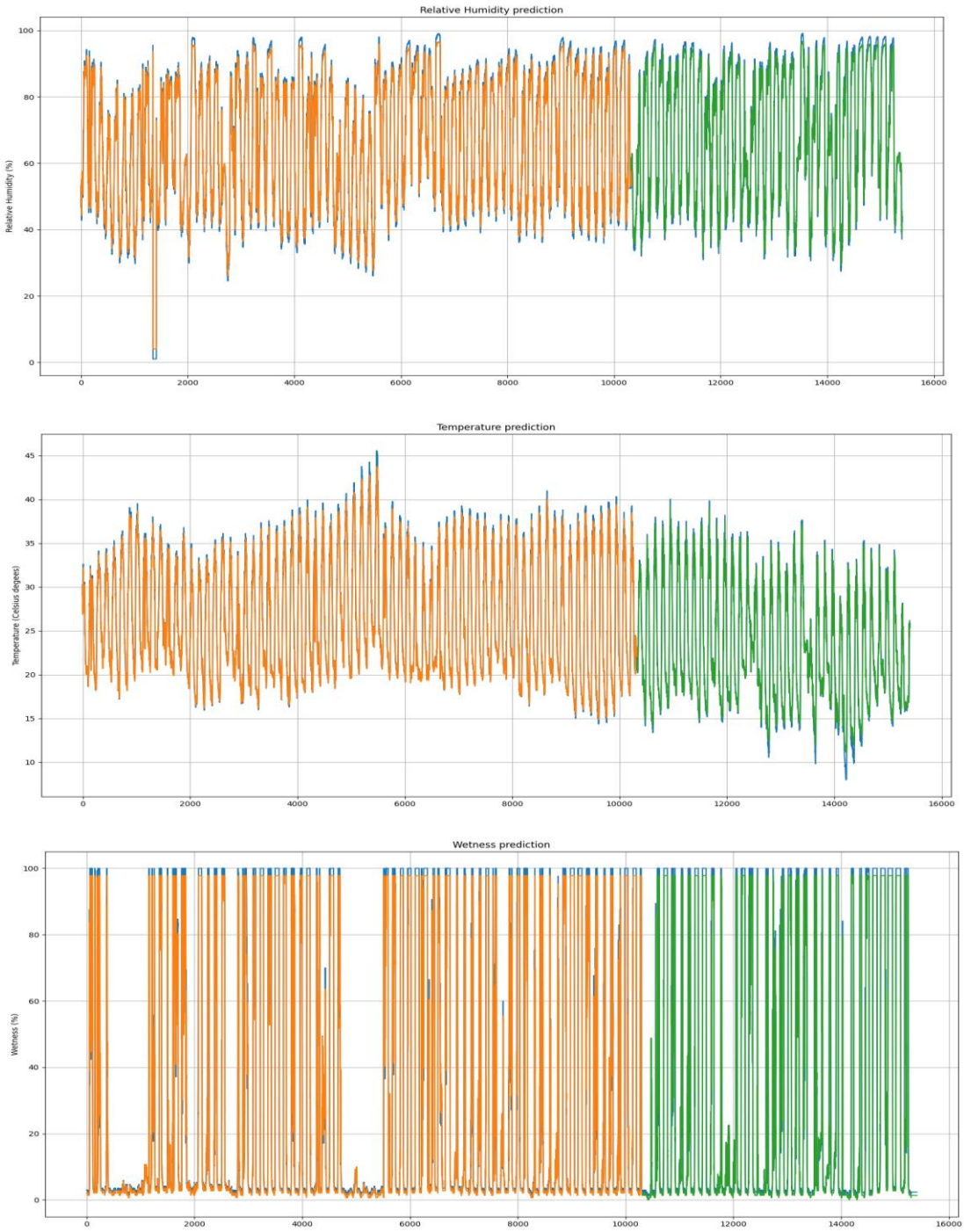

Figure 24: Prediction of Relative Humidity, Temperature and Wetness

11.2.3 **Predictive Model Training web service**

The component is a web service that allows producing an instance of a fully trained and validated predictive model by machine learning methods, carrying out multiple steps in data analytics on input testing/training/validation data made available from e.g. a database.

The development of the component has resulted in a library package coded in the Python programming language. The package can be used to instantiate a web-service that exposes a web-

based graphical user interface dashboard that allows carrying out the steps required for the model definition and training. Once the model is trained and quantitatively validated, it can be extracted from the filesystem managed by this web service in a PKL format, which is a format created by the Python module pickle. Then it and integrated in a prediction service providing an API (e.g. based on exchanging information and data in an AIM format), as done for instance for the Olive Phenology Prediction Service of WP4.

The three functions made available to users are 1. the selection of relevant input and output / target features, 2. the analysis of the selected features and the training of prediction models, and 3. the comparison of different trained models in terms of their quantitative performance / relative error on a validation data set. An illustration can be found in Figure 25.

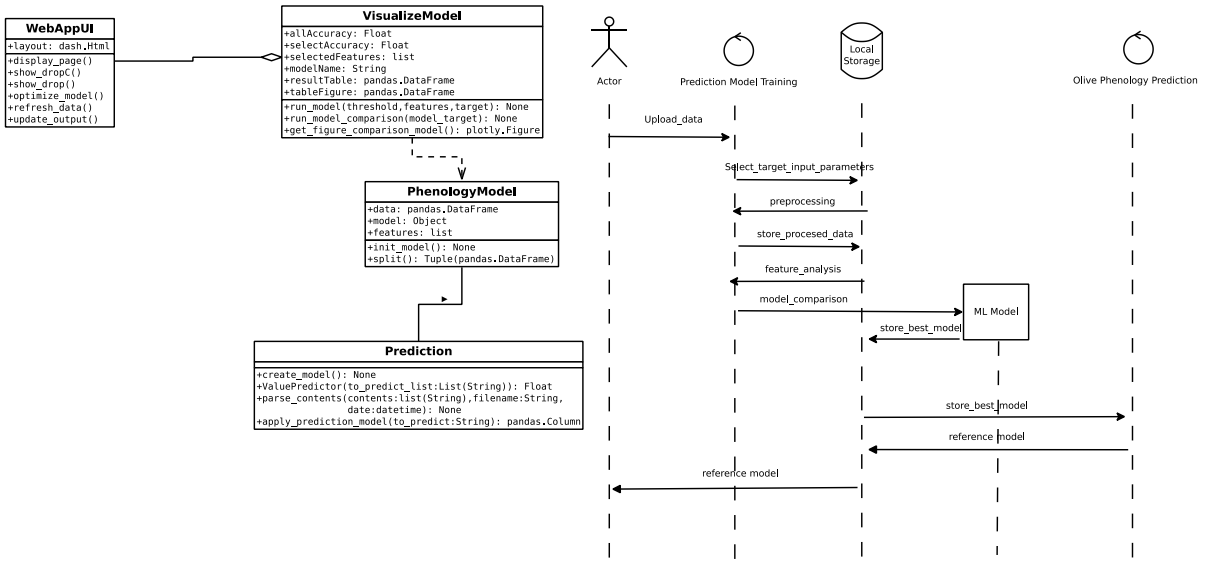

Figure 25: Illustration of Predictive Model Training web service

The final objective of the component is making available to the DEMETER project a pre-integrated and validated technological basis for the development and exploitation inside project Pilots of efficient and quantitatively validated predictive models that can then be exposed via additional web services that can be registered and made available for exploitation by project partners.

11.2.4 **Pattern Extraction with Computer Vision**

The component described here (depicted in Figure 26) allows the creation and use of computer-vision-based models for element counting. This component has been developed using the python programmatic language and some libraries such as flask and flask_restplus for the API creation or imageAI and tensorflow for the model generation based on the YOLOv3 model. The functionality provided is shown as an API that lists the operations to create and use the models created. The component has been developed as a standalone component and is currently being integrated with MLFLOW, ACS, BSE and DEH, so some changes might be expected.

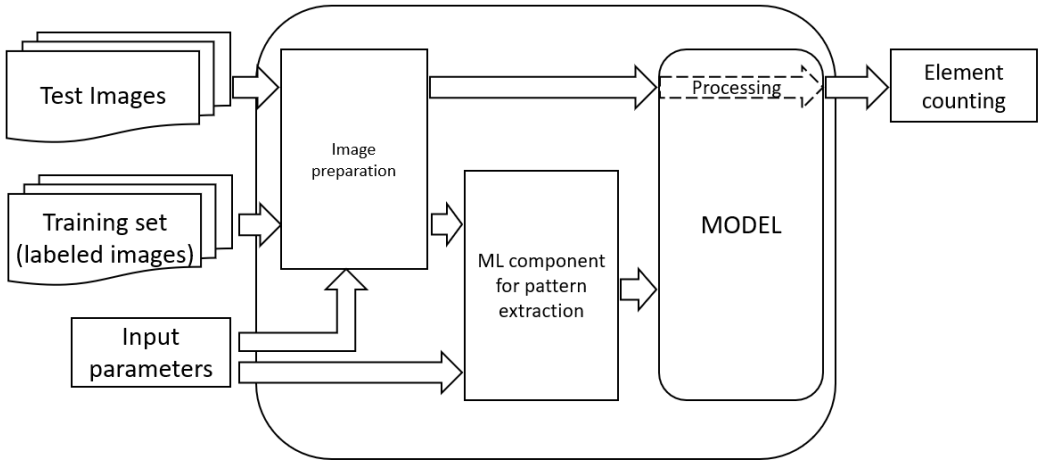

Figure 26: Computer vision knowledge extraction task

This component shows an API that allows the execution of some operations. These operations include the creation of a computer vision-based model for element counting, starting from a set of labelled images (using the Pascal VOC format). The quality of this training dataset is a key point in the accuracy of the outcomes of this component, where the more labelled images used in the model generation process, the better for a proper object identification. Once the model is generated, the component returns a JSON in AIM format where the model ID can be found in order to be used later on in the "countElements" function. Figure 27 depicts the sequence diagram of the model creation process.

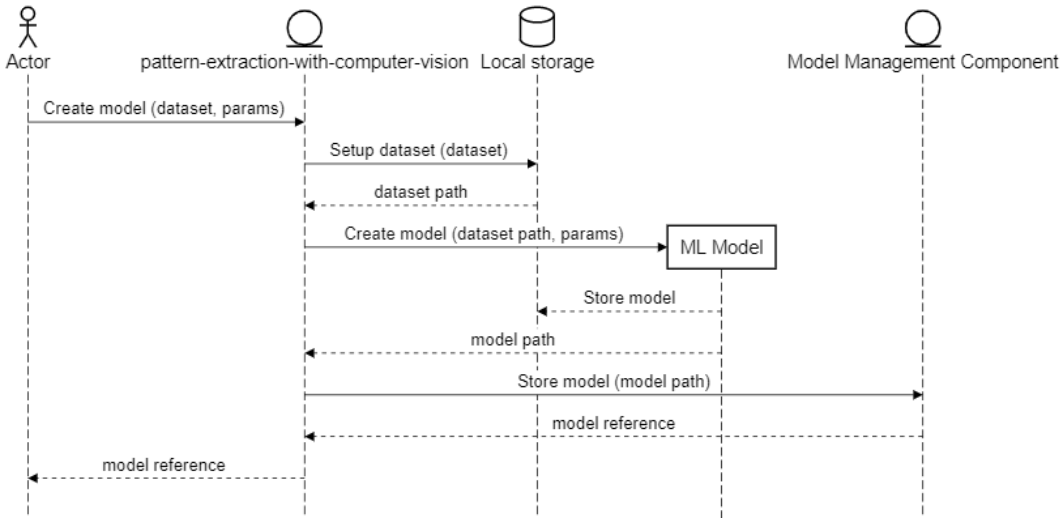

Figure 27: Sequence diagram for the model creation in the Pattern Extraction with Computer Vision component.

In order to perform the element counting (depicted in Figure 28), it is required an image file as well as a model identifier. Using that model identifier, the model is retrieved from the MLFLOW framework and used to detect the elements in the given image. The result of this call provides the areas detected as elements by the model, each one defined by the coordinates of two opposite corners of a square bounding the element as well as the label given to that element and the probability of that area to represent the proposed element.

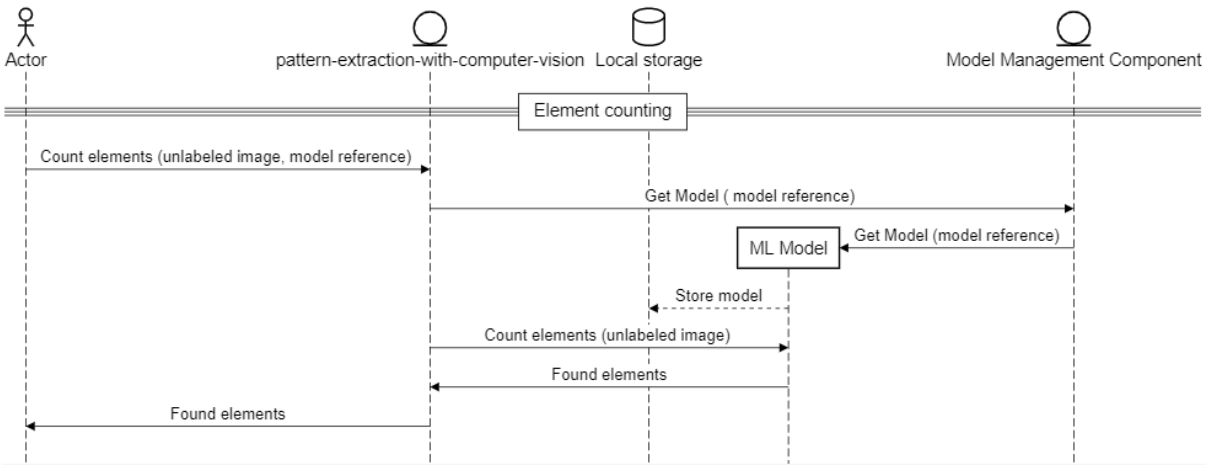

Figure 28: Sequence diagram for the element counting in the Pattern Extraction with Computer Vision component.

11.2.5 Data Analysis for Optimal Pesticide Usage

The data analytics for optimal pesticide usage is developed for four fungal diseases, as a digital model, wrapped with an API and provided as a docker container image. The component takes environmental parameters into account to quantify if conditions for a disease are met and returns recommendations on what operations to undertake on the farm to prevent the disease or to reduce its impact.

Disease prediction models for fungal diseases

Vineyard fungal diseases models are widely known in the agriculture. The development of sensors and IoT technology enable digitalization of these models providing timely remediation, minimizing farmers in-field effort. All models provided in this study quantify the Risk Index (RI) used to identify the need for remediation measures, that drives the decision support models provided to farmers.

In this study, the research was based on the fact that all disease models are heavily influenced by environmental conditions such as temperature and humidity. The risk forecast models are created for four fungal diseases: 1) Downy mildew, 2) Uncinula necator (powdery mildew), 3) Guignardia bidwellii (Black rot), 4) Botrytis cinerea (grey mould) disease; and pest 5) Lobesia botrana (grapevine moth).

Dataset and integration in pilot

For the precision viticulture pilot 5.1, the data was collected at the vineyard operated by the company ”13. jul Plantaže” located in the municipality of Podgorica in Montenegro (Figure 29). The company operates a huge vineyard in a single complex, covering area of over 2300 ha. The company is one of the largest wine producers in the South Eastern Europe, producing around 22 million kilos of grapes and more than 16 million bottled products annually. Figure 29 shows a map (from agroNET, a digital farm platform used in the pilot) displaying all sensor node locations in the vineyard.

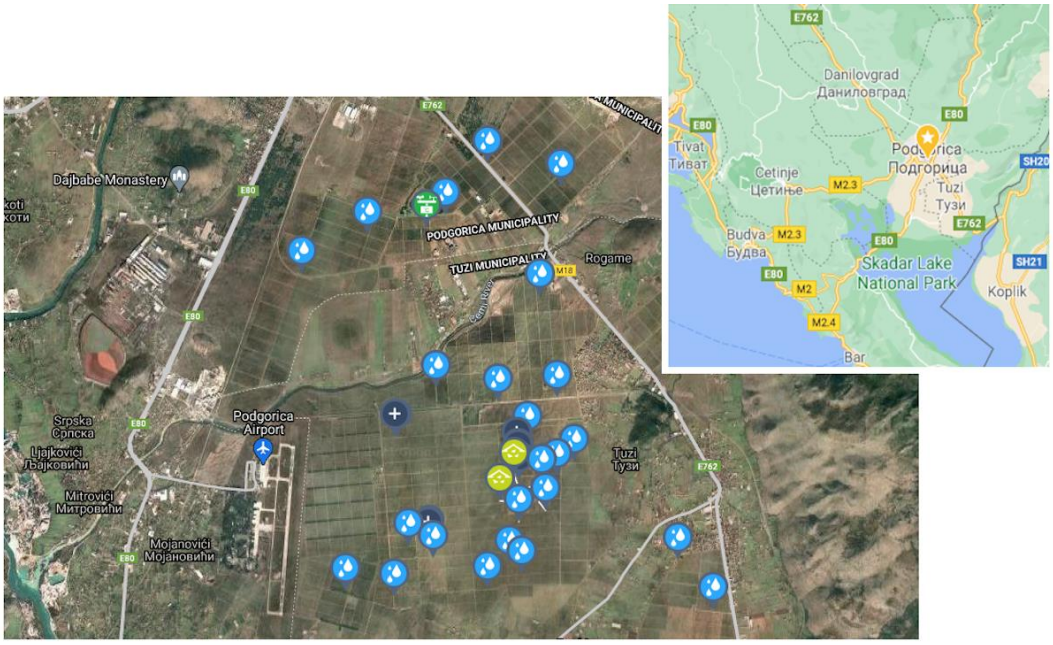

Figure 29 Map showing the pilot site and agroNET sensor node locations near Podgorica, Montenegro

The area of 50 ha is organized in tables (labelled "Table 4, 5, 6, 7, and 8" in Table 3), as shown in Fig. 2. The equipment installed includes a weather station for monitoring microclimate at the site (lot labelled as Table 6). The weather station is equipped with sensors for air temperature, air humidity, precipitation, leaf wetness, global and active solar radiation, and wind speed. Each lot, labelled Table 4 to 8 is equipped with sensor nodes for monitoring soil moisture at various depth, which can be used for irrigation optimization. There is a total of 6 soil moisture nodes, two equipped with Sentek sensors for monitoring soil moisture, soil temperature and salinity at six different depths, and four nodes equipped with ECH_2O or Teros 10 sensors for monitoring soil moisture at two different depths (see Figure 30).

Table below provides more information on the pilot site and installed sensors. The weather station is located in the middle of the pilot site between "Location Tables 5" and "Location Table 6" (see Table 3).

Table 3: Locations and number of the sensor's node

Location	*Grape variety*	*Sensor node location*	*Sensors*
Table 5	Vranac clone 4	IRIG4 42.367900, 19.293611	Soil moisture, Sentek, six depths 0-60cm
Table 6	Vranac clone 5	METEO 42.366227, 19.293732 IRIG6 42.367866, 19.293566	Weather Station Soil Moisture, Terros
Table 8 (100th row)	Vranac clone 5	IRIG2 42.363072, 19.294091	Soil moisture, ECH2O, two depths 30 and 60 cm
Table 4	Vranac clone 2	IRIG5 42.369426, 19.293347	Soil moisture, Sentek, six depths 0-60cm
Table 7	Vranac clone 5	IRIG3 42.364669, 19.293917	Soil moisture, ECH2O, two depths 30 and 60 cm,
Table 8 (200th row)	Vranac clone 7	IRIG1 42.363175, 19.297267	Soil moisture, ECH2O, two depths 30 and 60 cm

Figure 30 Pictures of the sensor nodes installed at the pilot site, from left to right: a) weather station; b) digital pheromone insect trap with camera, c) soil moisture node; d) three types of soil moisture sensors.

Application Programming Interfaces

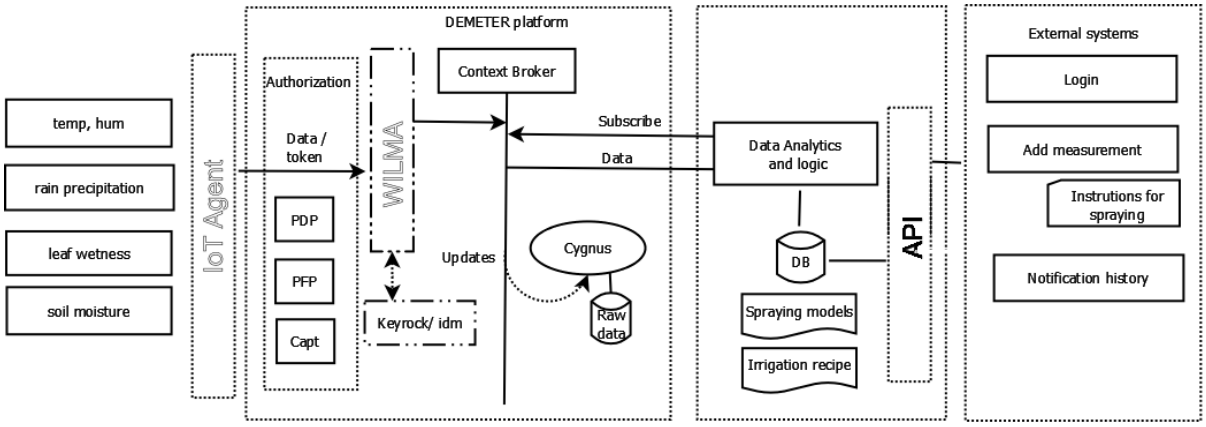

Figure 31 Data Analysis for Pesticide usage workflow.

The APIs are composed to enable authorisation process to obtain token used to POST the data to the "addmeasurement" endpoint (see also Table 4). Data analytics for optimal pesticide usage model is expecting a specific set of data, namely temperature, humidity, precipitation and leaf wetness are required to be able to quantify the risk index and create instructions for farmers. If some of the entries are empty, model at least needs temperature or humidity to be able to provide instructions. To be able to calculate risk more efficiently, it is required to have more data values, as the accuracy of the risk will be higher. The process is illustrated in Figure 31. As a result, the endpoint returns instruction for the farmer (see Figure 32).


```
{
  "items": [
    {
      "id": 394500,
      "message": "Taphrina deformans - There is a high risk of leaf infection. If the variety is still in the green point\" phase (broken buds, and green leaves are barely visible), and treatments have not been done in the last 2 days, treatments with contact preparations should be done today. If the treatment fails to be performed today, no later than 48 hours from the beginning of the precipitation, treatment with locally-systemic preparations should be done.  In plantations or varieties where the buds are open for more than three days no further treatments should be made.",
      "date": "2021-02-01T02:51:07.567",
      "dateText": "2/1/2021 3:51 AM",
      "macIdentifier": "pessl_01208AB2",
      "serviceName": "alertNET"
    }
  ]
}
```


Figure 32. Example of message with instruction based on a received environmental data

Table 4 Locations and number of the sensor's node

API name	Method/ Parameters	Description
/Login	Post/ username password	Returns auth token
/agroNET/addmeasurement	Post/ "date" "identifier" "temperature" "humidity" "precipitation" "leafWetness" "token"	Post environmental parameter values to get precipitation instructions
/notificationhistory	POST/ MAC identifier Start date End data Token	List of notifications with instructions

11.2.6 **Weather Forecast**

This component has been developed using R programming language. It uses as input latitude and longitude of a plot (a location code for AEMET web service) to retrieve weather forecast data (i.e. air temperature, air relative humidity, wind speed, rainwater, solar radiation, etc.) from external weather services to DEMETER (i.e. Openweather, Weatherbit, AEMET, etc.) to return it using an AIM data model.

The functionality of this component is used by DSS water balance model (WP4 - 4.B.3 Irrigation Requirements Estimation) to retrieve rainwater forecast for a plot. Also, the predictiveET0 DEE component (WP4, 4.B.3 Irrigation Requirements Estimation) is using this functionality too but as an internal R library to retrieve weather forecast information for a plot where the reference evapotranspiration (ETo) is going to be predicted. The procedure is illustrated in Figure 33.

The information from the external weather services is retrieved as plain text raw data, having one register for all values observed in a same date as shown in next example:

```
N date                 temp  HR rain windspeed radiation
1 2021-02-03 10:00:00 12.85 87 NA   1.89      NA
2 2021-02-03 11:00:00 13.91 66 NA   1.60      NA
3 2021-02-03 12:00:00 15.73 49 NA   1.54      NA
4 2021-02-03 13:00:00 17.24 40 NA   1.49      NA
5 2021-02-03 14:00:00 18.18 36 NA   1.61      NA
```

This information is then returned using a wrapper into an AIM data model with the terms, properties and structure shown in next example (used only for the air temperature value):

```
{
   "@context": [
      "https://w3id.org/demeter/agri-context.jsonld",
      {
         "qudt-unit": "http://qudt.org/vocab/unit/"
      }
   ],
```

```
"@graph": [
  {
    "@id": "urn:demeter:ObservationCollection:Openweather_temperature",
    "@type": "ObservationCollection",
    "description": "Air temperature",
    "observedProperty": {
      "@id": "http://purl.obolibrary.org/obo/IDOMAL_0000568"
    },
    "hasMember": [
      {
        "@id": "urn:demeter:Observation:Openweather_temperature_1"
      },
      {
        "@id": "urn:demeter:Observation:Openweather_temperature_2"
      },
      {
        "@id": "urn:demeter:Observation:Openweather_temperature_3"
      },
      {
        "@id": "urn:demeter:Observation:Openweather_temperature_4"
      },
      {
        "@id": "urn:demeter:Observation:Openweather_temperature_5"
      }
    ]
  },
  {
    "@id": "urn:demeter:Observation:Openweather_temperature_1",
    "@type": "Observation",
    "resultTime": "2021-02-03T10:00:00.000Z",
    "hasResult": [
      {
        "@id": "urn:demeter:QuantityValue:Openweather_temperature_1_1",
        "@type": "QuantityValue",
        "numericValue": 12.85,
        "unit": {
          "@id": "qudt-unit:DEG_C"
        }
      }
    ]
  },
  {
    "@id": "urn:demeter:Observation:Openweather_temperature_2",
    "@type": "Observation",
    "resultTime": "2021-02-03T11:00:00.000Z",
    "hasResult": [
      {
        "@id": "urn:demeter:QuantityValue:Openweather_temperature_2_1",
        "@type": "QuantityValue",
        "numericValue": 13.91,
        "unit": {
          "@id": "qudt-unit:DEG_C"
        }
      }
    ]
  },
  {
    "@id": "urn:demeter:Observation:Openweather_temperature_3",
    "@type": "Observation",
    "resultTime": "2021-02-03T12:00:00.000Z",
    "hasResult": [
      {
        "@id": "urn:demeter:QuantityValue:Openweather_temperature_3_1",
        "@type": "QuantityValue",
        "numericValue": 15.73,
        "unit": {
          "@id": "qudt-unit:DEG_C"
        }
      }
    ]
  },
  {
    "@id": "urn:demeter:Observation:Openweather_temperature_4",
    "@type": "Observation",
    "resultTime": "2021-02-03T13:00:00.000Z",
```

```
        "hasResult": [
          {
            "@id": "urn:demeter:QuantityValue:Openweather_temperature_4_1",
            "@type": "QuantityValue",
            "numericValue": 17.24,
            "unit": {
              "@id": "qudt-unit:DEG_C"
            }
          }
        ]
      },
      {
        "@id": "urn:demeter:Observation:Openweather_temperature_5",
        "@type": "Observation",
        "resultTime": "2021-02-03T14:00:00.000Z",
        "hasResult": [
          {
            "@id": "urn:demeter:QuantityValue:Openweather_temperature_5_1",
            "@type": "QuantityValue",
            "numericValue": 18.18,
            "unit": {
              "@id": "qudt-unit:DEG_C"
            }
          }
        ]
      }
    ]
}
```


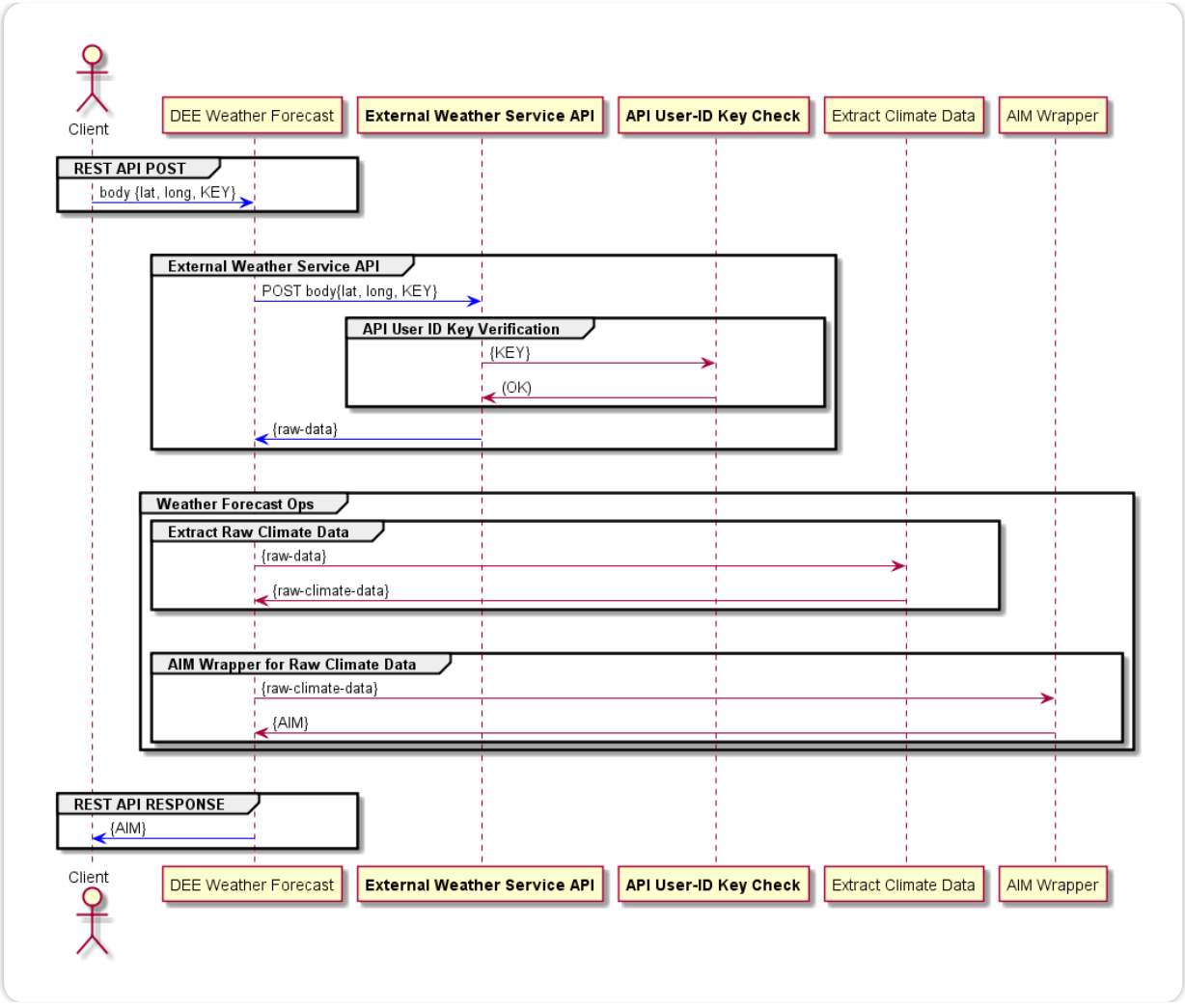

Figure 33 Weather Forecast Sequence Diagram

11.2.7 Data Analysis for Crop Irrigation

This component uses a mathematical agronomic model to estimate the needed irrigation water for a crop for a given day (litters by day) using the predicted reference evapotranspiration (ETo) and the crop coefficient (Kc) as well as other information related with the crop, the soil, the irrigation system and climate:

- Plot area (it is the same as the crop area) (m2)
- Crop category ("arable" or "woody")
- Number of plants (i.e. seeds planted, number of trees, etc.)
- Distance between plants in a row (m)
- Distance between rows (m)
- Crop plant family ("cereal", "poaceae", "herbaceus" or "fruit trees")
- Crop plant cup diameter (needed only for "woody")
- Crop plant coefficient (Kc) (this agronomic value depends on the crop phenological state)
- Soil percolating efficiency value.
- Water irrigation type ("drip" or "sprinkler")
- Water irrigation conductivity (dS/m at 25 °C)
- Rainwater (litres/day)
- Reference evapotranspiration (ETo) (mm/day).

It estimates the amount of needed irrigation water (litters by day) using an agronomic mathematical model programmed using Typescript. The result is returned using an AIM data model. For a “next day” prediction, rainwater forecast, and ETo prediction values for that day are used. The procedure is illustrated in Figure 34.

This component exposes a REST API:

```
curl --location --request POST 'http://localhost:port/cwmodel' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data-raw 'body = AIM data model'
```

This component receives in the **body** all the agronomic information using an expected AIM data model with the terms, properties and structure shown in next example:

```
{
  "@context": [
    "https://w3id.org/demeter/agri-context.jsonld"
  ],
  "@graph": [
    {
      "@id": "urn:demeter:AgriFarm:5c9cee4b649ebd42fef61b94",
      "@type": "AgriFarm",
      "name": "Mediterranean farm in Cartagena, Spain",
      "hasAgriParcel": [
        {
          "@id": "urn:demeter:AgriParcel:5c9cee4b649ebd42fef61b94",
          "@type": "AgriParcel",
          "name": "Tangerine parcel",
          "hasGeometry": {
            "@id": "urn: demeter:AgriFarm:geo:1",
            "@type": "Point",
            "asWKT": "POINT(37.7885278 -0.9463888 65)"
          },
          "area": 8000,
          "category": "woody",
          "numberOfPlants": 289,
```

```
            "distanceInRow": 6,
            "distanceBetweenRows": 5,
            "hasAgriCrop": {
              "@id": "urn:demeter:AgriCrop:5c9cee4b649ebd42fef61b94",
              "@type": "AgriCrop",
              "plantDiameter": 0.236,
              "plantFamily": "fruit trees",
              "cropCoefficient": 0.15,
              "maxWaterConductivity": 4
            },
            "hasAgriSoil": {
              "@id": "urn:demeter:AgriSoil:5c9cee4b649ebd42fef61b94",
              "@type": "AgriSoil",
              "percolatingEficiency": 0.9
            },
            "hasAgriWater": {
              "@id": "urn:demeter:AgriWater:5c9cee4b649ebd42fef61b94",
              "@type": "AgriWater",
              "irrigationType": "drip",
              "waterConductivity": 1
            }
          }
        ],
        "hasWeatherForecast": [
          {
            "@id": "urn:demeter:WeatherForecast:5c9cee4b649ebd42fef61b94",
            "@type": "WeatherForecast",
            "validity": "2021-04-01T00:00:00.00Z/2021-04-02T00:00:00.00Z",
            "precipitation": 0,
            "ETo": 1.2
          }
        ]
      }
    ]
}
```


To return the estimated calculated irrigation water, it uses the next AIM data model:


```
{
  "@context": [
    "https://w3id.org/demeter/agri-context.jsonld",
    {
      "qudt-unit": "http://qudt.org/vocab/unit/"
    }
  ],
  "@graph": [
    {
      "@id": "urn:demeter:Forecasting:1",
      "@type": "Forecasting",
      "description": "Irrigation water",
      "hasResult": [
        {
          "@id": "urn:demeter:idParcel1:Forecasting:cropWaterNeeds"
        }
      ]
    },
    {
      "@id": "urn:demeter:Forecasting:cropWaterNeeds",
      "@type": "QuantityValue",
      "numericValue": "108920.30379731294",
      "unit": {
        "@id": "qudt-unit:L-PER-DAY"
      }
    }
  ]
}
```

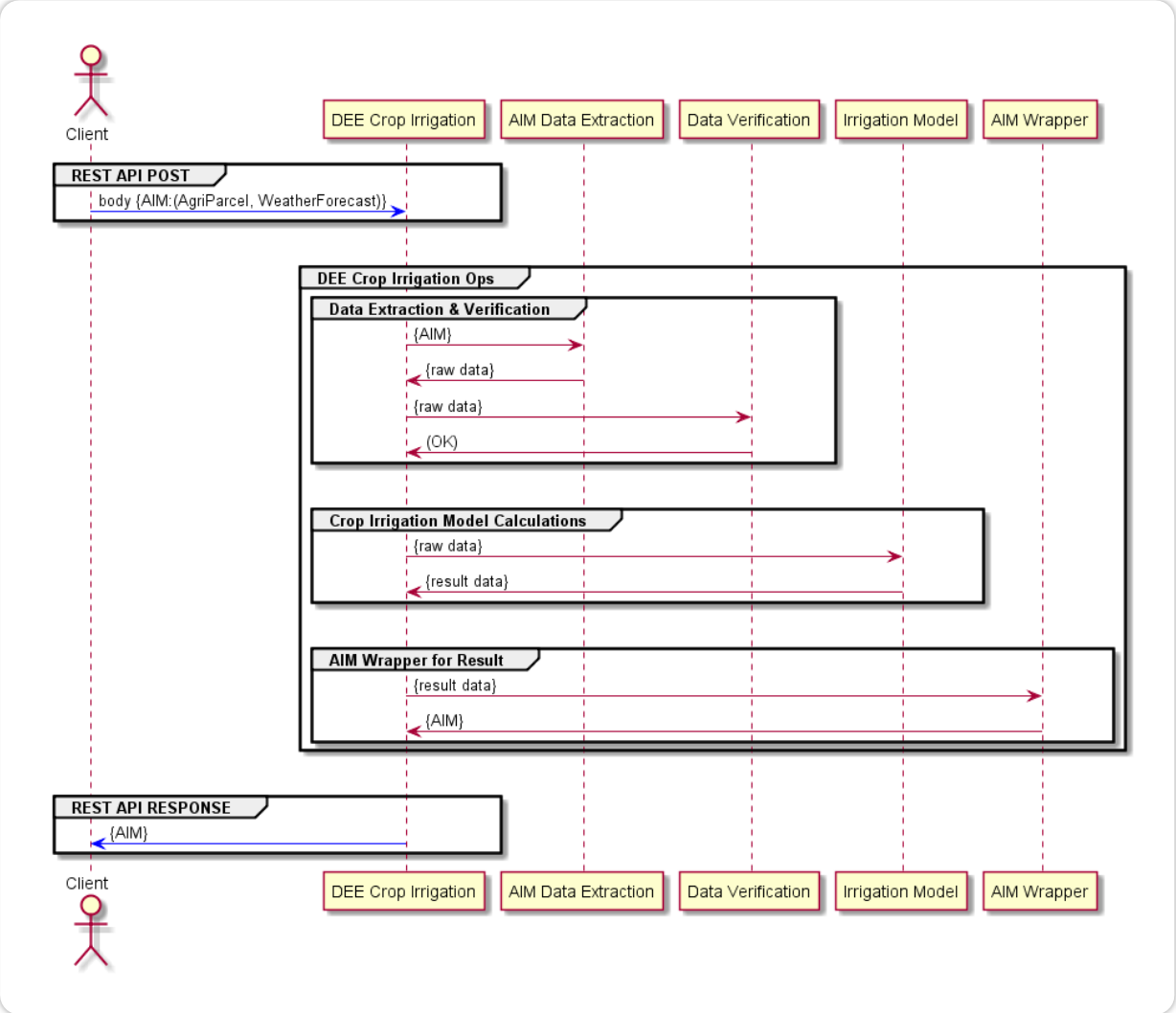

Figure 34 Crop Irrigation Water Estimation Sequence Diagram

11.3 *Machine Learning Tools*

11.3.1 Model Management

Model storage and management plays an important role for productive systems with machine learning components. Based on the state-of-the-art overview, provided in Section 7.2.1, MLFlow was determined as an appropriate fit for the requirements within DEMETER, and addresses them in the following ways:

1) It is a solution for storage of machine learning models and performance and accuracy metrics, catering to requirement 6.10 (Statistical Model Storage and Modification) and supporting efforts for requirement 6.12 (Data Analytics Performance and Accuracy Requirement)
2) It offers a Python and a R API integration, compatible with machine learning frameworks used by DEMETER developers.
3) MLFlow can run either on local hardware or cloud infrastructure and both options are available to DEMETER developers.
4) MLFlow offers a web-based user interface for easy accessibility.

5) MLFlow supports explainable AI methods, in particular SHAP[26]. In that way developers, and auditors (requirement 6.16) can investigate the influence factors of machine learning-based decisions within DEMETER. This is an important measure to prevent analysis bias (requirement 6.6) and allow for ethical machine-learning based decision making (requirement 6.11).

As indicated in Figure 35, the Model Management component is intended to be used by ML-training scripts of Targeted Analytics/Fusion Modules in order to store ML-Models (and associated metrics) in a managed way.

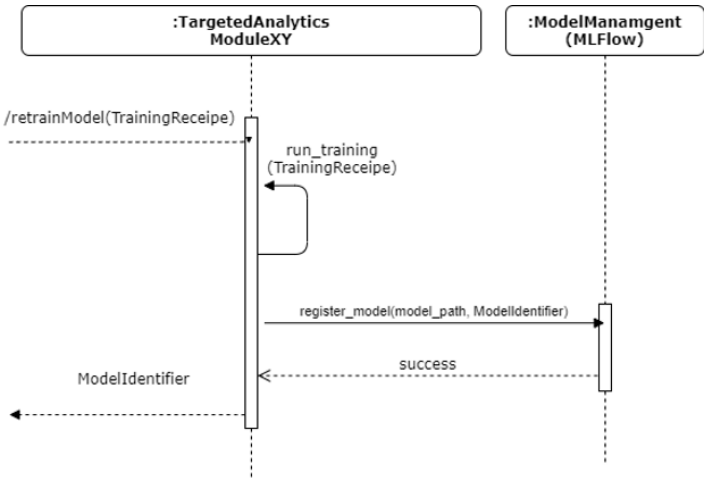

Figure 35: DEMETER model registry, using MLFLow

Implementation/Installation of Dockerized Version

To facilitate the on-premise deployment of DEMETER Model Management and to allow an integration with the DEMETER Enabler Hub, we provide a dockerized version, based on the original sources of MLFlow[27]. Because future changes to the original sources might be necessary to accommodate DEMETER requirements and the integration with other DEMETER components, DEMETER developers should use the Dockerfile provided in the DEMETER GitLab or the DEMETER Enabler Hub and no other deployments of MLFlow.

The Model Management component is based on a Dockerfile using **Python 3.7**. The Dockerfile automatically fetches and installs the latest version of **MLFlow 1.1.16** (at the time of writing) from the PyPI repository.

The respective Dockerfile is available at: https://gitlab.com/demeterproject/wp2/dataanalytics/model-management and was registered in the DEMETER Enabler Hub under the name "ML Model Management" (Resource ID: 6082d52bf190d8230b15ced1).

To run the Model Management component locally, use the following commands to build the Docker image:

[26] https://www.mlflow.org/docs/latest/python_api/mlflow.shap.html

[27] https://github.com/mlflow/mlflow

```
git clone https://gitlab.com/demeterproject/wp2/dataanalytics/model-
management.git

docker build . -t mlflow-tracking
```

After building the Docker image with the commands above, the Model Management component can be started, using the following command:

```
docker run -p 5000:5000 -v ${PWD}/mlruns:/mlruns -v
${PWD}/mlartifacts:/mlartifacts mlflow-tracking
```

This will expose the MLFlow endpoint on *Port 5000*, store metadata and metrics in SQLite-DB in folder *${PWD}/mlruns/mlruns.db*, and store model artifacts in folder *${PWD}/mlartifacts/*.

Implementation of Cloud-based Version

Apart from an on-premise deployment, the DEMETER Model Management component is also available as managed service, hosted by Fraunhofer FIT. This service is available to all DEMETER developers, but is meant for testing purposes only. Figure 36 shows the design of the MLFlow-based Model Management server, which is hosted at https://demeter.fit.fraunhofer.de .

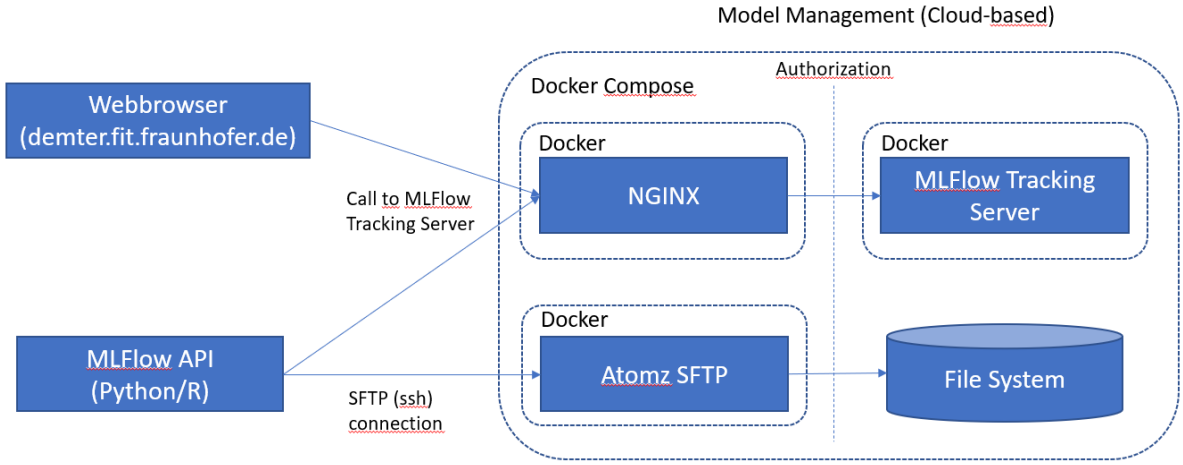

Figure 36: MLFlow Tracking Server Cloud-based Deployment

As the MLFlow Tracking Server does not offer authentication and authorisation facilities, a NGINX[28] reverse proxy was used to authorize and redirect external requests (from either a web browser or the MLFLow API) to the MLFlow Tracking Server (see Figure 36). Apart from regular REST-requests to the Tracking Server, a way to upload model artifacts to the server had to be established. This allows the MLFlow API to upload model artifacts (binary data) to the Server rather than storing it locally on the client. From the available options offered by the MLFlow API (s3, hdfs, ftp, sftp), we opted for SFTP as a secure and standardized method. The SFTP server was established using the Atomz SFTP[29] library, which was configured to store received files at a predetermined position on the server's file system, which is known by the MLFlow Tracking Server.

[28] https://www.nginx.com/
[29] https://github.com/atmoz/sftp

All individual services are deployed as Docker images and orchestrated using a Docker Compose script. We created a Jupyter notebook that contains the access credentials and demonstrates how to connect to the server: https://gitlab.com/demeterproject/wp2/dataanalytics/model-management/-/blob/master/examples/live_demo_notebook.ipynb

How to use it:

Examples on how to use the Model Management component, are available at: https://gitlab.com/demeterproject/wp2/dataanalytics/model-management/-/tree/master/examples

The file *minimal_example.py* contains just a few lines of code, demonstrating how to connect to an on-premise instance of the Model Management component and how to store a model and a metric.

A more extended example is the *fertilizers_example.py*. This script is based on a real-world use case from DEMETER and demonstrates where and how to integrate the Model Management component with a training script. Both examples are based on Python. For examples in R, refer to the following guide: https://www.mlflow.org/docs/latest/R-api.html.

Figure 37 shows a screenshot of the MLFlow UI. On the left side under "Experiments", the user can see a list of registered models. When the user clicks on a model on the left, all registered versions of the model will be shown in the right pane, accompanied by evaluation metrics and (optional) SHAP-based model explanations.

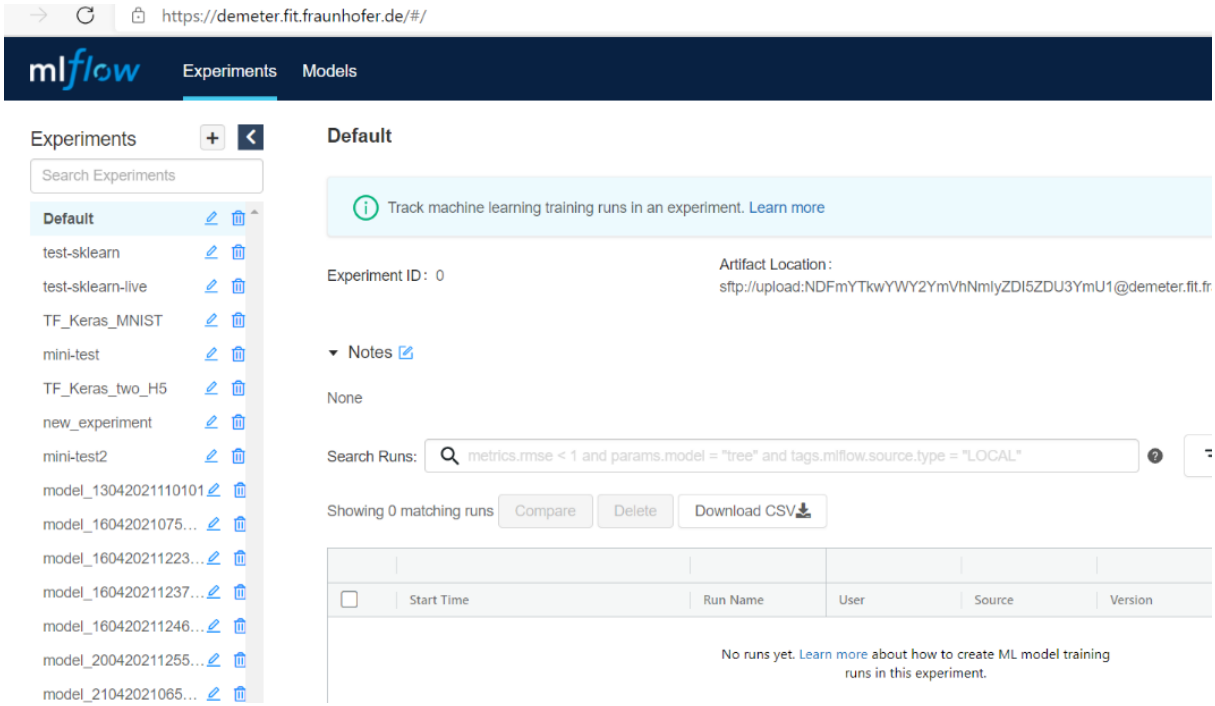

Figure 37: DEMETER Model Management UI, using MLFlow

11.3.2 **AIM-compliant Serving**

The AIM-compliant Serving component is complementary to the Model Management component. Models which were registered using the Model Management component can be fetched with an AIM-compliant Serving module and generate predictions on-demand. The result of the prediction will be

returned in AIM-format, while the input data will be checked for quality, using the Embeddable Data Quality component (see Section 11.1.3). This establishes a standardized but flexible way to access data analytics and data fusion enablers (requirements 5.7, 6.17). It also allows for a better integration with downstream components such as the DSS (requirements 5.29), while maintaining semantic interoperability between the services (requirement 5.4, 5.7, 6.9, 6.29) via AIM. A docker-based deployment allows for seamless scaling of the service to guarantee timely analytics (requirement 6.17). Furthermore, optional data quality checks can be integrated into the serving process to prevent inconsistencies due to data errors (requirement 6.17).

Implementation/Integration

The AIM-compliant Serving API was developed in Python, using the FastAPI[30] framework. It offers a very simple API endpoint through which a model prediction can be triggered:

/predict{model_name: str, model_version: int, input_data: str}

The model_name and model_version paraters determine which model is going to be fetched from the Model Repository. The *input_data* field should provide the data on which the prediction in to be performed. The *input_data* field is expected to provide one or multiple data points in an AIM-compliant json-ld format.

Figure 38 depicts the role of the AIM-Compliant Serving component within the Data Analytics and Knowledge Extraction Enabler. The following steps describe how a

Initially, a Targeted Data Analytics/Fusion Module must register a Model in the Model Repository, using the Model Management component (see Section 0).

(Optionally, the model can be equipped with Embeddable Data Quality Checks, as described in Section 11.1.3).

Then, the Analytics/Fusion Module developer must implement the AIM-Compliant Serving interface. This involves mapping of Input Data from AIM to the format which the ML-model accepts as input (*input_from_AIM* function), as well as a mapping from the ML-model's output to AIM (*prediction_to_AIM* function*)*. Further documentation and instructions are available in the repository at: https://gitlab.com/demeterproject/wp2/dataanalytics/analytics-api/-/blob/master/app/main.py

The service can them be tested and deployed in a Docker environment using the Dockerfile provided at top-level of the repository.

[30] https://fastapi.tiangolo.com/

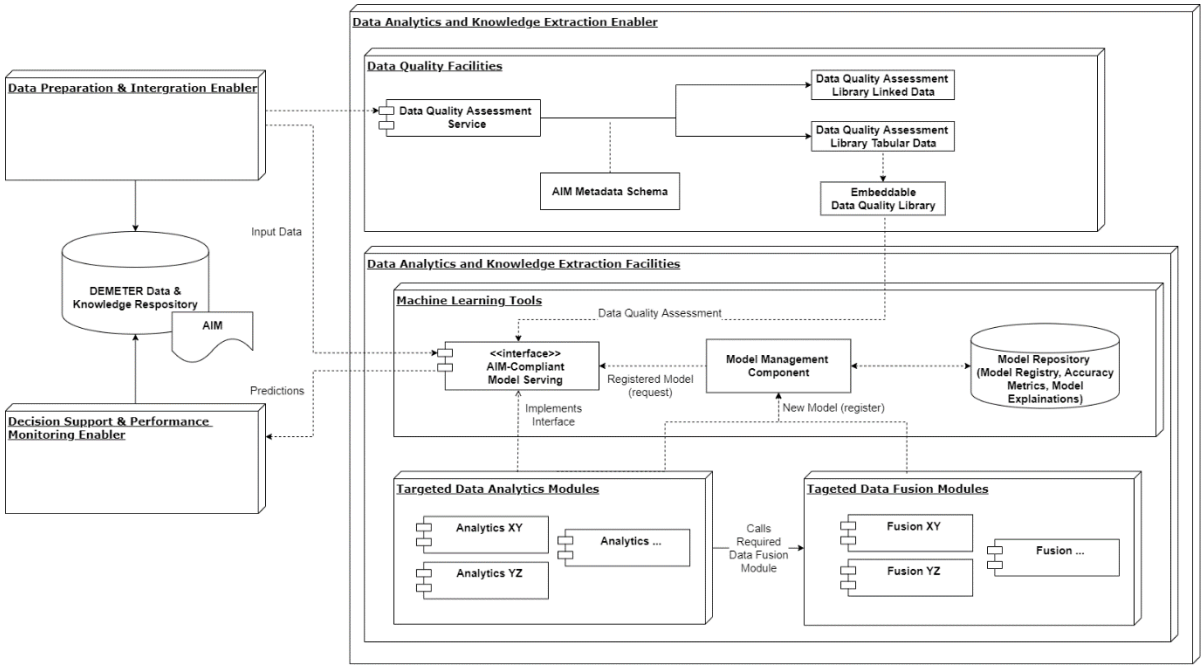

Figure 38: Data Analytics and Knowledge Extraction Enabler

12 Data Protection, Privacy and Traceability components

12.1 *Authentication*

12.1.1 Authentication Enabler Library Description

The Security Authentication Enabler library provides to the DEMETER components and the pilot developments an abstract way to access the Authentication OAuth 2.0 functionalities exposed by the DEMETER Authentication component REST API.

This library provides the following functions:

- Authentication by username and password
- Refresh authentication
- Revoke authentication token

The Security Authentication Enabler will interact with the Communication and Networking Enabler to obtain a secured communication channel to perform the authentication functionalities. This enabler will also provide to the Security Authorisation Enabler(s) the authentication token needed to perform authorisation functionalities.

The library needs to be imported in the programming language of choice, as well as the function. The following examples show how to import them for several well-known and widely used programming languages such as Python, Java and C#:

Python:

```
from demeter_authentication import login_with_password, login_with_client_credentail, refresh_token

authentication_token, expire_at = login_with_password("user1@example.com","password123")
```

Java:

```
import static demeter_authentication.*;

authentication_token, expire_at = login_with_password("user1@example.com","password123")
```

C#:

```
using demeter_authentication;

authentication_token, expire_at = login_with_password("user1@example.com","password123")
```

For more detailed information, see the README.md file in
https://gitlab.com/demeterproject/wp3/se/enablers/authentication/-/blob/master/README.md

The following functions are provided by this dynamic library in order to obtain, refresh and revoke authentication tokens:

Title	*Create token with Username and Password*
Function 1 *This field holds the name of the function used and the required (and optional) parameters*	
get_authentication_token(username, password)	
Output *This field holds the type of the output expected*	
Authentication token (string) and expiration (time/date)	
Params *This field holds the parameters (if any). Separated based on the fields below into required and optional.*	

Required:	
username=[string]	*String with the username to log in*
Required:	
password=[string]	*String with the password to log in*
Success response *<What should the status code be on success and is there any returned data? This is useful when people need to know what their callbacks should expect>*	
Authentication_Token:04c5b070-4292-4b3f-911b- *Authentication_Token_expires_at:"2018-03-20T15:05:35.697Z"*	*Authentication token and its expiration date to be used with following authentication/authorisation functions.*
Error response *This field holds the list of all possible error responses. Doing that, helps prevent assumptions of why the endpoint fails and saves a lot of time during the integration process.*	
400, "Invalid client: client is invalid"	*There has been a time out event while connecting to Keyrock Server*
400, "Invalid grant: user credentials are invalid"	*The username or password provided doesn´t match any registered user in Keyrock*
Sample call *This field holds a possible sample call to the described function*	
get_authentication_token ("user.example@example.com", "password1234")	
Notes *This field holds any additional helpful info related to the function described.*	

Title	*Refresh token*
Function 1 *This field holds the name of the function used and the required (and optional) parameters*	
refresh_authentication_token(authentication_token)	
Output *This field holds the type of the output expected*	
Authentication token (string) and expiration (time/date)	
Params *This field holds the parameters (if any). Separated based on the fields below into <u>required</u> and <u>optional</u>.*	
Required:	
authentication =[string]	*String with the authentication token*
Success response *<What should the status code be on success and is there any returned data? This is useful when people need to know what their callbacks should expect>*	
Authentication_Token: 65c6b870-3535-6b4f-345b-34a345f3ac7f *Authentication_Token_expires_at:"2018-03-20T15:05:35.697Z"*	*New authentication token and its expiration date to be used with following authentication/authorisation functions.*
Error response *This field holds the list of all possible error responses. Doing that, helps prevent assumptions of why the endpoint fails and saves a lot of time during the integration process.*	

400, "Invalid grant: refresh token is no longer valid"	*The token provided is no longer valid, therefore, a new authentication token is not provided.*
***Sample call** This field holds a possible sample call to the described function*	
refresh_authentication_token(65c6b870-3535-6b4f-345b-34a345f3ac7f)	
***Notes** This field holds any additional helpful info related to the function described.*	

Title	*Revoke token*
***Function 1** This field holds the name of the function used and the required (and optional) parameters*	
revoke_authentication_token(authentication_token)	
***Output** This field holds the type of the output expected*	
***Params** This field holds the parameters (if any). Separated based on the fields below into required and optional.*	
Required:	
authentication =[string]	*String with the authentication token*
***Success response** <What should the status code be on success and is there any returned data? This is useful when people need to know what their callbacks should expect>*	
0	*Success response for token deletion.*
***Error response** This field holds the list of all possible error responses. Doing that, helps prevent assumptions of why the endpoint fails and saves a lot of time during the integration process.*	
400, "Invalid grant: refresh token is no longer valid"	*The token provided is no longer valid.*
***Sample call** This field holds a possible sample call to the described function*	
revoke_authentication_token ("65c6b870-3535-6b4f-345b-34a345f3ac7f")	
***Notes** This field holds any additional helpful info related to the function described.*	

12.1.2 Authentication Component Description

The Demeter Identity Manager (IdM) Component is based on the FIWARE Keyrock GE and will provide the Keyrock's API for authentication based on the OAuth 2.0 protocol. The OAuth 2.0 protocol is defined in the RFC6749 standard, and described as: "The OAuth 2.0 authorisation framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf".

More information about the OAuth 2.0 protocol can be found at:

- https://auth0.com/docs/protocols/oauth2
- https://tools.ietf.org/html/rfc6749

The OAuth 2.0 protocol supports several grants ("methods") types for a client application to acquire an access token (which represents a user´s permission for the client to access their data) which can be

used to authenticate a request to an API endpoint. The Grant Types to be used for the Demeter components are:

- **Authorisation Code**: defined for apps running on a web server. The client will redirect the user to the authorisation server (Keyrock GE), and the user will then be asked to login to the authorisation server and to approve the client.
- **Password**: for logging in with a username and password.
- **Client credential**: the simplest of all of the OAuth 2.0 grants, this grant is suitable for machine-to-machine authentication where a specific user's permission to access data is not required.
- **Refresh token:** the access token obtained after being authenticated, provided with an expiration time; Keyrock GE provides a way to refresh the token which enables the client to get a new access token without requiring the user to be redirected.

The IdM provides functionalities to gain an identity within the system and manage the access privileges. Identity Manager Keyrock define the following common objects:

- **User**: Any signed-up user able to identify themselves with an email and password. Users can be assigned rights individually or as a group.
- **Application**: Any securable FIWARE application consisting of a series of microservices. Users or a group of users (i.e. organizations) will be granted permission to interact with the application.
- **Organization**: a group of users who can be assigned a series of rights. Altering the rights of the organization effects the access of all users of that organization. Users within an organization can either be members or admins. Admins are able to add and remove users from their organization, while members merely gain the roles and permissions of an organization. This allows each organization to be responsible for their members and removes the need for a super-admin to administer all rights.
- **Role**: a role is a descriptive bucket for a set of permissions. A role can be assigned to either a single user or an organization. A signed-in user gains all the permissions from all of their own roles plus all of the roles associated to their organization
- **X-Subject-Token:** identifies who has logged in on the application. This token is required in all subsequent requests to gain access.

The objects elements and functionalities, along with the relationship between the objects, can be seen in Figure 39:

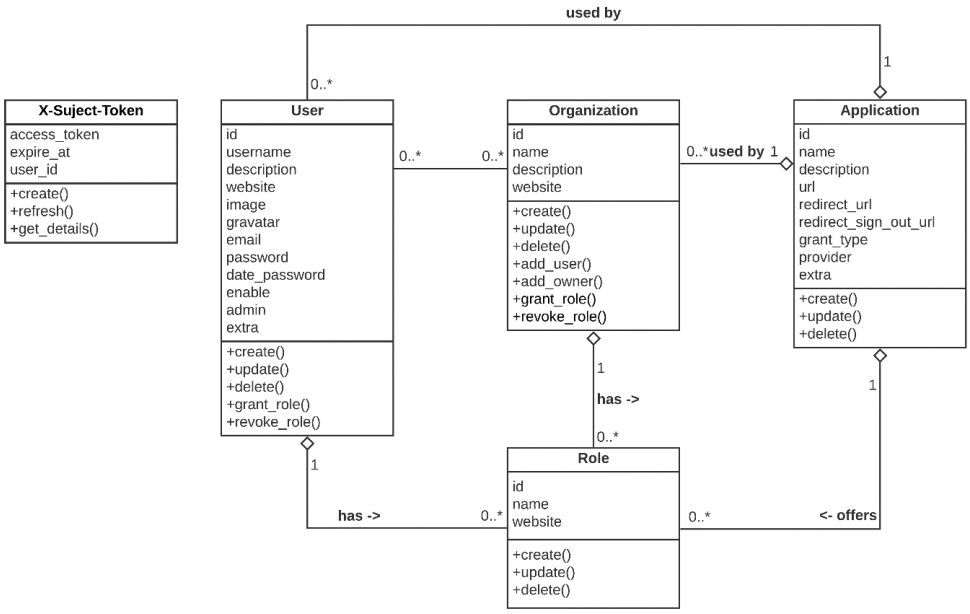

Figure 39 Relationship between Authentication objects

These IdM objects are used by the security layer to access DEMETER resources (functional layer), and they are needed to provide authentication and authorisation functionalities. Figure 40 shows the interaction that must be performed in order to obtain a X-Subject-Token for an authentication request with the Keyrock API:

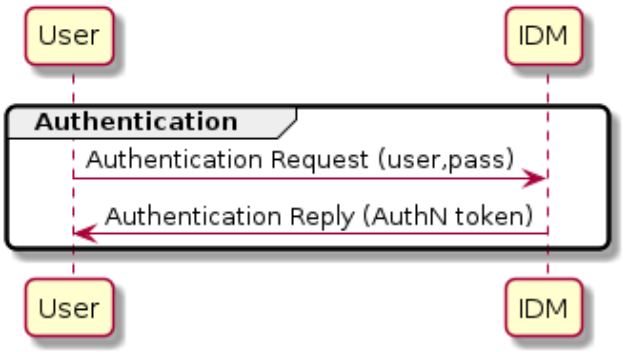

Figure 40 Authentication sequence diagram

12.1.3 Authentication Component Data models

User data model

The user data model is used to save, update and provide the user's information to the IdM. The user object contains the following fields:

- **Id:** a universally unique identifier (UUID) generated by Keyrock when the user is registered.
- **Username:** a sequence of characters that identifies a user when logging onto a computer or website.
- **Description:** a text that provides further details about the user.

- **Website:** URL provided at registration or during an update.
- **Image:** an image to be used by an application representing the user
- **Gravatar:** an image that follows you from site to site appearing beside your name when you do things like comment or post on a blog.
- **Email:** email provided by the user at registration or during an update; this field should be unique.
- **Password:** a string of characters, used to confirm the identity of the user.
- **Data_password:** date when the password was set.
- **Enable:** boolean value indicating whether the user is allowed to get access to resources using the IdM (for example used during registration of a user to validate his/her email).
- **Admin:** boolean value indicating whether the user has administration rights.
- **Extra:** field where a JSON object can be stored to provided extra information.

Application data model

The application object contains the following fields:

- **Id:** a universally unique identifier (UUID) generated by Keyrock when the application is registered.
- **Name:** a string of characters that identifies the application.
- **Description:** a text that provides further details about the application.
- secret:
- **URL:** application's URL
- **Redirect_URL:** URL required by the Oauth protocol.
- **Redirect_sign_out_URL:** the URL to which Keyrock will redirect a user if a sign out is performed from a service. If this is not configured a redirection to the domain indicated in URL parameter will be performed.
- **Grant_type:** list of grant type authentication allowed for the application.
- **Provider:** who is going to be the provider of the application: yourself or one of the organizations in which you are owner.
- **Extra:** field where a JSON object can be stored to provide extra information.

Organization data model

The organization object contains the following fields:

- **Id:** a universally unique identifier (UUID) generated by Keyrock when the organization is registered.
- **Name:** a sequence of characters (maximum length 64) that identifies the application.
- **Description:** a text that provides further details about the application.
- **website:** URL provided at registration or during an update.

Role data model

The Role object contains the following fields:

- **Id:** a universally unique identifier (UUID) generated by Keyrock when the organization is registered.
- **Name:** a sequence of characters (maximum length 64) that identifies the role.
- **Website:** URL provided at registration or during an update.

X-Subject-Token data model

The X-Subject-Token object contains the following fields:

- **Access_token:** string issued by Keyrock as a token identifier.
- **Method:** specifies the grant type method used for the authentication.
- **Expire_at:** If the access token expires, the server should reply with the duration of time the access token is granted for.

Authorisation data model mapping to AIM objects

As mentioned before, the previous data models of the objects used by the identity manager Keyrock may be linked to objects defined within the AIM, (e.g. person to AIM farmer). These links could be mapped using the definition provided at schema.org for person, organisation and role.

- **Person**

The definition of Person[31] provides a comprehensive set of properties related to a person. The user data model fields defined by Keyrock can be mapped to the following schema.org person's properties (see Table 5):

Table 5 Person fields mapping to Keyrock User Data Model

Keyrock User Data Model	Schema Person Data Model
id	identifier
username	name / alternatename
description	description
website	url
email (unique)	email
image	image

- **Organization**

The definition of Organization[32] provides a comprehensive set of properties related to an organization. The organization data model fields defined in Keyrock can be mapped to the following schema.org organization's properties (see Table 6):

Table 6 Organization fields mapped to Keyrock Organization Data Model

Keyrock Organization Data Model	Schema Organization Data Model
id	identifier
name	name/legalname/alternatename
description	description
website	url/sameas

- **Role**

[31] https://schema.org/Person
[32] https://schema.org/Organization

The definition of Role[33] provides a set of properties related to a role. The role data model fields defined in Keyrock can be linked to the following schema.org role's properties (see Table 7):

Table 7 Role fields mapped to Keyrock Role Data Model

Keyrock Role Data Model	Schema Role Data Model
id	identifier
name	rolname
description	description
website	url/sameas

12.1.4 Authentication component API description

The FIWARE Keyrock Identity Manager (IdM) API specifications comply with the Oauth 2.0 standard for authentication and user management and provide functionalities to access and manage information regarding the users, organizations, roles and applications.

A comprehensive FIWARE Keyrock API specification can be found at:

- https://keyrock.docs.apiary.io/
- https://swagger.lab.fiware.org/?url=https://raw.githubusercontent.com/Fiware/specifications/master/OpenAPI/security.Idm/Idm-openapi.json

IDs within Keyrock

IDs and tokens within Keyrock are generally universally unique identifier (UUID). The following table provides a description of the referenced Keyrock IDs together with sample values (see Table 8):

Table 8 UUID within Keyrock [from https://fiware-tutorials.readthedocs.io/en/latest/identity-management/index.html]

Key	Description	Sample Value
keyrock	URL for the location of the Keyrock service	Keyrock_URL:3005 for HTTP or Keyrock_URL:3443 for HTTPS
X-Auth-token	Token received in the Header when logging in as a user - in other words "Who am I?"	51f2e380-c959-4dee-a0af-380f730137c3
X-Subject-token	Token added to requests to define "Who do I want to inquire about?" - This can also be a repeat the X-Auth-token defined above	51f2e380-c959-4dee-a0af-380f730137c3
user-id	ID of an existing user, found with the user table	96154659-cb3b-4d2d-afef-18d6aec0518e
organization-id	ID of an existing organization, found with the organization table	e424ed98-c966-46e3-b161-a165fd31bc01
organization-role-id	type of role a user has within an organization either owner or member	member

The FIWARE Keyrock API provides functionality to manage:

- Authentication.
- Manage Applications.
- Manage Users.
- Manage Organizations.
- Manage Roles.

[33] https://schema.org/Role

These functionalities are described in the following sections based on the documentation provided at https://keyrock.docs.apiary.io/

Authentication

In order to manage and interact with the IdM through the API, an access token must be obtained to be included in the HTTP headers of the following actions (such as creating, updating or deleting users, organization, applications or roles). The API end points used to obtain, delete and get token details, along with the http verb to be used, are described in the following table:

Functionality	Endpoint	Http Verb
Create token	http://keyrock/v1/auth/tokens	Post
Get token details	http://keyrock/v1/auth/tokens	Get
Delete token	http://keyrock/v1/auth/tokens	Delete

Applications

In order to manage the applications that are registered to the IdM and are allowed to get access to DEMETER resources, the following endpoints are provided:

Functionality	Endpoint	Http Verb
Create application	http://keyrock/v1/applications	Post
Get application details	http://keyrock/v1/applications/application_id	Get
Update application	http://keyrock/v1/applications/application_id	Patch
Delete application	http://keyrock/v1/applications/application_id	Delete

Users

To manage the users that are registered to the IdM, the following endpoints are provided:

Functionality	Endpoint	Http Verb
Create user	http://keyrock/v1/users	Post
Get user details	http://keyrock/v1/users	Get
Update user	http://keyrock/v1/users/user_id	Patch
Delete user	http://keyrock/v1/users/user_id	Delete

Roles

In order to manage the roles that are defined for an application, the following REST API endpoints are provided:

Functionality	Endpoint	Http Verb
List roles	http://keyrock/v1/applications/application_id/roles	Get
Create role	http://keyrock/v1/applications/application_id/roles	Post
Get role details	http://keyrock/v1/applications/application_id/roles/role_id	Get
Update role	http://keyrock/v1/applications/application_id/roles/role_id	Patch
Delete role	http://keyrock/v1/applications/application_id/roles/role_id	Update

Organization

In order to manage the organizations that are registered to the IdM and are allowed to get access to DEMETER resources, the following endpoints are provided:

Functionality	Endpoint	Http Verb
List organizations	http://keyrock/v1/organizations	Get
Create organization	http://keyrock/v1/organizations	Post
Get organization details	http://keyrock/v1/ organizations /organizations _id	Get
Update organization	http://keyrock/v1/ organizations /organizations _id	Patch
Delete organization	http://keyrock/v1/ organizations /organizations _id	Update

Relationships between Applications, Organizations, Users and Roles

The IdM offers a series of endpoints in order to create relationships between the Applications, Organizations, Users and Roles, and to manage those relationships. The following functionalities are provided:

Functionality	Endpoint	Http Verb
List users within an organization	http://keyrock/v1/organizations/organization_id/users	Get
Add user to an organization	http://keyrock/v1/organizations/organization_id/users/user_id/ organization_roles/organization_role_id	Post
Remove user from an organization	http://keyrock/v1/organizations/organization_id/users/user_id/ organization_roles/organization_role_id	Delete
Read user´s role within an organization	http://keyrock/v1/organizations/organization_id/users/user_id/ organization_roles	Get
List granted organization roles	http://keyrock/v1/applications/application_id/organizations/ organization_id/roles	Get
Grant a role to an organization	http://keyrock/v1/applications/application_id/organizations /organization_id/roles/role_id/organization_roles/ organization_role_id	Post
Revoke a role from an organization	http://keyrock/v1/applications/application_id/organizations/ organization_id/roles/role_id/organization_roles/ organization_role_id	Delete
List granted roles to a user	http://keyrock/v1/applications/application_id/users/user_id/ roles	Get
Grant a role to a user	http://keyrock/v1/applications/application_id/users/user_id/ roles/role_id	Post
Revoke a role to a user	http://keyrock/v1/applications/application_id/users/user_id/ roles/role_id	Delete
Read user roles within an organization	http://keyrock/v1/applications/application_id/users/user_id/ roles	Get
List authorized organizations for an application	http://keyrock/v1/applications/application_id/organizations	Get
List authorized users for an application	http://keyrock/v1/applications/application_id/users	Get

12.2 *Authorisation*

12.2.1 Policy Administration Point

Policy Administration Point (PAP) is a Java project which is responsible for managing the authorisation policies (configuration point). PAP is not interfering in obtaining authorisation requests verdict, it does not offer an API for this aim.

The PAP presents a GUI for managing the XACML policies which must be defined according to a triplet (subject, resource, action). This GUI s available through a web explorer accessing to "http://<XACML-PublicIP>:8080/XACML-WebPAP-2".

In this sense, PAP allows to create/delete elements of this triplet, and in the same way, offers a friendly environment to define access control policies over them.

12.2.2 Policy Decision Point

This component offers an endpoint to evaluate access requests. It verifies if a subject can perform an action over a specific resource through defined access authorisation policies and issues an access verdict.

Policy Decision Point (PDP) is a Java project which is responsible for issuing positive/negative verdicts whenever an authorisation request is received.

When an authorisation request is received from the Capability Manager, it recovers from the body:

- the subject of the resource's request.
- the resource: endpoint + path of the resource's request.
- the action: method of the resource's request ("POST", "GET", "PATCH"…)

With this information, the PDP performs an

- Access to the XACML policies for validating authorisation requests and to obtain if the subject can access the resource and can perform the action over the resource (verdict).

Here is an example PDP request:

```
curl --location --request POST 'http://acs.bse.h2020-demeter-cloud.eu:8080/XACMLServletPDP/' \
--header 'Content-Type: text/plain' \
--data-raw '<Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
   <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
      <Attribute AttributeId=subjectType DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>subject</AttributeValue>
      </Attribute>
   </Subject>

   <Resource>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>resource</AttributeValue>
      </Attribute>
   </Resource>

   <Action>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
```

```
        <AttributeValue>action</AttributeValue>
      </Attribute>
   </Action>

   <Environment/>
</Request>'
```

With:

- subjectType: type of the subject of the resource's request.
- subject: subject of the resource's request.
- device: endpoint of the resource's request (protocol+IP+PORT).
- action: method of the resource's request ("POST", "GET", "PUT", "PATCH", "DELETE")
- resource: path of the resource request.

In this sense the next response requests could be obtained, depending on the verdict:

```
<Response>
  <Result ResourceID="resource">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
    <Obligations>
      <Obligation ObligationId="liveTime" FulfillOn="Permit">
      </Obligation>
    </Obligations>
  </Result>
</Response>
```

```
<Response>
  <Result ResourceID="resource">
    <Decision>NotApplicable</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```

```
<Response>
  <Result ResourceID="resource">
    <Decision>Deny</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```

12.2.3 **Capability Manager**

This component issues access tokens (capability token). The capability token is created only after the access request has been checked and the decision Permit has been made by the PDP.

The Capability Manager is developed in Python and makes use of the functionality developed in Java (.jar file). This component provides a REST API for receiving authorisation queries, which are tailored and forwarded to the PDP for a verdict.

When an authorisation request is received by the Capability Manager, it recovers from the JSON body:

- an authentication token which proceeds from authentication phase (access to IdM-Keyrock).
- an endpoint of the resource's request (protocol+IP+PORT). In DCapBAC scenario, it corresponds with PEP-Proxy component.
- the action/method of the resource's request ("POST", "GET", "PATCH"…)
- the path of the resource's request

With this information, Capability Manager performs:

- Access to authentication component (IdM-Keyrock) to validate authentication token.
- Access to XACML framework for validating authorisation requests (through PDP) and to obtain if the subject can access a resource and can perform the action over the resource (verdict).

If a positive verdict is received, finally, the Capability Manager issues an authorisation token, called Capability Token, which is a signed JSON document containing all the required information for the authorisation, such as the resource to be accessed, the action to be performed, and also a time interval during which the Capability Token is valid. This token will be required to access the resource (through PEP-Proxy).

Here is an example of a Capability Manager request:

```
curl --location --request POST 'https://acs.bse.h2020-demeter-cloud.eu:3030/' \
--header 'Content-Type: application/json' \
--data-raw '{"token": token,"ac": action, "de": device, "re": resource}
'
```

With:

- token: authentication token. For instance: "9f6a6fb9-51bb-4242-9008-f1a7b73e5a15"
- action: method of the resource's request ("POST", "GET", "PUT", "PATCH", "DELETE")
- device: endpoint of the resource's request (protocol+IP+PORT). For instance: "https://acs.bse.h2020-demeter-cloud.eu:1029"
- resource: path of the resource request. For instance: "/api/v1/resources"

In this sense the next response requests could be obtained depending on whether the Capability Token is obtained or not:

{"id": "jrqep8a59p2k3fduom4127dtsb", "ii": 1618386669, "is": "capabilitymanager@odins.es", "su": "8a94c9c1-7dc9-41e5-a558-f4db20e49a7a_ed6f1f0d-77fe-4c7c-83b3-0f40f9278d1b", "de": **device**", "si": "MEYCIQCvtmw+A1hIL+KQgl7VOechWHGvB1KVleJfqBZIM8WeYAIhAKp4UKgjU3WUkMhYUaw8DXm1MtRvPqyEcOsSJcB9CogC", "ar": [{"ac": **action**, "re": **resource**}], "nb": 1618387669, "na": 1618397669}

{"error":{"message":"Auth Token has expired","code":401,"title":"Unauthorized"}}
"Can't generate capability token"

12.2.4 **Policy Enforcement Point**

The PEP-Proxy secures access to resources by checking the capability token (CT) provided by a subject. The CT was issued previously from the Capability Manager, and if it is valid, the PEP-Proxy redirects the request to a specific endpoint. The PEP-Proxy is developed in Python and makes use of the functionality developed in Java (.jar file).

PEP_Proxy is the component responsible for receiving the queries aimed to access to a resource, they are accompanied by the corresponding Capability Token and forwards requests to the corresponding endpoint (for example BSE or DEH) and the responses back to the requester.

When an access resource request is received by PEP-Proxy:

- recovers the x-auth-token header (Capability Token).
- validate Capability Token.
- If Capability Token validation is successful, PEP-Proxy forwards the message and sends responses back to the requester.

This component does not offer a specific API. It exposes the same one as the component where access control is required. In this sense, the data model or API description can't be established in this section.

PEP-Proxy only requires an additional header called "x-auth-token" containing the Capability token. The other parts of the request, i.e. other headers, resource, query parameters and body, don't change regarding the format.

12.3 ***Traceability***

Permissioned blockchains register the transactions across the nodes of the network. The DEMETER Traceability Component will register authentication and authorisation events as transaction of the blockchain. The events contain the following fields:

- **Receiver**: user that obtain the right to access a Demeter resource.
- **Signature identifying the sender**: the security component.
- **Timestamp**: time of occurrence of the event.
- **Transfer right type**: auth(n)/auth(z) tokens.
- **An optional data field**: optional data to extend the information of the registered event.

Moreover, six extra functions have been implemented. Those have been developed to have more specific functionalities with respect to the previously mentioned events. They are focused on policies and tokens. On one hand, the policies can be registered, updated and checked. To register a policy, it must contain the following fields:

- **Domain:** Identification of the policy.
- **Digest:** The rules, subjects and actions of the policy.

With the same parameters, the digest of a registered policy can be updated. And if a policy is registered, its information can be checked. On the other hand, the tokens can be registered, revoked and verified. To register a token, it must contain the following field:

- **Id:** The token identification.

Using identifier parameter, a registered token can be revoked. Finally, a registered token can be verified to check the status, that is, if it is revoked or not.

The interaction with the Demeter Traceability Component has been implemented via an API. The API provides functionalities to register or read an event to the blockchain. The end points are described in the following Table 9:

Table 9. Overview – Traceability Agent endpoints

Functionality	Endpoint	Http Verb
Register event	http://audit_tool /registerEvent	Post
Get event details	http://audit_tool/getEvent	Get
Register policy	http://audit_tool /policy/register	Post
Update policy	http://audit_tool /policy/update	Post
Get policy details	http://audit_tool /policy/{domain}	Get
Register token	http://audit_tool /token/register	Post
Revoke token	http://audit_tool /token/revoke	Post
Verify token	http://audit_tool /token/{identifier}	Get

The "**registerEvent**" endpoint will register the transaction fields described above at the permissioned blockchain. This request needs the following parameters:

- **Sender**: identification of who is issuing the right, the DEMETER security components.
- **Recipient**: the beneficiary of the right, a DEMETER's user.
- **Payload**: details of the transaction.

It will return a 200 value with a key as a transaction ID if the transaction has been registered successfully, or 400 otherwise.

The "**getEvent**" endpoint allows to retrieve the details of a registered transaction using the transaction ID (hash) or execution time. It will return a 200 value and a payload with the event(s) details.

Policies section contains three different endpoints. The **"register"** endpoint will register the policies at the blockchain. Domain and digest are the parameters required to register a policy. The **"update"** endpoint allows to update a registered policy. This request also needs domain and digest parameters. The **"domain"** endpoint allows to retrieve the details of a registered policy using the domain parameter. It will return the digest associated with the domain and the version number (modifications) of the policy.

Tokens section is formed by three endpoints. The **"register"** endpoint allows to register the tokens at the blockchain. It requires the ID of the token as a parameter. The **"revoke"** endpoint will revoke the token, using the ID of the token. The **"identifier"** endpoint will return the status of the token associated with the introduced identifier. The status contains information whether the token is revoked or not.

12.4 *Confidentiality*

Confidentiality is provided by the Communication Encryption Enabler which is a software library written in C++. It can be imported and used in projects for the encryption and the decryption of text inside xml and json files. The library comprises an include directory, dll, .lib files and .so files to be used, respectively, in projects developed and running on Windows as well as projects developed and running on Linux. An illustration can be found in Table 10.

The library has only one access point: a method called 'startEnabler' which takes care, according to the inputs given, of dispatching the information about the file and the labels either to the encryption process or the decryption process. In the code repository, a .cpp file with an example of a main function calling encryption and decryption on an xml and json file is provided in order to show the correct usage of the library and the correct inputs to be given to the main method.

Table 10 Library Access Point description

Name	startEnabler	
Type	Inputs	Outputs description
method	1. typeOfInputFile (type of file you are calling the enabler on (must be 10 for a json file, or 11 for a xml file) 2. filePath (path of file you are calling the enabler on) 3. fileName (name of file you are calling the enabler on) 4. labels (list of labels whose values you want to encrypt) 5. actionOnInputFile (the action you are seeking to perform on the file) 6. key (the encryption/decryption key)	The method automatically creates the encrypted and the decrypted files inside the directory containing the original files.

13 Future plans and next steps

This deliverable represents the status of the enablers/components after round 1 of the pilots. For round 2 we describe the expected implications in this section. Of course, the developed solutions will require another feedback from the pilots. However, we do not expect a major change in the solutions, the core of the work should not be affected.

13.1 *Data Management and Integration components*

Regarding Data Management (see Section 10.1), the next steps include:

- Data management framework APIs improvements (code review, quality check, maintainability)
- Storage model improvements based on new DEMETER requirements
- Corrective maintenance and bug-fixing
- Enrichment of documentation to satisfy all DEMETER Stakeholders knowledge needs
- Technical integration support for the DEMETER PROVIDERS
- Continuous delivery
- Performance monitoring

Regarding the Data Preparation and Integration (DPI) components (see Section 10.2), the initial implementation of the CLI tool and corresponding enabler will be extended in the next stage with additional functionalities, such as supplementary pre/post processing routines, automatic semantic enrichment and link discovery, as well as with additional data access and collection methods to support different types of input data sources like relational databases and potentially non-SQL databases. Also, further pre-defined pipelines will be implemented in order to support more pilot specific use cases, including the implementation of the corresponding wrappers and/or mappings. Based on new pilot's needs, evolving from the implementation of pilot components or the use of other components (e.g., from WP3 or WP4), the access API methods will be also further extended to facilitate the retrieval of AIM-based data. Some predefined access methods will be provided, abstracting use of SPARQL queries over the underlying semantic data, but pilots will be able to generate their own methods too using SPARQL templates.

13.2 *Analytics and knowledge extraction components*

The next step for the analytics and knowledge extraction components is to improve the solutions.

To improve the data quality enablers (see Section 11.1) this is going to be done by workshops and feedback in the pilots. Running the applications in practice will surely reveal the potential for improvements. One main goal here is to restructure the codes from round 1 that they are easier to maintain and to adjust for future work and development.

Regarding the Targeted Data Analytics and data fusion (see Section 11.2) there will be different amounts of further efforts needed. For example, for the rice salinity identification, we will build an application in order to estimate rice salinity in farm areas that have no ground sensors, and at the same time identify the rice crop type. The output will be the decision whether to irrigate the rice - and open the water output valves for draining "stagnant" water due to increase salinity. The application will be implemented using python programming language and Machine Learning algorithms.

For maize fertilization, the next steps comprise further training of the model with new data coming from Pilot 1.3 and also parameter tuning in order to optimize the algorithm. Possible concerns are the possibility of not having a dataset big enough to properly train and test the algorithm. In that case, similar datasets need to be explored in order to achieve the same standards for the model. A very important aspect is also the integration of the model with other ML models that make the Decision Support System for Optimal Fertilizer Usage. Emphasis should be put on the way that the components communicate and the outputs are fused into the system so that they support farmers on having optimal nitrogen treatments on their fields. It is possible that the module might change the output or at least its format to facilitate the integration to the DSS and that is a challenge to be overcome.

The Predictive Model Training web service component is already fully integrated. It has been used as a basis of the Olive Phenology Prediction Service of WP4. The next step would be to extend it to further use cases and/or applications.

For the Pattern Extraction with Computer Vision we expect to start the data collection in the pilot 3.3 in order to create a proper training dataset needed to generate a model applied in a pilot. Although initial tests have been carried out with some lab-condition images of fruit flies (due to the nature of Pilot 3.3), it is expected to collect images by automatic traps in pilot 3.3 that will be used with this component (once a big amount of them are labelled to generate a model). To that point, this Pattern Extraction with Computer Vision component is being integrated in the WP4 component E.1 “Pest Estimation with Sterile Fruit Flies”. Additionally, next and current steps include the integration with the DEH, BSE, ACS, MLFLOW and AIM.

The Data analytics for Optimal Pesticide Usage is a proprietary component based on a forecast model created for four different fungal diseases. Considering the complexity of the given context and a number of parameters and influential factors involved for prediction, the next steps of the component development will be to simplify usage by lowering a number of input parameters. The focus will be on upgrading the algorithm to support assessment of the potential diseases with a lower amount of input information, based on the correlation task and number of devices used in the field.

For the Weather Forecast and the Data Analysis for Crop Irrigation we expect some possible updates regarding optimized code, since the enablers are in good progress.

For the Machine Learning Tools (see Section 11.3), we expect possible updates of the model management component with regards to its integration with embeddable data quality checks. Furthermore, the AIM-compliant serving API is expected to undergo updates related to the integration of Pilot-specific AIM-mappings.

13.3 *Data Protection, Privacy and Traceability components*

Demeter components and enablers (core and advanced) use the *Demeter* Access Control System as a mean for controlling and managing the access to them. ACS is composed by Authentication (see Section 12.1) and Authorisation (see Section 12.2) components, and can be enhanced using the Traceability (see Section 12.3) Component, as described in the previous Section.

ACS as a whole uses standard exchange formats and data (it uses protocols such as Oauth 2.0 or libraries like OpenSSL) and can serve those functionalities to new enablers and even applications of the pilots. Additionally, it provides a decoupling between the authorisation request and the access. Thanks to the use of the authorisation token, called Capability Token, a component can access as many times as the token is valid.

On the other hand, the integration with other Access Control Systems (authentication or authorisation third parties' components) is not considered in the scope of DEMETER. In case other pilots or applications are using external ACS, they will need to adapt and integrate these services by the development of specific connectors which are out of the scope of DEMETER.

For the Authentication (see Section 12.1), Authorisation (see Section 12.2) and Confidentiality (see Section 12.4) everything is in place; the latter component will be developed further in order to have an increasingly stable and versatile library. DEMETER Traceability component (see Section 12.3) use permissioned blockchain and an API Agent. The Agent connects the other components to the blockchain, avoiding the need of technical knowledge about repository implementation details. Those connections are made through several endpoints, and more functionalities can be added by deploying more smart contracts on the blockchain and adding the corresponding new endpoint in the Agent.

Choosing a blockchain follows the goal of providing an immutability feature for systems logs and data. However, to obtain this immutability, the architecture of deployment of the solution should have decentralized control over the computation nodes. The deployment of the particular architecture is out of the scope of the traceability component design and provision, but is key to offer the chased immutability feature.

A.1. Updated Technical Requirements

A.1.1. **Data integration: Semantic Interoperability/integration Requirements**

Requirement ID	DK3.1	**Version**	0.3	**Last Update Date**	12/04/2021
Title	Guarantee interoperability between communicating entities in the platform				
Description	Guarantee interoperability between communicating entities in the platform (e.g., using well-defined APIs, protocols or being able to exchange data in files in AIM format). This includes supporting heterogeneous integration points (connection protocols) by means of frameworks and open APIs for the quality-aware distribution of batch and stream processing analytics, with minimal development effort from application developers and domain experts. Relevant entities to support include: ● Irrigation systems exposing open and standard APIs for information exchange between the water management and control systems (Standard Model of Water Management irrigation). ISO 21622 is in a stable version. ● Tractors and machinery implementing standard protocols like ISOBUS (which includes CANBus), in order to extract, e.g., engine data (including fuel consumption) and emissions. Machinery also include those on the field (planters, harvesters, etc.) and in a dairy farm (e.g. milking robots, automated data capturing systems, feeding equipment) ● Farm management systems and smart agriculture platforms, such as DNET's agroNET platform, supporting standards interfaces like NGSI-LD. ● Breeding and milking farm systems, including veterinarian reports and audio ● Transportation and processing company systems ● Blockchain platforms, such as OriginTrail ● RFID (based on ISO 14223) or other devices used to identify and monitor animals. ● Apiary management systems, such as ControlBee, and related services, e.g., pollination optimization service ● Farmers' Decision Support Systems, such as ePSU ● EO (Web) services and systems providing different EO data like satellite imagery, geo-location, GIS, drone footage. ● IoT and sensor management systems, collecting real-time data from IoT devices (sensors and weather stations)				
Addressed by Enabler(s) / Module(s)	AIM BSE				

	Functional Interoperability enabler (FIE) Semantic Interoperability enabler (SIE)
Relevant Pilot(s)	• Irrigation systems: 1.1, 1.2, 1.3, 1.4, 3.1, 3.2 • Tractors and machinery: 2.1 • Farm management systems: 2.4 • Breeding and milking farm systems: 4.1, 4.2, 4.3, 4.4, 5.2, 5.4 • Transportation and processing company systems: 4.2, 5.1, 5.2, 5.4 • Blockchain platforms: 4.2, 5.1, 5.4 • RFID or other devices to monitor animals: 5.2 • Apiary management systems: 5.3 • Farmers' Decision Support Systems: 1.3, 1.4, 2.4, 3.1, 4.2, 5.3 • EO services providing different EO data: 5.3 • IoT and sensor management systems: 3.1, 3.2, 3.3, 4.1, 4.3, 5.2, 5.3
Relevant Task(s)	T2.1, T3.2
Relevant Objective(s)	O2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space O2 / WP3 2. Stakeholder Open Collaboration Space O3, O5 / WP4,7 7. Cost- and power-effective IoT data acquisition O2 / WP3 11. Data integration across the entire dairy supply chain O1, O4 / WP2, 5 14. Smart fruit pesticides management O6, O2, O1 / WP5, 2, 3 15. Open AKIS for irrigated crops O1, O2 / WP2, 5 16. Mechanical weed control using hyperspectral cameras and continuous crop data logging O6, O2 / WP5, 4, 3 17. Water Management Model and Coordination Broker O1, O6 / WP2, 5
Involved stakeholders/actors	Software and hardware providers for DEMETER
Prerequisite(s)	Specifications of the communicating entities in the platform must be well described: data formats and protocols to be exchanged must be clear and unambiguous
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	PSNC, m2xpert, TECNALIA
Status	Proposed + reviewed

Comments/Remarks	

Requirement ID	DK3.2	Version	0.3	Last Update Date	01/04/2021
Title	Integration of heterogeneous data types				
Description	DEMETER needs to provide mechanisms to integrate different types of incoming data. These mechanisms must be adapted to the data types identified by the pilot needs (WP5) on one hand. On the other hand, the integration interfaces must be flexible and robust enough to allow for data integration from other sources as well based on general standards of data exchange (e.g. structured by accepting standard structured data formats like e.g. (Geo-)JSON or similar). Integration of heterogeneous data types must consider granularity, lifespan and volume of the incoming data. Therefore, the data integration process must allow for application of rules regarding these meta-parameters: E.g. volume-based exclusion, addition of historization parameters and enforcement of dataset lifespan rules. The following types of data are expected to be integrated: ● Irrigation and fertilization data ● Weather data from various data sources (sensors, stations, web sites, external imported excel or image files). This data should include current data and predictions as well as historical data for (at least) temperature, rainfall and humidity. ● Engine data (including fuel consumption) and emissions (from CAN-Bus) ● Farm management data from different farm management systems, such as farm work organization, control of farm processes and control of machines, farm life organization, supply chain activities (production, transport, retail) ● Field data: detailed yield information, planting dates, etc. ● Machinery data ● Crop data (including quality data) ● DEM (Digital Elevation Map) data ● LPIS (Land Parcel Information System) official data of CAP integration ● Satellite data ● Milk related data ● Animal welfare data ● Pesticide usage data ● RFID data ● Sensor data (e.g., regarding soil, crops) ● Farm Telemetry Data				

	• EO data, e.g., satellite data from Landsat and Sentinel, GPS data (in different standard formats like GeoJSON and GeoXML), shapefiles, etc. • Meteorological stations data • Statistical data
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision Data Preparation and Integration Enabler
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	O2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm enabler dashboards 11. Data integration across the entire dairy supply chain 14. Smart fruit pesticides management 15. Open AKIS for irrigated crops 17. Water Management Model and Coordination Broker
Involved stakeholders/actors	Technology providers, semantic technologies experts
Prerequisite(s)	Pilots' requirements
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	PSNC, TECNALIA
Status	Proposed
Comments/Remarks	Changed objective, it's really about O2 not information models (O1)

Requirement ID	DK3.3	**Version**	0.1	**Last Update Date**	12/12/2019
Title	Access to linked (integrated) datasets				

Description	DEMETER needs to provide access to linked (integrated) datasets, from heterogeneous storage systems available in the DEMETER pilots. Such access/retrieval solution involves in particular a scalable triplestore with a Linked Data Interface (e.g., Virtuoso) enabling the provision of a federated layer over different datasets, but also relational databases (e.g., PostgreSQL) or other non-relational storages (e.g., Hadoop) when needed.
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision DPI Enabler Semantic store (Virtuoso) SPARQL endpoint
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	O2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 11. Data integration across the entire dairy supply chain 15. Open AKIS for irrigated crops 17. Water Management Model and Coordination Broker
Involved stakeholders/actors	Technology providers, semantic technologies experts
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	PSNC, ICCS
Status	Proposed
Comments/Remarks	

Requirement ID	DK3.4	**Version**	0.1	**Last Update Date**	04/12/2019
Title	Methods and tools for data integration				

Description	DEMETER needs to identify and select for reuse, as much as possible, suitable methods and tools for the generation and publication of Linked Data in order to provide an integrated view over different datasets. These components include: i) standard languages for the specification of mappings between source datasets and AIM, e.g., RML, R2RML; ii) tools for the (semi-)automatic generation of mappings, e.g., Geotriples, D2RQ, virtuoso sponger; iii) tools for data transformation that process the specified mappings, including RDFizers tools like Geotriples, RML-Processor, D2RQ, virtuoso sponger; iv) tools for query translation, service wrapping and data federation, such as D2RQ, Virtuoso, Metaphactory; v) tools for discovery of links between datasets, e.g., SiLK, Limes, geo-L
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision Data Preparation and Integration Enabler
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	O2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm enabler dashboards 11. Data integration across the entire dairy supply chain 14. Smart fruit pesticides management 15. Open AKIS for irrigated crops 17. Water Management Model and Coordination Broker
Involved stakeholders/actors	Technology providers, domain experts, semantic technologies experts
Prerequisite(s)	AIM, triplestore storage
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	PSNC, ICCS, m2xpert, TECNALIA
Status	Proposed + reviewed
Comments/Remarks	

Requirement ID	DK3.5	Version	0.1	Last Update Date	04/12/2019
Title	Select suitable tools for the semantic annotation of datasets				
Description	Identify and select, if possible, suitable tools for the semantic annotation of datasets, e.g., non-structured data. Some example tools include FOODIE annotation service, Agrotagger or DBSpotlight				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	• Information model for water management: 1.1, 1.2, 1.3, 1.4, 3.1, 3.2 • Information model of crops, pests, treatment and fertilization data: 1.3, 1.4, 2.2, 3.1, 3.2, 3.3, 5.1, 5.3 • Information model of soil data: 1.4, 3.2 • Information model for weather data: 1.4, 2.2, 3.1 • Information model of Vehicle data and emissions: 2.1 • Information model for farms and animals: 4.1, 4.2, 4.3, 4.4, 5.2, 5.3, 5.4 • Information model of farm economic data: 2.4 • Information model of status and field data: 1.3, 3.1, 3.4, 5.2 • Information model for the traceability of crops, dairy products, poultry products: 5.1, 5.2, 5.4				
Relevant Task(s)	T2.2				
Relevant Objective(s)	Objective 1: Information Modelling				
Relevant Innovation(s)	Innovation 8: Unified agriculture ontology O1 / WP2				
Involved stakeholders/actors	Solution providers, standardization organizations				
Prerequisite(s)	Data models should be based on existing ontologies				
Type	Functional				
Priority Level	Desirable				
Identified by Partner(s)	PSNC, m2xpert, TECNALIA				
Status	Proposed + reviewed				
Comments/Remarks					

Requirement ID	DK3.6	Version	0.1	Last Update Date	12/12/2019
Title	Linked Data exploration/visualization interfaces				
Description	DEMETER must provide Linked Data interfaces for exploration and exploitation by visualization frameworks like HSLayers Virtuoso or Metaphactory. These interfaces must be capable of providing data based on relational or semantic-/graph-based queries.				
Addressed by Enabler(s) / Module(s)	Data Preparation and Integration Enabler				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 4: Establish a benchmarking mechanism Objective 5: User Orientated Solutions				
Relevant Innovation(s)	1) Agriculture Interoperability Space 5) Farm enabler dashboards				
Involved stakeholders/actors	Technology providers, semantic technologies experts				
Prerequisite(s)	Datasets previously published in the Linked Data format				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	PSNC, TECNALIA				
Status	Proposed + reviewed				
Comments/Remarks					

Requirement ID	DK3.7	Version	0.1	Last Update Date	12/12/2019
Title	Query translation with interoperable API				

Description	DEMETER needs to guarantee a common query language interface via an interoperable API (e.g., REST or SOAP (if any) technologies) that allows to extract data from heterogeneous databases and data sources. The service API should be able to direct the query to a specific database or source, using query languages like SPARQL (able to provide integrated view over different datasets) or SQL syntax. The API should represent results in a standard format (e.g., JSON or XML), and it should support, if possible, content negotiation to allow the clients to specify their preferred representation for results. The querying to the different data sources should be, ideally, transparent to the user, who will only need to make the API call, and the service API will then be in charge of making any necessary translation to retrieve the data results from the corresponding data source. Then, the service will serialize results in the requested format, if supported.
Addressed by Enabler(s) / Module(s)	DPI Enabler ephedra (Metaphactory)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	O2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm enabler dashboards 11. Data integration across the entire dairy supply chain 14. Smart fruit pesticides management 15. Open AKIS for irrigated crops 17. Water Management Model and Coordination Broker
Involved stakeholders/actors	Technology providers, semantic technologies experts
Prerequisite(s)	Pilots' requirements
Type	Functional
Priority Level	Desirable
Identified by Partner(s)	PSNC, m2xpert, TECNALIA
Status	Proposed + reviewed
Comments/Remarks	

A.1.2. **Data Management Requirements**

Requirement ID	DK4.1	**Version**	0.3	**Last Update Date**	13/12/2019
Title	Data management lifecycle				
Description	DEMETER needs to guarantee a set of good practices, architectural techniques and tools able to manage the complete data lifecycle management process: ● acquiring data ● validating data ● storing data To ensure the accessibility, reliability of the data involved in DEMETER context. A Big Data approach should be guaranteed in order to better manage both the amount of data coming from heterogeneous sources: ● sensors, ● weather data provider ● farm systems and services and variety of data types: ● Soil data ● Water data ● Imagery data ● Production data ● Field data ● Farm data ● Meteorological data ● Geospatial data ● Logistics data Furthermore, it should be able to receive data from big data sources and data warehouses.				
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2				
Relevant Objective(s)	O2. Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 4. Earth Observation data service 5. Farm Enabler Dashboards 7. Cost- and power-effective IoT data acquisition				

	11. Data integration across the entire dairy supply chain 14. Smart fruit pesticides management 15. Open AKIS for irrigated crops 16. Mechanical weed control using hyperspectral cameras and continuous crop data logging
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG, ATOS, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.2	**Version**	0.3	**Last Update Date**	12/12/2019
Title	Data availability				
Description	DEMETER shall offer a constant access to the infrastructure to store and retrieve data, with an architecture ensuring a solid availability for the consumption and interaction of all the applications and processes involved (trying to contain the delay of those interactions within seconds, depending on the volume of data involved and on communication infrastructure and decentralized system performance).				
Addressed by Enabler(s) / Module(s)	DEMETER data management module (DEH and BSE)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2				
Relevant Objective(s)	O2. Build knowledge exchange mechanisms, ensuring data availability				
Relevant Innovation(s)	1. Agriculture Interoperability Space				

	2. Stakeholder Open Collaboration Space 5. Farm Enabler Dashboards 9. Secure Agricultural data sharing services 11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG, ATOS, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.3	**Version**	0.2	**Last Update Date**	12/12/2019
Title	Data integration mechanisms				
Description	DEMETER has to provide a solution capable of including and integrating data from heterogeneous sources. This solution has to be implemented providing the necessary mechanisms (services APIs) to allow the interoperability with the data from the different partners involved.				
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision DPI Enabler				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1, T2.2, T2.4				
Relevant Objective(s)	Objective 2. Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space 5. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain				

Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG, ATOS, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.4	**Version**	0.4	**Last Update Date**	12/04/2021
Title	API framework data and semantic interoperability				
Description	DEMETER needs to guarantee an API framework providing interfaces to the outside that allow operations on data, such as for relational databases CRUD operations (e.g. Create, Read/Retrieve, Update, Delete or Destroy) or for semantic repository browse, search, tagging, ontology management, export and import (in order to preserve data consistency). This API framework has to be designed supporting different types of protocols (e.g. HTTP), as well as using standard input and output formats for services (e.g. JSON).				
Addressed by Enabler(s) / Module(s)	DEMETER data management module (DEH and BSE) Virtuoso for ontology management				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1, T2.2				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space 9. Secure Agricultural data sharing services 11. Data integration across the entire dairy supply chain				

Involved stakeholders/actors	Technology providers, solution providers.
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.5	**Version**	0.2	**Last Update Date**	24/01/2020
Title	Services documentation and logging				
Description	The API framework will be properly documented, providing all information of all the resources generated. Additionally, all the infrastructure generated must generate detailed logs in order to guarantee the traceability of all the interactions carried out with it.				
Addressed by Enabler(s) / Module(s)	Policy enforcement point (PEP-Proxy) Capability manager Policy administration point (PAP) Policy decision point (PDP) Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T2.4				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanism				
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space				

	4. Earth Observation data service 9. Secure Agricultural data sharing services 10. Agri-food Decision support services based on SOA 15. Open AKIS for irrigated crops
Involved stakeholders/actors	Solution providers, technology providers.
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ATOS, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.6	**Version**	0.2	**Last Update Date**	07/12/2019
Title	Data storage availability for heterogeneous datasets				
Description	DEMETER needs to guarantee that the data coming from heterogeneous sources: ● Sensors ● weather data providers ● farm systems and services Structured and unstructured format from external or internal systems: ● geospatial data ● imagery data and specific data domains could be saved and made consistent. DEMETER should consider a solution for data storage taking into account that some of the involved data in the Pilots' could be stored locally and their technological infrastructure (in which case the data will be available on request through service APIs within the DEMETER architecture). Data lake methods (from structured files to unstructured data such as videos, emails and images) or warehouse (data in a structured format) could be taken into consideration given the variety of data.				

	In any cases DEMETER should guarantee that the selected method is able to support the portability of data, as well as having sufficient space to avoid running out of storage to manage all the incoming data.
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision DEMETER data management module (DEH and BSE)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms, adopt a best solution for data availability in DEMETER project
Relevant Innovation(s)	4. Earth Observation data service 11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	pilots' requirements
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG, ATOS, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.7	**Version**	0.1	**Last Update Date**	06/12/2019
Title	Data storage deployment				
Description	DEMETER may support different deployment solutions for data storage, including cloud (at least) and on-premises (if any). The idea is that DEMETER storage may be provided by external cloud providers, or by DEMETER partners, and it should be possible to move between the different solutions.				
Addressed by Enabler(s) / Module(s)	DEMETER data management module (DEH and BSE)				

Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Desirable
Identified by Partner(s)	ENG, ATOS, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.8	**Version**	0.1	**Last Update Date**	06/12/2019
Title	High availability of data storage				
Description	DEMETER needs to guarantee the continuous and reliable operation of the data storage system for a desirable length of time, and that all of a system's failure modes are known and well defined. The mechanisms to achieve that include (among others): ● Physical data redundancy ● System monitoring ● Definition and implementation of backup strategies, as well as data restore strategies ● Load balancing mechanisms along with reliability features to improve recovery time and overall uptime ● Fault tolerance strategies, so no data is lost in case of system failures, through failover solution mechanisms and data recovery mechanisms (failover-cloud-architecture, standby-servers, failover-clusters and similar concepts) Additionally, the mechanisms deployed for data loss prevention and availability should not have any impact on the regular data flows performance.				

Addressed by Enabler(s) / Module(s)	
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	ICT and Infrastructure and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG, ATOS, INTRA, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.9	**Version**	0.1	**Last Update Date**	09/12/2019
Title	Data storage of related metadata				
Description	DEMETER should permit the storage of arbitrary metadata as related data. The data described by the metadata can be of any type and be in any format, and the metadata should be accessible both by humans and machines. Concrete example: In a triple store holding an RDF graph, it should be possible to create a related but separate named graph holding the quality metadata of the original RDF graph, which became available after performing quality assessment.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				

Relevant Task(s)	T2.2
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	ICT (metadata modelling experts) and technological providers technology providers
Prerequisite(s)	pilots' requirements
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG, ATOS, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.10	**Version**	0.2	**Last Update Date**	12/12/2019
Title	Data synchronization				
Description	DEMETER needs to guarantee synchronization of data coming from heterogeneous sources through established methodologies and technologies. This will be done by means of data access mechanisms (both for providing and consuming data) that are designed with the data integrity in mind, keeping up to date all the different data instances stored within the DEMETER data storage solution.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space				

	2. Stakeholder Open Collaboration Space 5. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.11	**Version**	0.2	**Last Update Date**	23/01/2020
Title	Data synchronization frequency				
Description	DEMETER needs to guarantee that incoming data should be as fresh and accurate as possible, and the frequency of main data sets synchronization and updates should be monitored to allow for planning and managing these processes ahead, in case of any limitations on incoming data. The goal is to establish consistency among data from the source to the target data storage and vice versa and the continuous harmonization of the data over time.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm Enabler Dashboards				

	11. Data integration across the entire dairy supply chain 16. Mechanical weed control using hyperspectral cameras and continuous crop data logging 20. Tracking of organic supply chain by electronic labelling of wines
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	pilots' requirements
Type	Functional
Priority Level	Desirable
Identified by Partner(s)	ENG, ATOS, PSNC
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.12	**Version**	0.3	**Last Update Date**	13/12/2019
Title	Data synchronization (batch and real-time)				
Description	DEMETER needs to guarantee data synchronization in batch, and should guarantee data synchronization in real-time.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space 5. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain 20. Tracking of organic supply chain by electronic labelling of wines				

Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG
Status	Proposed + review
Comments/Remarks	

Requirement ID	DK4.13	**Version**	0.3	**Last Update Date**	29/01/2020
Title	Data synchronization stability				
Description	DEMETER needs to ensure the required internet service (with enough bandwidth) to process real-time updates of data regardless the data source, size, frequency (i.e. real time data), etc. The project has to avoid the appearance of potential communication problems (especially those involving data sources such as IoT devices or platforms that process sensor data). Additionally, in those cases where the required communication infrastructure is not available, ensure mechanisms to avoid data loss (e.g. storing data on-site and schedule the off-line upload of that data).				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space 5. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain				

Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG
Status	Proposed + Review
Comments/Remarks	

Requirement ID	DK4.14	**Version**	0.2	**Last Update Date**	13/12/2019
Title	Data synchronization in a pub-sub fashion				
Description	DEMETER should guarantee the synchronization of context data coming from heterogeneous data sources (e.g. IoT sensors, query results from consumer services like Data Brokerage Service, LPIS Land Parcel Information System, Farm Telemetry, EO data from Landsat and Sentinel or Meteorological stations) using technologies able to manage the entire lifecycle of context information including capabilities to publish and subscribe or similar to a message queue				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space 5. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain				
Involved stakeholders/actors	ICT and technological providers				

Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG, INTRA, LESPROJEKT
Status	Proposed + Review
Comments/Remarks	ATOS (Text re-structured to improve legibility)

Requirement ID	DK4.15	**Version**	0.2	**Last Update Date**	12/12/2019
Title	Data access methods				
Description	DEMETER needs to guarantee standard access methods to different types of data, showing high-level interfaces that are able to offer high-level functionality for access to DEMETER data, interoperability services (APIs) and heterogeneous datasets. The selected access method should guarantee access to on-premises data, or in the cloud.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	REFER TO section 2.1.2 of Demeter proposal				
Involved stakeholders/actors	ICT and technological providers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				

Identified by Partner(s)	ENG, PSNC
Status	Proposed
Comments/Remarks	

Requirement ID	DK4.16	**Version**	0.2	**Last Update Date**	28/01/2020
Title	Connection caching mechanisms				
Description	DEMETER should guarantee that data access interoperability services (APIs) provide caching mechanisms connection, such as connection pool (which instead of establishing and cleaning up database connections as needed, allow the use of an existing pool of connections that are kept open and available at all times) and provide performance, functionality and low maintenance.				
Addressed by Enabler(s) / Module(s)	DEMETER data management system (DEH and BSE)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space				
Involved stakeholders/actors	ICT and technological providers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ENG				
Status	Proposed + Review				
Comments/Remarks					

Requirement ID	DK4.18	Version	0.3	Last Update Date	13/12/2019
Title	Data preprocessing for advanced analytics				
Description	DEMETER needs to guarantee architectural design and techniques (such as Business Intelligence Tools and technologies) able to understand and ease the interaction with data coming from heterogeneous sources, getting useful information needed for advanced analytics in DEMETER.				
Addressed by Enabler(s) / Module(s)	Adaptive Visualisation Framework				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1, T2.2, T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Pilot's Decision Support System 3. Farm enabler dashboards				
Involved stakeholders/actors	ICT and technological providers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ENG				
Status	Proposed + Review				

Requirement ID	DK4.19	Version	0.3	Last Update Date	24/01/2020
Title	Business Intelligence client Tool				
Description	DEMETER needs to select a suitable Business Intelligence client Tool that would help users to understand trends and derive insights from				

	the data so that they can make better, tactical and strategic business decisions. DEMETER should help to guarantee that the input data for the tool is cleaned in order to obtain good results from the insights and for future data discovery.
Addressed by Enabler(s) / Module(s)	Adaptive Visualisation Framework
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 3: Empower the farmer, as a prosumer, to gain control in the data-food-chain Objective 4: Establish a benchmarking mechanism for agriculture solutions and business Objective 6. Demonstrate the impact of digital innovations across a variety of sectors and at European level
Relevant Innovation(s)	5. Farm Enabler Dashboards 10. Agri-food Decision support services based on SOA 14. Smart fruit pesticides management 19. Sensor-based organic ingredient verification in biscuit production
Involved stakeholders/actors	Solution providers, technology providers
Prerequisite(s)	pilots' requirements
Type	Functional
Priority Level	Desirable
Identified by Partner(s)	ENG, ATOS, PSNC
Status	Proposed + Review
Comments/Remarks	

Requirement ID	DK4.20	**Version**	0.2	**Last Update Date**	13/12/2019
Title	Data discovery tools and technologies				

Description	DEMETER must ensure that the data discovery tools and technologies selected as Advanced Business Intelligence Tools, guarantee a set of features for data discovery: ● Advanced data search function (on structured and unstructured data) ● Data storage features (on a proprietary database) to be able to model data from disparate sources ● Pull data from a good set of datasets ● Join data from different sources ● Link data to different dataset (data relationships) ● Support and implement Geo-location features ● Support and implement reports, charts and table ● Support for advanced analytics (e.g. R, Python, Apache Spark (if any) to support for example statistical analysis ● Integration with data preparation, analysis and analytics.
Addressed by Enabler(s) / Module(s)	
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.1, T2.2, T2.3
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology 9. Secure Agricultural data sharing services
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG
Status	Proposed + Review
Comments/Remarks	ATOS (Seems OK)

Requirement ID	DK4.21	Version	0.2	Last Update Date	13/12/2019
Title	Data Aggregation Tools				
Description	DEMETER shall offer a set of tools and technologies for data aggregation that allow for preparation of combined datasets from various sources within the DEMETER data storage layer. These tools and technologies shall reduce time spent during data mining clean-up and data preparation phases to ease the evaluation of data sources for later statistical analysis.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space 3. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain				
Involved stakeholders/actors	ICT and technological providers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ENG				
Status	Proposed + reviewed				
Comments/Remarks					

Requirement ID	DK4.22	Version	0.2	Last Update Date	12/12/2019
Title	Data Flow / Pipeline Procedure Support				

Description	As an extension to requirement DK4.21 DEMETER shall provide means to create automated data flow, pre-processing, cleaning and aggregation procedures. Therefore, DEMETER must allow definition of data pipelines. These data pipelines shall not only allow for classical ETL transactions (e.g. batch processing from DB to DB) but also for more advanced, also stream-based, data pipelines from participating actor to actor.
Addressed by Enabler(s) / Module(s)	DPI Enabler (ETL)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space 3. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	Technology providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ATOS
Status	Proposed + reviewed
Comments/Remarks	

Requirement ID	DK4.23	**Version**	0.2	**Last Update Date**	13/12/2019
Title	Data Grouping, Filtering and Aggregation Function Set				

Description	As an extension to requirement DK4.21 DEMETER needs to guarantee a set of functions in order to allow for the aggregation of multiple datasets, data types and data sources: ● Grouping datasets to organize data in different categories, organizing the results based on the values of the key, in order to have multiple datasets in the same table result ● Dataset filtering of grouped data on large scale datasets (to filter grouped data in order to have a single table result) ● Dataset aggregation on numerical values should be able to lead to a single result if the aggregation output can in principle be a scalar value or if aggregation rules on unstructured data enforce a scalar output ● Data aggregation could also occur at the database level, as in the case of NoSQL server ● The system must implement the whole set of basic functions that allow data to be aggregated: ○ Sum ○ Average ○ Maximum ○ Minimum ○ Count ○ Count distinct ● The system must implement advanced functions that allow data to be aggregated: ○ Windowing Functions ○ Ranking functions ● The system should persist the aggregated value returned to the query in a database
Addressed by Enabler(s) / Module(s)	
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 2. Stakeholder Open Collaboration Space 5. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain

Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG, m2Xpert
Status	Proposed + reviewed
Comments/Remarks	ATOS (According to GA, Farm Enabler Dashboards is Objective 5)

Requirement ID	DK4.24	**Version**	0.2	**Last Update Date**	12/12/2019
Title	Data Warehousing Support				
Description	DEMETER shall support technologies that enable information aggregation and analysis from multiple database systems similar to concepts like e.g. OLAP Cubes. This system must support a predefined set of features: ● Must allow to query (a subset of) a database through a high level of representation of entities and relationships ● Must allow the selection of table columns from different data sources and set filters ● Must not require any knowledge of data structures ● Must require a semantic knowledge of the data ● Must allow free management of results ● Must support data export ● Must allow repeatable execution of requests ● Must work on a limited data domain				
Addressed by Enabler(s) / Module(s)	Adaptive visualisation framework				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space				

	2. Stakeholder Open Collaboration Space 3. Farm Enabler Dashboards 11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG
Status	Proposed + reviewed
Comments/Remarks	

Requirement ID	DK4.25	**Version**	0.3	**Last Update Date**	28/01/2020
Title	Multi-tenancy				
Description	DEMETER needs to guarantee that the data is distributed according to the criteria of multi-tenancy data management. This criterion must be maintained at all infrastructure levels: web servers (for the incoming data to DEMETER), infrastructure services (interoperability services between technological components and for the exchange of data within DEMETER), database				
Addressed by Enabler(s) / Module(s)	DEMETER data management system (DEH and BSE)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2				
Relevant Objective(s)	O1: Analyze, adopt, enhance information models O2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space				

Involved stakeholders/actors	Solution providers, Technology Providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ENG
Status	Proposed
Comments/Remarks	

A.1.3. **Data Quality & Fusion Requirements**

Requirement ID	DK5.1	**Version**	0.1	**Last Update Date**	03/12/2019
Title	Fusion of data from (content-wise) heterogeneous data sources				
Description	Data fusion shall support fusion of data from (content-wise) heterogeneous data sources.				
Addressed by Enabler(s) / Module(s)	Irrigation requirements estimation. Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology 11. Data integration across the entire dairy supply chain				

Involved stakeholders/actors	Solution providers, …
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.2	**Version**	0.1	**Last Update Date**	03/12/2019
Title	Selection of information extracted from a metadata model				
Description	Data fusion should permit selection of information extracted from a metadata model.				
Addressed by Enabler(s) / Module(s)	Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1, T2.2, T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology 11. Data integration across the entire dairy supply chain				
Involved stakeholders/actors	Farmers/Domain experts, solution providers, …				
Prerequisite(s)	None				

Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ROT
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.3	**Version**	0.1	**Last Update Date**	06/12/2019
Title	Data fusion support for multiple file formats				
Description	Data fusion shall support the integration of multiple file formats such as CSV, XLS, JSON and XML. Briefly, data fusion (ingestion) shall support parsing, serializing and deserializing of structured data like JSON, XML and CSV since these are the most used and well-known standards and formats in which data sources formulate the acquired data. Supporting a vast number of formats shall also facilitate the analytical usage which is of the utmost importance within DEMETER.				
Addressed by Enabler(s) / Module(s)	Data Access Facilities Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance Information Models Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology 11. Data integration across the entire dairy supply chain				

Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed
Comments/Remarks	We might be missing a requirement in DK1, which is to have models for different types of data formats.

Requirement ID	DK5.4	**Version**	0.1	**Last Update Date**	2019-12-12
Title	Knowledge extraction (Agriculture Information Model - based)				
Description	Data fusion must provide mechanisms for extracting AIM-based (Agriculture Information Model - based) knowledge from specific raw data. [IR1]				
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	4.3[LC2], 4.4[LC3] [VAM4] ... (ALL)				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 2: Knowledge Exchange Mechanisms				
Relevant Innovation(s)	11. Data integration across the entire dairy supply chain				

Involved stakeholders/actors	Solution providers
Prerequisite(s)	• AIM has been specified. • Raw data sources have been identified. • Connection between raw data and domain/expert knowledge is clear (e.g., from expert interviews)
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.5	**Version**	0.1	**Last Update Date**	04/12/2019
Title	Data fusion techniques for distributed (big) data				
Description	DEMETER needs to provide appropriate data fusion techniques in order to process and analyze distributed data coming from many different sources, and also provide, obtain and automatically process sensing big data.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	O1: Analyze, adopt, enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.				
Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm enabler dashboards 11. Data integration across the entire dairy supply chain				

Involved stakeholders/actors	Farmers, solution providers, semantic technologies experts.
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS, FhG.FIT, FhG.IESE, ROT
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.7	**Version**	0.1	**Last Update Date**	31/01/2020
Title	Standardized data fusion				
Description	Data fusion should utilize existing software packages, libraries and frameworks in order to fuse data from different sources (where appropriate) and expose open and standard APIs. This fact will allow irrigation communities to expand their irrigation system with different vendors' devices, having a heterogeneous environment that enables the fusion among the sources (1.1, 1.2). In addition, this data fusion enables the use of tools which expect a different data model (e.g. by embedding data into new models after they are fused); potentially supporting the standardization, normalization and merging of data coming from disparate sources.				
Addressed by Enabler(s) / Module(s)	AIM-compliant Serving Analytics API Moisture estimation Plant Water Status Estimation Plant Water Status Estimation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	O1: Analyze, adopt, enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.				

Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm enabler dashboards 8. Unified agriculture ontology 11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	Solution providers, semantic technologies experts.
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS, FhG.FIT, ROT
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.8	**Version**	0.1	**Last Update Date**	31/01/2020
Title	Fusing data from different systems				
Description	Data fusion needs to enable fusing data originating from existing systems involved in the pilots (legacy systems), e.g., data fusion shall implement a system which fuses imagery and sensor data for pest management control (3.3, 5.1), or data fusion needs to provide support for components allowing to harness satellite data for applications in farm telemetry and fuse them with real-time streaming data from wireless sensor networks, with particular interest in Crop Monitoring and Predictions.				
Addressed by Enabler(s) / Module(s)	Data Integration Components Moisture estimation				
Relevant Pilot(s)	3.3, 5.1, ...				
Relevant Task(s)	T2.3				
Relevant Objective(s)	O1: Analyze, adopt, enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.				

Relevant Innovation(s)	1. Agriculture Interoperability Space 4. Earth Observation data service 5. Farm enabler dashboards 11. Data integration across the entire dairy supply chain 16. Mechanical weed control using hyperspectral cameras and continuous crop data logging
Involved stakeholders/actors	Solution providers, semantic technologies experts.
Prerequisite(s)	None
Type	Functional
Priority Level	Medium
Identified by Partner(s)	ICCS, ROT
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.11	**Version**	0.2	**Last Update Date**	04/05/2021
Title	Provide recommendations based on data quality measurement results				
Description	On one hand, data quality components should measure the data quality (by mathematical formula). But on the other hand, also provide quality assessment results, that means an interpretation of the measurement result whether the quality is acceptable or not. For example, whether the measurement exceeds pre-defined accepted threshold.				
Addressed by Enabler(s) / Module(s)	Data Preparation Components				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models				

Relevant Innovation(s)	1. Agriculture Interoperability Space 11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	Farmers, solution providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	FhG.IESE
Status	Proposed
Comments/Remarks	This requirement merges DK5.11 and DK5.12 (some aspects of DK5.12 were also covered in DK6.15)

Requirement ID	DK5.13	**Version**	0.1	**Last Update Date**	06/12/2019
Title	Support for interoperability and interchangeability of data quality assessment				
Description	Assessment is necessary to ensure the quality of both the acquired and the processed data. The components should be able to quantify the quality tests by providing related metrics as well as the outcome of the assessment in a way that facilitates interoperability and interchangeability and is comprehensible by machines. A way is by using existing software packages, libraries, frameworks (where appropriate) and standards such as the W3C Data Quality Vocabulary.				
Addressed by Enabler(s) / Module(s)	Data Quality Assessment as-a-service				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models Objective 2: Build knowledge exchange mechanisms				

Relevant Innovation(s)	1. Agriculture Interoperability Space
Involved stakeholders/actors	ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS, Fraunhofer
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.14	**Version**	0.2	**Last Update Date**	30/01/2020
Title	Decision-support for data source selection				
Description	DEMETER components for data quality should allow the assessment of the quality of the data of each alternative data source available (if there are several). Thus, different modes of access to these data sources (such as static data dumps, query APIs, or streams) should be considered in developing this component. This assessment should be used to support the decision-making (i.e., the decision of which data source could/should be used).				
Addressed by Enabler(s) / Module(s)	AIM-compliant Serving				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	O1: Analyze, adopt and enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.				
Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm enabler dashboards				

Involved stakeholders/actors	Farmers, solution providers, semantic technologies experts.
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS, FhG.FIT, FhG.IESE, ROT
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.15	**Version**	0.2	**Last Update Date**	30/01/2020
Title	Optimum value extraction				
Description	Data quality components need to extract the optimum values per data type (from multiple sources potentially) filtering out irrelevant, outdated or low-quality data (if appropriate). Therefore, the data quality components shall provide support in which boundaries the different data values should be (range/ thresholds).				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	3.1[LC1], 5.2[LC2], …[LC3] [VAM4] (ALL)				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Knowledge Exchange Mechanisms				
Relevant Innovation(s)	6. Performance evaluation of Decision Support Systems				
Involved stakeholders/actors	Solution providers				
Prerequisite(s)	Given the data in the pilot or use case, an understanding of “optimum” w.r.t. relevance, timeliness and quality is required.				
Type	Functional				

Priority Level	Medium
Identified by Partner(s)	ICCS
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.17	**Version**	0.1	**Last Update Date**	30/01/2020
Title	Different measurements for measuring data quality				
Description	Data quality components should utilize the appropriate quality measurements for each type (or source if appropriate) of data; e.g. different quality measurements might apply to image data and others to weather data etc.				
Addressed by Enabler(s) / Module(s)	Data Quality Assessment as-a-service Embeddable Data Quality Checks				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models				
Relevant Innovation(s)	1. Agriculture Interoperability Space 4. Earth Observation data service 5. Farm enabler dashboards 11. Data integration across the entire dairy supply chain				
Involved stakeholders/actors	Solution providers				
Prerequisite(s)	Given the data in the pilot or use case, an understanding of “optimum” w.r.t. relevance, timeliness and quality is required.				
Type	Functional				
Priority Level	Medium				
Identified by Partner(s)	ICCS, FhG.FIT, FhG.IESE				

Status	Proposed
Comments/Remarks	

Requirement ID	DK5.19	**Version**	0.1	**Last Update Date**	03/12/2019
Title	Handling of missing or contradicting data				
Description	Data fusion and Data Quality components should be able to deal with missing or contradicting data (perhaps with assistance from the analytics and DS system that takes as input the fused data). For example, IoT data stream analysis might be used for the detection of abnormal sensor measurements.				
Addressed by Enabler(s) / Module(s)	Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models Objective 2: Knowledge Exchange Mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 11. Data integration across the entire dairy supply chain				
Involved stakeholders/actors	Farmers, solution providers, …				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS				
Status	Proposed				

Comments/Remarks	

Requirement ID	DK5.20	**Version**	0.1	**Last Update Date**	2019-12-12
Title	Quality-preserving collection and fusion				
Description	The quality of the data should be preserved in its collection and fusion phase, avoiding, for instance, compression methods that might impact on the information potentially inferable from it (e.g. applying compression algorithms with image data to ease its transference, but losing quality).				
Addressed by Enabler(s) / Module(s)	Irrigation requirements estimation Moisture estimation Plant Water Status Estimation				
Relevant Pilot(s)	1.3, 1.4, 3.2, 3.3[LC1], … (ALL)				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 2: Knowledge Exchange Mechanisms Objective 4: Benchmarking Mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm enabler dashboards 6. Performance evaluation of Decision Support Systems 16. Mechanical weed control using hyperspectral camera and continuous crop data logging [LC2]				
Involved stakeholders/actors	Solution providers				
Prerequisite(s)	The required degree of data (e.g., image) quality has been defined in the respective pilot/use case.				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ATOS, ICCS				
Status	Proposed				

Comments/Remarks	

Requirement ID	DK5.21	**Version**	0.1	**Last Update Date**	06/12/2019
Title	Data provenance to ensure fused data quality				
Description	Fusion and data quality components should be able to track both the origin and the route of the data within the processing chain to monitor that the fused data maintain their quality and that no corrupted (or inconsistent) data are fused which would lower the quality. The result should be that data maintain and enhance their quality along the fusion process. This requirement ensures the high quality of the services provided by DEMETER by verifying the quality of the data used in the process.				
Addressed by Enabler(s) / Module(s)	Data Quality Assessment as-a-service Embeddable Data Quality Checks				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 9. Secure Agricultural data sharing services				
Involved stakeholders/actors	ICT and technological providers				
Prerequisite(s)					
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS, ATOS				
Status	Proposed				
Comments/Remarks					

Requirement ID	DK5.22	Version	0.1	Last Update Date	05/12/2019
Title	Data analysis and process actions in a timely manner				
Description	DEMETER needs to provide initial processing, merged data, quality assessment and data aggregation, in order to analyze and process actions in a timely manner.				
Addressed by Enabler(s) / Module(s)	Data Quality Assessment Non-functional requirement				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	O1: Analyze, adopt, enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.				
Relevant Innovation(s)	1. Agriculture Interoperability Space 5. Farm enabler dashboards				
Involved stakeholders/actors	Farmers, solution providers, semantic technologies experts.				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS, FhG.FIT, FhG.IESE, ROT				
Status	Proposed				
Comments/Remarks					

Requirement ID	DK5.23	Version	0.1	Last Update Date	31/01/2020
Title	Ensure quality of data streams				

Description	Data fusion and data quality components should help to understand the quality of continuous data streams in order to employ a system which algorithmically ensures high quality of continuous data streams (2.1)
Addressed by Enabler(s) / Module(s)	Data Quality Assessment as-a-service
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.3
Relevant Objective(s)	O1: Analyze, adopt, enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.
Relevant Innovation(s)	1. Agriculture Interoperability Space
Involved stakeholders/actors	Farmers, solution providers, ...
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS, ROT
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.25	**Version**	0.1	**Last Update Date**	31/01/2020
Title	Involvement of domain experts and stakeholders				
Description	Domain experts should be involved to understand already known data quality issues, i.e. collect feedback from stakeholders to identify and document their data quality needs. This will allow us to define specific data quality models for their context/needs. Furthermore, this stakeholder involvement (domain expertise) is required in order to evaluate the data quality results.				

Addressed by Enabler(s) / Module(s)	Data Quality Assessment as-a-service
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.3
Relevant Objective(s)	O1: Analyze, adopt, enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.
Relevant Innovation(s)	1. Agriculture Interoperability Space
Involved stakeholders/actors	Farmers, solution providers, ...
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Medium
Identified by Partner(s)	FhG.IESE
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.26	**Version**	0.1	**Last Update Date**	31/01/2020
Title	Access to (real) data and metadata				
Description	Access to data and metadata have to be provided in order to be able to analyze the data quality. There should be real data available in order to evaluate the developed approach/analysis technologies with real data. In addition, consistent/valid data should be also available in order to improve the appropriate data analysis. This data itself (especially the data type) as well as the metadata should be stable at least within a pilot.				
Addressed by Enabler(s) / Module(s)	Data Quality Assessment as-a-service				

Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.3
Relevant Objective(s)	O1: Analyze, adopt, enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.
Relevant Innovation(s)	1. Agriculture Interoperability Space
Involved stakeholders/actors	Farmers, solution providers, …
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Medium
Identified by Partner(s)	FhG.IESE
Status	Proposed
Comments/Remarks	

Requirement ID	DK5.29	**Version**	0.1	**Last Update Date**	31/01/2020
Title	Easily understandable and machine-executable data quality metrics				
Description	On the pilot side, data quality metrics should be documented in a clear, unambiguous way. On the technical side, it may be necessary to reduce complex data quality metrics (e.g., metrics that require taking into account multiple data sources or complex domain knowledge) to simpler approximations. At least for simple metrics, tools should support domain experts in writing down their quality metrics in a way that makes them immediately machine-executable.				
Addressed by Enabler(s) / Module(s)	Data Quality Assessment as-a-service Embeddable Data Quality Checks				
Relevant Pilot(s)	**ALL**				

Relevant Task(s)	T2.3
Relevant Objective(s)	O1: Analyze, adopt, enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms.
Relevant Innovation(s)	1. Agriculture Interoperability Space
Involved stakeholders/actors	Farmers, solution providers, …
Prerequisite(s)	None
Type	Functional
Priority Level	Low
Identified by Partner(s)	FhG.FIT
Status	Proposed
Comments/Remarks	

A.1.4. **Data Analytics & Machine Learning Requirements**

Requirement ID	DK6.1	**Version**	0.2b	**Last Update Date**	24/01/2019
Title	Data analytics support for data in various formats				
Description	Achieving interoperability in the agriculture domain is the utmost goal of DEMETER activities. Since it is necessary to integrate data from numerous sources of varying types, the data analytics needs to support data in several different formats from the deployed modules. Processing numerical data for predictive analytics, using text for data log automation, analyzing annotated satellite and UAV imagery, e.g. for disease detection on plants and crops, are just some core functionalities that are necessary on DEMETER. Regarding structured data, DEMETER should support parsing, serializing and deserializing to be given as input to analytics DEMETER components or to external services that rely on				

	standardized data formats as achieved by serializing these to established formats like JSON, XML and CSV. In order to allow data analytics processes to function on datasets comprised of structured JSON, XML and CSV data DEMETER shall support parsing of incoming and outgoing structured datasets as well as deserialization and serialization in order to support internal as well as externally located data analytics services.
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision Data wrappers/translators as input for the analytics enablers Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology
Involved stakeholders/actors	Farmers, ICT and technological providers
Prerequisite(s)	DK1.1
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS, m2Xpert
Status	Proposed+reviewed+updated
Comments/Remarks	

Requirement ID	DK6.3	**Version**	0.1	**Last Update Date**	11/12/2019

Title	Data analytics should provide appropriate input for the Decision Support Systems
Description	Data analytics are the final step of data processing. The aim of this requirement is that DEMETER provide the desired input to the Decision Support Systems in accordance with the specific use-case requirements, so as to achieve the ultimate goal of providing the stakeholders with the necessary tools and results to facilitate and enhance their decision-making processes. This should be made possible by introducing artificial intelligence in several applicable areas and deploying the appropriate algorithms to build actionable intelligence in the agri-food sector.
Addressed by Enabler(s) / Module(s)	AIM-compliant Serving Data Quality Assessment as-a-service Pattern Extraction with Computer Vision Irrigation analytics enabler fertilizer analytics enabler DSS Integration Analytics API Moisture estimation Plant Water Status Estimation Data analytics for weather forecast
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 10. Agri-food Decision support services based on SOA
Involved stakeholders/actors	Farmers, ICT and technological providers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS

Status	Proposed
Comments/Remarks	

Requirement ID	DK6.6	**Version**	0.2	**Last Update Date**	03/12/2019
Title	Avoid analysis bias				
Description	Data analytics should avoid bias from (1) technical and (2) methodological perspective. Ad 1.) analysis approaches need to be checked against bias, e.g., prioritization of certain aspects in the data. Particular emphasis should be put on handling of outliers, over- / underfitting, confounding variables Ad 2.) Analysis shall be aware of selection bias, confirmation bias, interpretation bias, prediction bias, information bias, fishing for results.				
Addressed by Enabler(s) / Module(s)	Model Management				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models				
Relevant Innovation(s)	10. Agri-food Decision support services based on SOA 6. Performance Evaluation of Decision Support Systems				
Involved stakeholders/actors	Farmers, solution providers, …				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ATOS, Fraunhofer				
Status	Proposed + reviewed				

Comments/Remarks	

Requirement ID	DK6.8	**Version**	0.2	**Last Update Date**	12/4/2021
Title	Support for Exploratory Numerical Data Analysis				
Description	DEMETER Data Analytics must support methods for exploratory data analysis (EDA) on given data or query sets independent of the deployed software solution. This can either be achieved by granting specific secure access to the DEMETER data storage layer based on a distinct eligible role or group membership by one or several ways that ease the interoperability: developing secure API methods that allow access to the required data, allowing usage of a command line tool, allowing interactive tools like e.g. Jupyter Notebook or by implementing an interactive analytics frontend: ● Standard statistical methods applicable on numerical datasets (standard deviation, absolute deviation variance, mean, median, range) ● Correlation analysis of multiple features of a given numerical data or query set with different methods (e.g. Pearson, Spearman)				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 4: Establish a benchmarking mechanism				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Technology providers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	m2Xpert				

Status	Proposed + reviewed
Comments/Remarks	

Requirement ID	DK6.9	**Version**	0.2b	**Last Update Date**	12/4/2021
Title	Support for Semantic Cross-Referencing				
Description	Analytical interoperability must be enabled by the DEMETER Data Analytics system by allowing for semantic cross-referencing. DEMETER Data Analytics shall enable actors to transparently compare semantically similar data- or query set features/fields between analysis of business domains that do not share the exact same namespace but, in reality, share comparable concepts (e.g. analysis on irrigation efficiency in the arable crop domain using a different namespace than the one used in the same analytical task within the fruit growing domain). To this end, it should support the use of data from existing ontologies.				
Addressed by Enabler(s) / Module(s)	AIM-compliant Serving AIM Metadata Schema				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Technology providers				
Prerequisite(s)	DK1.1 in particular (DK1.x in general)				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	m2Xpert				
Status	Proposed + reviewed + updated				

Comments/Remarks	

Requirement ID	DK6.10	**Version**	0.2	**Last Update Date**	09/12/2019
Title	Statistical Model Storage and Modification				
Description	DEMETER applications that rely on machine learning models must ensure not only the storage of the models deployed for predictions in the project solution but also the mechanisms to allow the update of the deployed models (regardless the modality, from canary release to a Big-Bang deployment).				
Addressed by Enabler(s) / Module(s)	Model Management Moisture estimation Plant Water Status Estimation Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 4: Establish a benchmarking mechanism				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Technology providers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	m2Xpert				
Status	Proposed + reviewed + updated				
Comments/Remarks					

Requirement ID	DK6.11	Version	0.2	Last Update Date	05/12/2019
Title	Ethical data analytics				
Description	Data analytics should follow principles of trustworthy and ethical AI. I.e. follow the recommendations from the High-Level Experts Group on AI fostered by the EU and the AI Alliance (Ethics Guidelines for Trustworthy AI). It should also follow the Ethical Observatory set up by the AI4EU project. By means of this requirement, the models generated should take into account the ethical principles proposed for AI systems, looking for characteristics such as fairness, explicability, etc.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3 (also T2.1 and T2.4)				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	8: Unified agriculture ontology 9: Secure Agricultural data sharing services 11: Data integration across the entire dairy supply chain 14: Smart fruit pesticides management 15: Open AKIS for irrigated crops 17: Water Management Model and Coordination Broker				
Involved stakeholders/actors	Solution providers, Technical partners, AI developers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ATOS				
Status	Proposed+reviewed				
Comments/Remarks	This is a cross-cutting issue that goes beyond WP2 and should be followed in any AI-related task in other WPs				

Requirement ID	DK6.12	Version	0.2	Last Update Date	05/12/2019
Title	Dataset should facilitate the generation or acquisition of datasets for training ML models.				

Description	DEMETER should facilitate the use or generation of labelled datasets for training supervised ML and DL models. To this extent DEMETER should: • Make an extensive search of training datasets in the literature and among project partners in the fields used by the pilots and make them available. In case there are no existing datasets for training, DEMETER should provide the means to easily labelling the data provided by the pilots. This may entail the use of crowdsourcing tools (i.e. Amazon Mechanical Turk) in case the labelling can be done by non-experienced users, or other frameworks that might ease the process of labelling by the project partners of farmers.
Addressed by Enabler(s) / Module(s)	
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	4: Earth Observation data service 8: Unified agriculture ontology 11: Data integration across the entire dairy supply chain 14: Smart fruit pesticides management 15: Open AKIS for irrigated crops 17: Water Management Model and Coordination Broker
Involved stakeholders/actors	Farmers, solution providers, technical partners, crowd workers.
Prerequisite(s)	Using supervised learning techniques
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ATOS
Status	Proposed + reviewed
Comments/Remarks	Labeling datasets for training purposes can be a manual and time-consuming task that shouldn't be underestimated. Therefore,

	searching existing labelled data and providing a minimal set of labelled datasets valid to train the models proposed by the pilots could be a good idea to start with

Requirement ID	DK6.13	**Version**	0.2	**Last Update Date**	12/12/2019
Title	Accurate, consistent and timely analytics				
Description	The ultimate aim of DEMETER is to empower farmers and demonstrate the capabilities that technology and digitalization can bring to agrifood domain through the use of responsive, intelligent Decision Support Systems. Data analytics should be given as input to the DSS, so the need for great accuracy and consistency is quite obvious. Also, time is a prerequisite for the responsiveness of those systems and thus, the analytics shall be given fast and in a timely manner as required by the various Decision Support tools and the system and of sufficiently high (the desired level) quality.				
Addressed by Enabler(s) / Module(s)	Data analysis for optimal pesticide usage Data Quality Assessment Irrigation requirements estimation. Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 6: Demonstrate the impact of digital innovation				
Relevant Innovation(s)	1. Agriculture Interoperability Space 10. Agrifood Decision support services based on SOA				
Involved stakeholders/actors	Farmers, ICT and technological providers				
Prerequisite(s)	None				
Type	Non-Functional				

Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed + reviewed
Comments/Remarks	

Requirement ID	DK6.14	**Version**	0.2	**Last Update Date**	12/12/2019
Title	Data Analytics Performance and Accuracy Requirement				
Description	Data analytics should not just be responsive, consistent and timely as described in DK6.13. It is extremely important that the system performs on high level of accuracy and achieves high percentage of True Positives (which would indicate for example an alert to take some action such as fertilizing the plants), as well as minimizing (or eliminating) the False Positives and False Negatives. E.g. if analytics show that plants are not healthy this should be spotted accurately and promptly and the DSS should take the appropriate action, without overfertilizing as a result of falsely reporting such an occurrence.				
Addressed by Enabler(s) / Module(s)	Data analysis for optimal pesticide usage Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 6: Demonstrate the impact of digital innovation				
Relevant Innovation(s)	1. Agriculture Interoperability Space 10. Agrifood Decision support services based on SOA				
Involved stakeholders/actors	Farmers, ICT and technological providers				

Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed + reviewed
Comments/Remarks	

Requirement ID	DK6.15	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Data Analytics should deal with data quality issue(s)				
Description	According to the assessment of the data quality, an appropriate analysis technique should be chosen to handle the identified issue(s). Analytics should have the ability to deal with (some) missing data or data from different sources which might be inconsistent. The goal is still to achieve useful results and provide the necessary data for the DSS.				
Addressed by Enabler(s) / Module(s)	Embeddable Data Quality Checks Model Management Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 1: Analyze, adopt, enhance information models Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space				
Involved stakeholders/actors	Solution providers				

Prerequisite(s)	DK5.11
Type	Non-Functional
Priority Level	Medium
Identified by Partner(s)	ICCS, Fraunhofer IESE
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.16	**Version**	0.2	**Last Update Date**	11/12/2019
Title	Auditable DSS and Analytics				
Description	DEMETER should provide mechanisms to label inappropriate or incorrect outputs from the analytics or DSS for the user. When the model(s) that offered that result allows it, the system should provide the capability to audit the decision process that led to that output (e.g. input variables, weights given to those inputs, etc.) especially if it gives incorrect advice to the user. Additionally, the possibility of adopting approaches similar to reinforcement learning could be evaluated in case it could be possible and wouldn't have a negative impact on the system performance.				
Addressed by Enabler(s) / Module(s)	Model Management				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models, Objective 3: Empower the farmer to gain control in the data-food-chain Objective 4: Establish a benchmarking mechanism				
Relevant Innovation(s)	6. Performance evaluation of Decision Support Systems 10. Agrifood Decision support services based on SOA 11. Data integration across the entire dairy supply chain				

Involved stakeholders/actors	Solution providers, Technology providers, Farmers
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed + reviewed
Comments/Remarks	This requirement will complement DK6.11 where it is a requirement that the AI be fair, explicable etc.

Requirement ID	DK6.17	**Version**	0.2	**Last Update Date**	23/01/2020
Title	Flexible architecture for analytics				
Description	It would be desirable to have modular systems: if we want to replace one analytics algorithm with something newer/better that whole system should be able to work after the change without (much) adaptation. The end user (e.g. farmer) should be oblivious to such changes.				
Addressed by Enabler(s) / Module(s)	AIM-compliant Serving Analytics API Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.3 (and T2.2)				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 3: Empower the farmer to gain control in the data-food-chain				

Relevant Innovation(s)	1. Agriculture Interoperability Space 10. Agrifood Decision support services based on SOA
Involved stakeholders/actors	Solution providers
Prerequisite(s)	Architecture design decisions (input /guidelines from WP3)
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS, Fraunhofer IESE
Status	Proposed + reviewed
Comments/Remarks	

Requirement ID	DK6.19	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Computer vision for pattern extraction				
Description	Computer vision techniques will be evaluated and deployed for generating advanced analytics over image data from various sources, aiming to extract patterns. Especially algorithms and methods from geo-spatial data (satellite, drones, ground images) for different purposes should be made available.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	1.1, 1.4, 3.3, 3.4				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Farmers, ICT and technological providers				

Prerequisite(s)	None
Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	ATOS
Status	Proposed + Reviewed
Comments/Remarks	

Requirement ID	DK6.20	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Analytics using EO and environmental data for plant monitoring				
Description	Sources like machinery and satellite data, as well as environmental information like weather and soil data are crucial for controlling and securing the health of the plants and ensuring the high quality of the production. The solution should integrate the aforementioned sources and combine them with EO data in order to analyze the interaction of parameters and provide valuable assistance to the farmers for the optimization of field and crop management.				
Addressed by Enabler(s) / Module(s)	Environmental monitoring to support Data analysis for optimal pesticide usage Moisture estimation Plant Water Status Estimation Data analytics for crop irrigation				
Relevant Pilot(s)	3.1-3.4				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Farmers, ICT and technological providers				

Prerequisite(s)	None
Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.21	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Cross-farm data processing				
Description	This requirement describes the deployment of systems for live support of agricultural processes and the connected supply chains based on autonomous documentation. This shall be based on functionalities like capturing high precision data, merging with data from other farms/machines and deriving required documentation parameters via data analytics and knowledge management techniques. Another relevant case is the establishment of a trust-based and compliant data market for agricultural enterprise data that sits between the owners and operators of agricultural data Clouds and the farmer, and that will include both a technical platform and advisory services that will ensure easy adoption of data and technology by farmers.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	2.2, 2.3, 3.2				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				

Involved stakeholders/actors	Farmers, ICT and technological providers
Prerequisite(s)	None
Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.22	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Environmental data analytics				
Description	Data analytics over environmental data to provide precise advice about cultivation. The proposed solutions will enable a more efficient usage of inputs such as water, energy, macro-nutrients, and pesticides. This will decrease the expenses and thus increase the profits of small farmers while also reducing their environmental impact.				
Addressed by Enabler(s) / Module(s)	Irrigation analytics enabler Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation				
Relevant Pilot(s)	3.2				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				

Involved stakeholders/actors	Farmers, ICT and technological providers
Prerequisite(s)	None
Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.23	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Analytics for optimized fertiliser use				
Description	Decision Support Systems shall be deployed to assist farmers on the optimization of irrigation and fertilization as well as of integrated pest management, through an on-line platform e.g. for olive farms and advisers. The proposed platform shall integrate software, sensors and open data sources to provide farmers and technicians a complete and efficient analytical assistance that contributes towards reduced costs and protecting the environment.				
Addressed by Enabler(s) / Module(s)	Fertilizer analytics enabler				
Relevant Pilot(s)	3.1, 3.2				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Farmers, ICT and technological providers				
Prerequisite(s)	None				

Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.24	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Analytics over UAV data for disease forecasting				
Description	The data analytics modules shall feed Decision Support Systems with the acquired and pre-processed UAV data so that they provide suitable advices to farmers concerning the health status of the crops. Especially, goal is to implement a system that is going to forecast on possible phytopathologies. The outputs above shall also contribute to the creation of product passports e.g. by certifying the quality of wine production.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	1.4, 3.4, 5.1				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Farmers, ICT and technological providers				
Prerequisite(s)	None				
Type	Pilot-specific				
Priority Level	Mandatory				

Identified by Partner(s)	ICCS
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.25	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Integrating heterogeneous data for automation purposes				
Description	The solution provided aims to increase production saving water and improving the automation of the irrigation zones through interoperable remote-control systems and robust management systems adapted to the conditions required by the irrigated agriculture. Another use case concludes at integrating Demeter project machinery data combined with crop- and field-specific info to analyse the interaction of parameters as described in aforementioned cases. Advice will be provided to farmers for the optimization of field management.				
Addressed by Enabler(s) / Module(s)	Irrigation requirements estimation				
Relevant Pilot(s)	1.1, 1.3, 2.2, 3.4				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Farmers, ICT and technological providers				
Prerequisite(s)	None				
Type	Pilot-specific				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS				

Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.26	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Analytics on animal properties for milk quality reasoning				
Description	DEMETER aims to create a dataflow dashboard of multiple uses like animal product accounting, settlement and payment, including decision support based on AI Machine learning from sensor data. Another case shall be animal welfare tracking. This can be implemented by an animal welfare scoring system with appropriate ICT tools to measure relevant parameters on a continuous, real time basis using stream data.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	4.1, 4.2, 4.3				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Farmers, ICT and technological providers				
Prerequisite(s)	None				
Type	Pilot-specific				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS				
Status	Proposed+reviewed				
Comments/Remarks					

Requirement ID	DK6.27	Version	0.2	Last Update Date	31/01/2020
Title	Analytics for optimal water quality				
Description	A Decision Support System platform shall be implemented by DEMETER in order to deploy the smart irrigation system needed by the farmers to optimize water quality control. This will be achieved by analytics over inputs concerning salinity and temperature. The use cases are mostly arable crops such as corn and maize.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	1.3, 1.4				
Relevant Task(s)	T2.3				
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Farmers, ICT and technological providers				
Prerequisite(s)	None				
Type	Pilot-specific				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS				
Status	Proposed+reviewed				
Comments/Remarks					

Requirement ID	DK6.28	Version	0.2	Last Update Date	31/01/2020
Title	Analytics for optimal usage of pesticides				

Description	This requirement aims at promoting technology, methods and IoT solutions to optimize precision farming practices e.g. of Mediterranean Woody Crops (Apple, Olive and Grape) to predict and support in decision making regarding the effective usage of pesticides. For example, a characteristic use case to be addressed is the provision of a set of tools to monitor and manage the Mediterranean fruit fly which is a dangerous pest with a wide range of distribution and host plants. Automatic capture traps and remote sensing technologies will be employed to predict infestation and support in taking decisions, such as optimal pesticide usage.
Addressed by Enabler(s) / Module(s)	Data analysis for optimal pesticide usage
Relevant Pilot(s)	3.1, 3.2, 3.3
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology
Involved stakeholders/actors	Farmers, ICT and technological providers
Prerequisite(s)	None
Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.29	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Linking of Analytics Data to the DSS to enable cross-sectoral interoperability of the platforms				

Description	This requirement aims to cover the needs for on- and post-farm activities management both from technical and business perspectives. Deployed in various fields like vineyards, cattle farm operations, apiculture and the poultry industry, the data analytics modules should reason over acquired sensor data and provide suitable advice to the farmers. Appropriate passports should also be created for the products produced which would then be made available to (among others) the supply chain stakeholders engaged.
Addressed by Enabler(s) / Module(s)	AIM-compliant Serving
Relevant Pilot(s)	5.1, 5.2, 5.3, 5.4
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 3: Empower the farmer, as a prosumer
Relevant Innovation(s)	1. Agriculture Interoperability Space 11. Data integration across the entire dairy supply chain
Involved stakeholders/actors	Farmers, ICT and technological providers, retailers, consumers
Prerequisite(s)	DK1.9, DK1.10, TBA
Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.30	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Data analytics over GPS and map data				
Description	This requirement encapsulates the need for a DSS for live support of agricultural processes and the connected supply chains based on				

	autonomous documentation. This will include capturing high precision data from geolocation and map systems, merging with data from other farms/ machines and deriving required documentation parameters via data analytics and knowledge management techniques.
Addressed by Enabler(s) / Module(s)	
Relevant Pilot(s)	2.2
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 3: Empower the farmer, as a prosumer
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology
Involved stakeholders/actors	Farmers, ICT and technological providers
Prerequisite(s)	TBA
Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	m2Xpert
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.31	**Version**	0.2	**Last Update Date**	31/01/2020
Title	Event triggering via real time analytics				
Description	Real time analytics using real time streaming data from wireless sensor networks, capable of triggering alarms/notifications/recommendations in order to improve farm operations and productivity. (all 4.X) [ICCS]				

	Event triggering is a fundamental service in order to empower the farmer. This requirement suggests a full dataflow dashboard with animal product accounting, settlement and payment, including decision support based on AI Machine learning from sensor data. This may be enriched with an animal welfare scoring system with appropriate ICT tools to measure relevant parameters on a continuous, real time basis and with prediction models of cow welfare and health based on analysis of streaming data from cow sensors. Given the fact that chicken sensor data be also available as input, the same process can be exploited in poultry, as well for benchmarking farm efficiency across the farms through IoT devices and AI based algorithms
Addressed by Enabler(s) / Module(s)	
Relevant Pilot(s)	4.1, 4.2, 4.3, 4.4
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms Objective 3: Empower the farmer, as a prosumer
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology
Involved stakeholders/actors	Farmers, ICT and technological providers, retailers, consumers
Prerequisite(s)	TBA
Type	Pilot-specific
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed+reviewed
Comments/Remarks	

Requirement ID	DK6.32	**Version**	0.2	**Last Update Date**	05/12/2019

Title	Data analytics for predictability of resource/input needs
Description	The analytics should enable the Decision support systems to give improved performance recommendations to agri-food stakeholders in order for them to have better control of their assets (i.e. prediction of their needs for supplies) and to regulate their productivity on a cost- and time-effective manner.
Addressed by Enabler(s) / Module(s)	Fertilizer analytics enabler
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.3
Relevant Objective(s)	Objective 2: Build knowledge exchange mechanisms
Relevant Innovation(s)	10. Agrifood Decision support services based on SOA
Involved stakeholders/actors	Farmers, ICT and technological providers
Prerequisite(s)	DK1.3, DK1.5, DK1.6
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Status	Proposed + reviewed
Comments/Remarks	

A.1.5. **Data Security & Privacy Requirements**

Requirement ID	DK7.1a	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Tools/application requirements: Lightweight messaging protocols that can also handle encryption				

Description	All tools and applications in Demeter must communicate using lightweight messaging protocols handling encryption.
Addressed by Enabler(s) / Module(s)	Communication Encryption Enabler
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.4, T3.4
Relevant Objective(s)	1, 2
Relevant Innovation(s)	1. Agriculture Interoperability Space 9. Secure Agricultural data sharing services
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.1b	**Version**	0.1	**Last Update Date**	09/12/2019
Title	Tools/application requirements: Secure way to handle a network of devices				
Description	All devices in the network must communicate in a secure way				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				

Relevant Task(s)	T2.4, T3.4
Relevant Objective(s)	2, 6
Relevant Innovation(s)	9
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	
Priority Level	Mandatory
Identified by Partner(s)	VICOM
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.1c	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Tools/application requirements: Secure transport layer for data exchange				
Description	Data exchange must be done using a secure transport.				
Addressed by Enabler(s) / Module(s)	Policy enforcement point (PEP-Proxy) Capability manager				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4				
Relevant Objective(s)	1				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				

Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, ROT, UMU, ODINS
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.1d	**Version**	0.2	**Last Update Date**	12/04/2021
Title	Tools/application requirements: Encryption should begin at sensor or embedded device levels.				
Description	Each installed sensor shall be equipped with lightweight cryptography in order to assure encryption on such low level, as well. If the sensor has not the needed capability to support lightweight cryptography the encryptions should be done as near of it as possible				
Addressed by Enabler(s) / Module(s)	Communication Encryption Enabler				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4, T3.4				
Relevant Objective(s)	O1: Analyze, adopt and enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms; O3: Empower the farmer, as a prosumer, to gain control in the data-food-chain.				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ROT				

Status	Proposed
Comments/Remarks	

Requirement ID	DK7.1e	**Version**	0.2	**Last Update Date**	12/04/2021
Title	Tools/application requirements: Secure way to handle resource constrained devices.				
Description	Resource constrained devices must communicate in a secure way. It is needed to include here the IoT broker when it is necessary to connect with passive or very limited sensors.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4, T3.4				
Relevant Objective(s)	O1: Analyze, adopt and enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms; O3: Empower the farmer, as a prosumer, to gain control in the data-food-chain.				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ROT				
Status	Proposed				
Comments/Remarks					

Requirement ID	DK7.1f	Version	0.2	Last Update Date	12/04/2021
Title	Tools/application requirements: Monitoring of intrusion detection.				
Description	Access control on sensors shall allow detection and monitoring on intrusions.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4, T3.4				
Relevant Objective(s)	O1: Analyze, adopt and enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms; O3: Empower the farmer, as a prosumer, to gain control in the data-food-chain.				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				
Type	Functional				
Priority Level	Mandatory				
Identified by Partner(s)	ROT				
Status	Proposed				
Comments/Remarks					

Requirement ID	DK7.1g	Version	0.1	Last Update Date	09/12/2019
Title	Tools/application requirements: Management of alarms in case of intrusion attempts (e.g. Jamming).				

Description	Every intrusion attempt must trigger an alarm mechanism.
Addressed by Enabler(s) / Module(s)	
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.4, T3.4
Relevant Objective(s)	O1: Analyze, adopt and enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms; O3: Empower the farmer, as a prosumer, to gain control in the data-food-chain.
Relevant Innovation(s)	9. Secure Agricultural data sharing services
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ROT
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.2a	**Version**	0.1	**Last Update Date**	09/12/2019
Title	Standards: Common formats and standards for information exchange that are also secure				
Description	Devices and all other entities that share information will need to do so in a standard way and using common formats, and these need to be secure.				

Addressed by Enabler(s) / Module(s)	Irrigation requirements estimation Moisture estimation Plant Water Status Estimation Data analytics for weather forecast Data analytics for crop irrigation
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.4, T3.4
Relevant Objective(s)	2, 6
Relevant Innovation(s)	9
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU, ODINS
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.2b	**Version**	0.1	**Last Update Date**	09/12/2019
Title	Standards: Formats and standards must allow cryptography.				
Description	Cryptography must be taken into consideration when setting the standards and formats of data.				
Addressed by Enabler(s) / Module(s)	Communication Encryption Enabler				
Relevant Pilot(s)	**ALL**				

Relevant Task(s)	T2.2, T2.4
Relevant Objective(s)	O1: Analyze, adopt and enhance existing Information Models in the agri-food sector; O2: Build knowledge exchange mechanisms; O3: Empower the farmer, as a prosumer, to gain control in the data-food-chain.
Relevant Innovation(s)	9. Secure Agricultural data sharing services
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	Functional
Priority Level	Mandatory
Identified by Partner(s)	ROT
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.2c	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Standards: A multi-hop communication routing should be improved.				
Description	Multi-hop routing can be more efficient if improved.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T3.4				
Relevant Objective(s)	2				
Relevant Innovation(s)	7. Cost and power-effective IoT data acquisition				

Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.3a	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Distributed, capability and attribute-based access control system				
Description	The Access Control based system for Demeter must support the facility of being distributed and have the be able to use attributes or capability-based descriptions in order to make access control decisions				
Addressed by Enabler(s) / Module(s)	Pattern Extraction with Computer Vision Policy enforcement point (PEP-Proxy) Capability manager Policy administration point (PAP) Policy decision point (PDP)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4, T3.4				
Relevant Objective(s)	1, 2				
Relevant Innovation(s)	1. Agriculture Interoperability Space 9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				

Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU, ODINS, TSSG
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.3b	**Version**	0.1	**Last Update Date**	09/12/2019
Title	Authentication and authorisation, traceability: Secure transport layer for authn/authz				
Description	The transport layer for all matters regarding authentication and authorisation must be secure.				
Addressed by Enabler(s) / Module(s)	Policy enforcement point (PEP-Proxy) Capability manager				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4, T3.4				
Relevant Objective(s)	2, 6				
Relevant Innovation(s)	9				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				
Type					
Priority Level	Mandatory				
Identified by Partner(s)	VICOM				
Status	Proposed				

Comments/Remarks	

Requirement ID	DK7.3c	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Policy language for defining the access to resources				
Description	A human readable language is needed to allow the flexible but easy description of access control policies by various stakeholders to their data and resources, ideally following an attributed based access control paradigm.				
Addressed by Enabler(s) / Module(s)	Policy administration point (PAP)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.2, T3.4				
Relevant Objective(s)	2				
Relevant Innovation(s)	1. Agriculture Interoperability Space 9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Developers, End Users				
Prerequisite(s)	Access Control solution				
Type	Non-Functional				
Priority Level	Mandatory				
Identified by Partner(s)	VICOM, UMU, ODINS, TSSG				
Status	Proposed				
Comments/Remarks					

Requirement ID	DK7.3d	**Version**	0.1	**Last Update Date**	11/12/2019

Title	Authentication and authorisation, traceability: Data handling policy language to set how requested data is handled and passed on.
Description	It is necessary to have a data handling policy language in order to set how to handle and pass requested data.
Addressed by Enabler(s) / Module(s)	Policy administration point (PAP) Policy decision point (PDP)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.4
Relevant Objective(s)	1
Relevant Innovation(s)	9. Secure Agricultural data sharing services
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.3e	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Authentication and authorisation, traceability: Define which users and devices will have access to what, and when and how (permissions and restrictions for each).				
Description	Permissions and restrictions have to be established for all users and devices.				

Addressed by Enabler(s) / Module(s)	Policy administration point (PAP) Policy decision point (PDP)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T3.5
Relevant Objective(s)	1
Relevant Innovation(s)	9. Secure Agricultural data sharing services
Involved stakeholders/actors	Administrators and developers
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU, ODINS
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.3f	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Authentication and authorisation, traceability: Appropriate traceability for heterogeneous datasets				
Description	It is necessary to establish suitable traceability for heterogeneous datasets.				
Addressed by Enabler(s) / Module(s)	Policy enforcement point (PEP-Proxy) Capability manager Policy administration point (PAP) Policy decision point (PDP)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4				

Relevant Objective(s)	1
Relevant Innovation(s)	9. Secure Agricultural data sharing services
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU, ODINS
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.3g	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Capability of the data owner to specify who can access, process and store its data				
Description	All data owners need to have full control over the processing, sharing and storage of their data, irrespective of where the data is being stored.				
Addressed by Enabler(s) / Module(s)	Policy enforcement point (PEP-Proxy) Capability manager Policy administration point (PAP) Policy decision point (PDP)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T3.5				
Relevant Objective(s)	1				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				

Involved stakeholders/actors	Administrators and developers, end users, data owners.
Prerequisite(s)	Attribute based access control
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU, ODINS, TSSG
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.4a	**Version**	0.1	**Last Update Date**	09/12/2019
Title	Content: Content encryption/decryption and encoding of data				
Description	All data handled must be encrypted.				
Addressed by Enabler(s) / Module(s)	Communication Encryption Enabler				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4, T3.4				
Relevant Objective(s)	2, 6				
Relevant Innovation(s)	9				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				
Type					
Priority Level	Mandatory				
Identified by Partner(s)	VICOM, ROT, UMU, ODINS				

Status	Proposed
Comments/Remarks	

Requirement ID	DK7.4b	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Content: Protect personal data.				
Description	Security in Demeter must protect all personal data in the platform.				
Addressed by Enabler(s) / Module(s)	Policy enforcement point (PEP-Proxy) Capability manager Policy administration point (PAP) Policy decision point (PDP)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4				
Relevant Objective(s)	1				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				
Type	Non-Functional				
Priority Level	Mandatory				
Identified by Partner(s)	VICOM, UMU, ODINS				
Status	Proposed				
Comments/Remarks					

Requirement ID	DK7.4c	**Version**	0.1	**Last Update Date**	11/12/2019

Title	Content: Protect sensitive data.
Description	Security in Demeter must protect sensitive data in the platform.
Addressed by Enabler(s) / Module(s)	Policy enforcement point (PEP-Proxy) Capability manager Policy administration point (PAP) Policy decision point (PDP)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.4
Relevant Objective(s)	1
Relevant Innovation(s)	9. Secure Agricultural data sharing services
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM, UMU, ODINS
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.5	**Version**	0.1	**Last Update Date**	09/12/2019
Title	Regulation requirements (signatures, storage, anonymization): Comply with GDPR technical requirements				
Description	All stored data must comply with existing regulations. This requirement will only focus on the technical aspects, not organizational.				
Addressed by Enabler(s) / Module(s)					

Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.4, T3.4
Relevant Objective(s)	2, 6
Relevant Innovation(s)	9
Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	
Priority Level	Mandatory
Identified by Partner(s)	VICOM
Status	Proposed
Comments/Remarks	Each measure should be done on a case-by-case basis.

Requirement ID	DK7.6	**Version**	0.1	**Last Update Date**	11/12/2019
Title	Regulation requirements (signatures, storage, anonymization): Perform Court-proof logging and audit logs.				
Description	Court-proof logging and audit logs must be performed for regulation requirements.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4				
Relevant Objective(s)	1				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				

Involved stakeholders/actors	Developers
Prerequisite(s)	None
Type	Non-Functional
Priority Level	Mandatory
Identified by Partner(s)	VICOM
Status	Proposed
Comments/Remarks	

Requirement ID	DK7.7	**Version**	0.2	**Last Update Date**	16/04/2021
Title	Preservation of data access rights.				
Description	Before any aggregation or fusion, the rights over the data used should be studied and appropriate anonymization or aggregation techniques should be applied. This functionality can't be automated and should be studied and implemented in a case-by-case procedure. The enabler should support encryption, steganography or sanitization techniques.				
Addressed by Enabler(s) / Module(s)					
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.4				
Relevant Objective(s)	1				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Developers				
Prerequisite(s)	None				
Type	Non-Functional				

Priority Level	Mandatory
Identified by Partner(s)	ATOS
Status	Proposed
Comments/Remarks	

A.2. Tables of components/enablers

Name (Section)	Data management (10.1)
Status	Finished/Deployed on DEMETER Cloud. Below are the internet addresses to reach the applications of the data management module: • https://bse.h2020-demeter-cloud.eu/api/BSE/ (DEH RRM - Resource Registry Management APIs) • https://bse.h2020-demeter-cloud.eu/api/BSE/ (BSE - Brokerage Service Environment APIs) • https://acs.bse.h2020-demeter-cloud.eu:5443/sign_up/ (ACS - Access Control Server)
Implementation	The general documentation of each data management software module is on the DEMETER project source code repository or Gitlab instance to the following address: https://gitlab.com/demeterproject/: • https://gitlab.com/demeterproject/wp3/demeterenablerhub/resourceregistrymanagement (DEH RRM - Resource Registry Management) • https://gitlab.com/demeterproject/wp3/bse (BSE - Brokerage Service Environment) • https://gitlab.com/demeterproject/wp3/se/enablers/authentication (ACS - Access Control Server – authentication and user information retrieval) • https://acs.bse.h2020-demeter-cloud.eu:3030/ (ACS - Access Control Server – authorisation) Each repository describes the architecture of the component, the list of features and therefore the endpoints that the API exposes to the outside, the requirements for installation and the docker instructions to install an instance of the software. The documentation for the APIs of the modules that make up the DEMETER data management system, the documentation can be reached at the following internet addresses: • https://deh.h2020-demeter-cloud.eu/swagger (DEH Resource Registry Management APIs) • https://bse.h2020-demeter-cloud.eu/api/swagger/ (BSE - Brokerage Service Environment) • https://gitlab.com/demeterproject/wp3/se (ACE - Access Control Server) The documentation on the DEH Resource Registry Management APIs usage in the form POSTMAN[34] Project for DEMETER developers can be reached at the following internet address: • https://gitlab.com/demeterproject/wp3/demeterenablerhub/resourceregistrymanagement/-/tree/master/postman_collections
Integration within DEMETER	*All pilots to integrate.*
Related requirement (s)	The requirements of the data management module correspond to the DK4 requirement class. These requirements have been collected as part of the definition of WP2 technical requirements and available both in this Deliverable (D2.4) and in the first version or D2.2

[34] https://www.postman.com/

Name (Section)	Data Preparation & Integration (DPI) Enabler (10.2)
Status	In progress
Implementation	Service live instance • https://dpi-enabler-demeter.apps.paas-dev.psnc.pl/api/swagger/ Documentation: CLI: * https://git.man.poznan.pl/stash/projects/DEM/repos/pipelines/browse/README.md Web service: • https://docs.psnc.pl/display/DEM/The+architecture+of+dpi-enabler Source: CLI: • https://git.man.poznan.pl/stash/projects/DEM/repos/pipelines/browse
Integration within DEMETER	Enabler used by Pilot 2.4, potentially other pilots too Pilot 2.3, Pilot 5.3. The enabler is used as a service to generate AIM data that can be used in the pilots. It is not required direct integration with pilot services.
Related requirement(s)	3.2, 3.3, 3.4, 3.5, 3.7, 4.1, 4.3, 4.15, 4.22

Name (Section)	Data Quality Assessment Service of Tabular Data Service (11.1.1)
Status	In progress
Implementation	https://gitlab.com/demeterproject/wp2/dataquality/data-quality-assessment/-/tree/master/dqa_for_machinery_data
Integration within DEMETER	• Pilot specific implementations: 2.1, 2.2. • Status of integration readiness: In progress.
Related requirement(s)	5.13, 5.14, 5.17, 5.25, 5.26, 6.3

Name (Section)	Data Quality Assessment – Linked Data (11.1.2)
Status	In progress
Implementation	https://gitlab.com/demeterproject/wp2/dataquality/data-quality-notebooks/
Integration within DEMETER	Early in Integration: Data Quality Assessment Service Enabler
Related requirement(s)	5.13, 5.17, 5.21, 5.23, 5.26, 5.29

Name (Section)	Embeddable Data Quality Checks (11.1.3)
Status	Finished/Deployed
Implementation	https://gitlab.com/demeterproject/wp2/dataquality/data-quality-api
Integration within DEMETER	Almost done/Testing: AIM-compliant Serving Enabler, Model Management Enabler
Related requirement(s)	5.17, 5.21, 5.29, 6.15

Name (Section)	Optimal Fertilizer Usage (for arable crops) (11.2.1)
Status	Almost done/Testing
Implementation	https://gitlab.com/demeterproject/wp2/dataanalytics/optimal-fertilizer-usage
Integration within DEMETER	Pilot 1.3 Almost done/Testing
Related requirements	DK5.25, DK6.3, DK6.6

Name (Section)	Maize irrigation APP (11.2.2)
Status	Almost done/Testing
Implementation	To be uploaded soon, when finalized/completed
Integration within DEMETER	Pilot 1.3 Almost done/Testing
Related requirements	DK6.20, DK6.22, DK6.27

Name (Section)	Predictive Model Training web service (11.2.3)
Status	Almost done/Testing
Implementation	https://gitlab.com/demeterproject/wp2/dataanalytics/predictionapi
Integration within DEMETER	The component has been used as a basis of the Olive Phenology Prediction Service of WP4. Integration is in the status Fully Integrated. The Testing status reported above for the component refers to its extendibility to further use cases/applications.
Related requirement(s)	- DK6.3 - DK6.6 - DK6.9 PARTIALLY through the possibility of implementing a layer dealing with the AIM JSON-LD format on top of the component, as is done e.g. in the Olive Phenology Prediction Service of WP4 (see notes in section 11.2.3).

Name (Section)	Pattern Extraction with Computer Vision (11.2.4)
Status	In progress.
Implementation	https://gitlab.com/demeterproject/wp2/dataanalytics/pattern-extraction-with-computer-vision
Integration within DEMETER	<The pattern extraction with computer vision component has been integrated with MLFLOW and is currently being integrated with the DEH, ACS and BSE. In order to do so, some infrastructure's dependencies are being solved with the help of Tragsa.
Related requirement(s)	DK6.3 DK6.1, as it is based on the use of image files DK5.4 The AIM model used by the component has been updated.

Name (Section)	Data analytics for optimal pesticide usage (11.2.5)
Status	Finished/Deployed
Implementation	https://gitlab.com/demeterproject/wp2/dataanalytics/data-analysis-for-optimal-pesticide-usage
Integration within DEMETER	(1) Integration in Pilot 5.1 (2) Integration readiness: Almost done/Testing
Related requirement(s)	6.13, 6.14, 6.20, 6.22, 6.28

Name (Section)	Weather forecast (11.2.6)
Status	Almost done & testing
Implementation	https://gitlab.com/demeterproject/wp2/dataanalytics/meteoforecast This version will be updated by next version to use data fusion with multiple data sources.
Integration with DEMETER	The functionality of this component is used by DSS water balance model and predictiveET0 (WP4 - 4.B.3 Irrigation Requirements Estimation). In the first case it is used to retrieve rainwater forecast but in the last case it is used in current version as an internal R library though in next version it will be used as a DEE component. Also, as any other DEE component, it will be integrated with ACS, DEH and BSE.
Related requirement(s)	3.1, 3.2, 3.4, 3.7, 4.3, 4.5, 5.1, 5.2, 5.3, 5.4, 5.19, 6.1, 6.3, 6.13, 6.14, 6.15, 6.17, 6.22, 7.2a

Name (Section)	Crop Irrigation water analysis based on ETo-Kc (11.2.7)
Status	Almost done & testing
Implementation	https://gitlab.com/demeterproject/wp2/dataanalytics/crop-irrigation-based-on-eto-kc
Integration with DEMETER	This component is used by DEE component DSS water balance model (WP4 - 4.B.3 Irrigation Requirements Estimation). Also, as any other DEE component, will be integrated with ACS, DEH and BSE.
Related requirement(s)	3.1, 3.2, 3.4, 3.7, 4.3, 4.5, 5.1, 5.2, 5.3, 5.4, 5.19, 6.1, 6.3, 6.10, 6.13, 6.14, 6.15, 6.17, 6.20, 6.22, 7.2a

Name (Section)	Model Management (11.3.1)
Status	Finished/Deployed
Implementation	https://gitlab.com/demeterproject/wp2/dataanalytics/model-management
Integration within DEMETER	In Progress/Almost done/Testing: 1.1, 1.2, 1.3 Not started/Early in Integration: 2.1, 3.1, 3.3, 3.4, 4.2, 4.3, 4.4, 5.1, 5.2
Related requirement(s)	6.6, 6.10, 6.11, 6.14, 6.15, 6.16

Name (Section)	AIM-compliant Serving (11.3.2)
Status	Almost done/Testing
Implementation	https://gitlab.com/demeterproject/wp2/dataanalytics/analytics-api
Integration within DEMETER	In Progress/Almost done/Testing: 1.1, 1.2, 1.3 Not started/Early in Integration: 2.1, 3.1, 3.2, 3.3, 3.4, 4.1, 4.2, 4.3, 4.4, 5.1, 5.2, 5.3, 5.4
Related requirement(s)	5.4, 5.7, 6.3, 6.9, 6.17, 6.29

Name (Section)	Authentication Enabler
Status	Almost done/Testing
Implementation	https://gitlab.com/demeterproject/wp3/se/enablers/authentication
Integration within DEMETER	Authentication Component (IdM)
Related requirement(s)	7.2, 7.3, 7.4, 7.5, 7.6, 7.7

Name (Section)	Authentication Component (IdM)
Status	Almost done/Testing
Implementation	https://gitlab.com/demeterproject/wp3/se/components/acs/-/blob/master/authentication/README.md
Integration within DEMETER	The pilots or other enablers/components where some integration took place are: • **Traceability Component:** This integration is almost done/Testing. • **Capability Manager:** This integration is in progress.
Related requirement(s)	7.2, 7.3, 7.4, 7.5, 7.6, 7.7

Name (Section)	Policy Administration Point (12.2.1)
Status	Finished/Deployed
Implementation	https://gitlab.com/demeterproject/wp3/se/components/acs/-/tree/master/authorisation/XACML_PAP_PDP
Integration with DEMETER	This component is deployed in Demeter Cloud infrastructure. It is an administrative component to define access authorisation policies. In this sense, it has not directly interaction with other authentication/authorisation components. Regarding the rest of Demeter, the integration of traceability has Early in Integration status.
Related requirement(s)	7.3a, 7.3b, 7.3c, 7.3d, 7.3e, 7.3g

Name (Section)	Policy Decision Point (12.2.2)
Status	Finished/Deployed/Testing
Implementation	https://gitlab.com/demeterproject/wp3/se/components/acs/-/tree/master/authorisation/XACML_PAP_PDP
Integration with DEMETER	This component is deployed in Demeter Cloud infrastructure. It is accessible from Capability Manager component. The status of this integration is Fully Integrated/Testing. Regarding the integration of traceability, the status is Early in Integration.
Related requirement(s)	7.3a, 7.3b, 7.3d

Name (Section)	Capability Manager (12.2.3)
Status	Finished/Deployed/Testing
Implementation	https://gitlab.com/demeterproject/wp3/se/components/acs/-/tree/master/authorisation/Py_CapabilityManagerWebService
Integration with DEMETER	This component is deployed in Demeter Cloud infrastructure. It is accessible from DEH and BSE. The status of both integrations is Fully Integrated/Testing. Capability Manager accesses to IdM-Keyrock component (authenticantion) and Policy Decision Point (PD). The status of both integrations is Fully Integrated/Testing. Regarding the integration of traceability, the status is Early in Integration.
Related requirement(s)	7.1b, 7.1c, 7.3a, 7.3b

Name (Section)	Policy Enforcement Point (12.2.4)
Status	Finished/Deployed/Testing
Implementation	https://gitlab.com/demeterproject/wp3/se/components/pep_proxy
Integration with DEMETER	This component is deployed in Demeter Cloud infrastructure. It is accessible from DEH and BSE. The status of both integrations is Fully Integrated/Testing. Regarding the integration of traceability, the status is Early in Integration.
Related requirement(s)	7.1b, 7.1c, 7.3a, 7.3b, 7.4a

Name (Section)	Trazability DLt and Trazability Agent (12.3)
Status	Almost done/Testing
Implementation	
Integration within DEMETER	The pilots or other enablers/components where some integration took place are: • **KeyRock:** This integration is almost done/Testing. • **Capability Manager:** This integration is in progress.
Related requirement(s)	Trazability and Governance

Name (Section)	Communication Encryption Enabler (12.4)
Status	Stable version finished and uploaded; Continuous development.
Implementation	https://gitlab.com/demeterproject/wp2/secur/encryption
Integration within DEMETER	This enabler is fully integrated with the Transaction Manager component in Pilot 4.2. The Transaction Manager component is written in Python3 and the enabler is loaded as a shared (dynamic) library. All the component is containerized via Docker.
Related requirement(s)	7.1a, 7.1e, 7.4a, 7.4b, 7.4c