D2.2 DEMETER Data and Knowledge extraction tools

Contents

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 as Internet of Things, Artificial Intelligence, Big Data, Decision Support, Benchmarking, Earth Observation, etc., in order 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. The adoption of such technologies will facilitate and speed-up the deployment of interoperable data driven smart farming solution providing decision support and control systems for the agricultural sector that empower farmers to take better decisions, and that allow them to harness the full value of their own data and knowledge as well as those shared with others. 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, and to promote the targeted technological, business, adoption and socio-economic impacts, 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, as well as the decision making of farmers and other stakeholders. These requirements are handled by the Data & Knowledge (DK) enablers, discussed in this deliverable, and which 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: Data Preparation & Integration, Data Management, and Data Analytics & Knowledge Extraction, which are presented in this deliverable. 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.

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. Hence, it is a core element of the Data Preparation & Integration enabler, which transforms data into AIM format or translates queries using AIM terms. Similarly, the data generated by the enablers, e.g., knowledge extracted, data quality information, is also represented according to AIM model.

The design and implementation of the different enablers is based on an exhaustive analysis of the state of the art, which is presented in the first part of this document. This analysis includes relevant approaches, methods and techniques related to the DK facilities, but also a review of the existing tools, services and applications supporting them, and which could be re-used/leveraged in the implementation. After that, the requirements driving the implementation of these enablers are

presented 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, and Data Security & Privacy. After that, 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, first a description of the approach is provided, followed by a presentation of the design, including UML diagrams, and the ongoing implementation work. Finally, the document concludes presenting also future work towards the final version of these enablers.

Note that this document marks the initial release of the data and knowledge extraction tools, which will be followed by a second and final release in one year.

2 Acronyms

A.I.S.	Agricultural Interoperability Space
AA	Attribute Authority
ABAC	Attribute-Based Access Control
ABE	Attribute-Based Encryption
AES	Advanced Encryption Standard
AGRIS	International System for Agricultural Science and Technology
AIM	Agriculture Information Model
API	Application Program Interface
BSE	Brokerage Service Environment
CDH	Cloudera Distribution Including Apache Hadoop
CLI	Command Line Interface
CM	Capability Manager
CNN	Convolutional neural network
CP-ABE	Ciphertext-policy Attribute-based Encryption
CSV	Comma-separated values
CT	Capability Token
DAC	Discretionary Access Control
DCapBAC	Distributed Capability-Based Access Control
DEE	Demeter Enhanced Entity
DEH	DEH Enabler HUB
DEMETER RA	DEMETER Reference Architecture
DES	Data Encryption Standard
DLM	Data Lifecycle Management
DQV	W3C Data Quality Vocabulary
DSA	Digital Signature Algorithm
ECDH	Elliptic-curve Diffie–Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EDOAL	Expressive and Declarative Ontology Alignment Language
ETL	Extract, transform, load
FAO	Food and Agriculture Organization
FIPA	Foundation for Intelligent Physical Agents
FOAF	Friend of a Friend
FOODIE	Farm-oriented Open Data In Europe
GATE	General Architecture For Text Engineering

GAV	Global As View
GDPR	General Data Protection Regulation
GML	Geography Markup Language
GQM	Goal Question Metric
GUI	Graphical User Interface
HDFS	Hadoop Distributed File System
HDP	Hortonworks Data Platform
HMAC-MD5	Hash-based Message Authentication Code
HTTP	HyperText Transfer Protocol
IBE	Identity-Based Encryption
IDEA	International Data Encryption Algorithm
IdM	Identity Management
ILM	Information Lifecycle Management
IoT	Internet of Things
IRI	Internationalized Resource Identifier
ISO	International Organization for Standardization
J2EE	Java 2 Platform Enterprise Edition
JSON-LD	JavaScript Object Notation for Linked Data
JWT	JSON WebToken
KML	Keyhole Markup Language
LAV	Local As View
LOD	Linked Open Data
LoRaWAN	Long Range Wide Area Network
MAC	Mandatory Access Control
ML	Machine Learning
NALT	National Agricultural Library Thesaurus
NI-ZKP	Non-Interactive Zero Knowledge Proof
OAuth	Open Authorization
OGC	Open Geospatial Consortium
OIDC	OpenID Connect
OMG	Object Management Group
ORDBMS	Object-Relational Database Management System
OWL	Web Ontology Language
PAP	Policy Administration Point
PDP	Policy Decision Point
PKC	Public Key Cryptography
PLF	Precision Livestock Farming
POI	Point of Interest
PSK	Pre-shared key
QaaS	Query as a Service
QoS	Quality of Service
R2RML	RDB to RDF Mapping Language
RBAC	Role-Based Access Control
RC4	Rivest Cipher 4
RDBMS	Relational Database Management System
RDF	Resource Description Framework
RDF2Vec	RDF to vector
RDFS	Resource Description Framework Schema

REST	Representational State Transfer
RML	RDF Mapping Language
RSA	Rivest–Shamir–Adleman
SAML	Security Assertion Markup Language
SANSA	Semantic Analytics Stack
SASMINT	Semi-Automatic Schema Matching and Integration
SDML	Signed Document Markup Language
SIOC	Semantically-Interlinked Online Communities
SKC	Symmetric Key Cryptography
SKOS	Simple Knowledge Organization System
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
SOSA	Semantic Sensor Network Ontology
SPARQL	Simple Protocol and Rdf Query Language
SQL	Structured Query Language
SQuaRE	Software Product Quality Requirements and Evaluation
SRP	Secure Remote Password protocol
SSL	Secure Sockets Layer
SVR	Support Vector Machine Regression
SWRL	Semantic Web Rule Language
TLS	Transport Layer Security
Triple DES	Triple Data Encryption Algorithm
UML	Unified Modeling Language
URI	Uniform Resource Identifier
VGI	Volunteered Geographical Information
W3C	World Wide Web Consortium
WKT	Well Known Text
WMS	Web Map Service
XACML	Extensible Access Control Markup Language
ZBAC	AuthoriZation-Based Access Control

3 List of Authors

Company	Author
PSNC	Raul Palma
PSNC	Soumya Brahma
ICCS	Ioanna Rousaki
ICCS	Ioannis Vetsikas
ICCS	George Routis
ICCS	Marios Paraskevopoulos
Fraunhofer FIT	Till Doehmen
Fraunhofer FIT	Md. Rezaul Karim
Engineering	Antonio Caruso
VicomTech	Oscar Miguel Hurtado
Fraunhofer IESE	Anna Maria Vollmer
Fraunhofer IESE	Patricia Kelbert
TECNALIA	Sonia Bilbao
TECNALIA	Belén Martínez
TECNALIA	Alejandro Rodríguez

TECNALIA	Fernando Jorge
ATOS	Jesus Benedicto Cirujeda
ATOS	Jesus Martinez Gadea
ATOS	Tomas Pariente Lobo
ATOS	Sergio Salmeron Majadas
INTRA	I. Oikonomidis
INTRA	A. Poulakidas
ROT	Lorenzo Bortoloni
OdinS	Juan Antonio Martinez
UMU	Manuel Mora

4 Introduction

This deliverable presents the first release of the "DEMETER Data and Knowledge extraction tools". The work presented in this document is the output of tasks 2.2 (Data Management and Integration), 2.3 (Targeted data fusion, analytics and knowledge extraction) and 2.4 (Data Protection, Privacy, Traceability and Governance Management) of the Work Package 2. The combined results presented in this document realize, on the one hand, the implementation of the Data & Knowledge (DK) enablers, which are part of the advanced enablers in DEMETER, and which use and rely on 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 AIM described in detail in D2.1. On the other hand, this document also presents 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 D3.1. More specifically, the rest of the document is structured as follows:

Section 5 provides an analysis of 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, including Data Management and Integration, covering methods and models for data access policies/rights, data integration approaches in agriculture, data enrichment, data models alignments and data exploration; Data Analytics and Knowledge Extraction, covering data processing analytics tools for big data and targeted for the agriculture sector, data quality assessment and cleaning; Data Protection, Privacy and Traceability, covering data provenance, authentication and authorization protocols, privacy and security by-design technologies, and risk analysis and estimation.

Section 6 provides an overview of the technical requirements extracted by Task 2.2, Task 2.3 and Task 2.4. This is a complete list regarding the capabilities/functionalities needs that DEMETER must or should deliver with respect to data integration, including semantic interoperability, data management, data quality & fusion, data analytics, including machine learning, data security & privacy.

Section 7 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 8 presents the data management components, including an introduction to the design/approach, covering also aspects of multi-tenancy, availability, scalability & QoS, and details

about the implementation realized via the Brokerage Service Environment and the DEMETER Enabler Hub, and complemented with some data management components in DEMETER-enhanced entities.

Section 9 presents the data preparation & integration components, including a discussion of DEMETER's approach for data integration using Linked Data as a federated layer, the design, state and work regarding the implementation of data preparation & integration pipelines, the underlying components and on-going work in the implementation of the enabler API.

Section 10 presents the data quality components, including discussion of the quality requirements, the design and on-going work regarding the implementation of data quality enabler API, and underlying facilities quality assessment of linked data, tabular data, and the aspects of provenance dealt by the enabler.

Section 11 presents the data fusion and analytics components, including the design of the enabler API, and the on-going work regarding the modules for targeted data analytics, targeted data fusion, training data and label acquisition, algorithm selection and feature engineering, FAIR analytics, model storage and DSS integration.

Section 12 presents 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 13 concludes the document presenting also future work towards the final version of the Data and Knowledge extraction tools (D2.3), while Section 14 provides the respective references used.

Additionally, the deliverable includes two Annexes. Annex A presents in tabular form the mapping of requirements presented in Section 6 with the components presented in the components Sections (Sections 8-12). Annex B includes the authentication endpoints documentation, including examples of http requests to the IdM Endpoints that provide authentication functionalities.

The enablers presented in this deliverable complement the deliverable D3.1 DEMETER Reference Architecture (Release 1). This work will be used in the following deliverables which follow:

- D3.2 DEMETER technology integration tools (June 2020)
- D4.2 Decision Enablers, Advisory Support Tools and DEMETER Stakeholder Open Collaboration Space (June 2020)
- D5.3 Testbed, deployment, system extensions and applications for pilot round 1 (July 2020)

The revised (final) version of the DEMETER Data and Knowledge extraction tools is planned for release on May 2021 and will be presented in D2.4.

5 Related Work

5.1 Data Management and Integration

This section presents the State of the Art regarding the management and access to data as well as the integration of data in the agri-food sector.

5.1.1 Open and interoperable data models

The scope of the idea “open and interoperable data integration model” is extremely challenging to comprehensively review, however it is possible to observe this through the lens of the directions of standards organizations such as W3C, OMG and the Open Geospatial Consortium. A 2016 survey [6] focused on the use of XML and Web Services and comparison to the “Semantic Web”. Evidently use of JSON based APIs has a much greater momentum over XML, however the foundation of the Semantic Web remains relevant, and JSON-LD (JSON Linked Data) is a way of implementing those foundations with today’s services. A recent OGC report increases the understanding of state-of-the-art for JSON, JSON Schema, and JSON for Linked Data (JSON-LD) technologies as applied to the encoding of an ISO 19109-conformant Unified Modelling Language (UML) application schema within the OGC community [7]. This report also references two other approaches relevant to data modelling such as UML and JSON-Schema. Other emergent technologies relevant to data model interoperability include GraphQL (as used by Facebook), which is gaining rapid popularity due to its ability to break the single fixed schema view of data and query interlinked data elements (graphs). Interoperability means the ability to connect related pieces of information into a knowledge graph and one possible characterization of the SotA that fits these trends is formal semantics describing data accessible via APIs and the role of controlled vocabularies in sharing the knowledge. The mix of technologies for the APIs varies, however the use of formal semantics using RDF and OWL has been remarkably stable for several decades. The technical pattern that is emerging is a strong formalism (UML or OWL) driving derivation of multiple simple schema options (GraphQL, JSON schema, XML schema, CSV tables). A solution to the problem of “binding” such schema to controlled vocabularies is less well established, however many organizations publish “profiles” of general standards that declare allowable terms. Work in the W3C, in part driven by OGC need to describe interoperability profiles, is capturing Use Cases and developing recommendations [8][9].

Interoperability of data models can only be achieved by modularity of the models, in other words the more different areas a model covers the more likely an overlap will occur with a related model, so interoperability requires both compatibility of these overlaps and a means to align the relations between them. For the purpose of easy integration specific “Alignment models” describing how one model relates to other modules can be shared and reused as well. The W3C Semantic Sensor Network Ontology exhibits this modularity in Figure 1 .

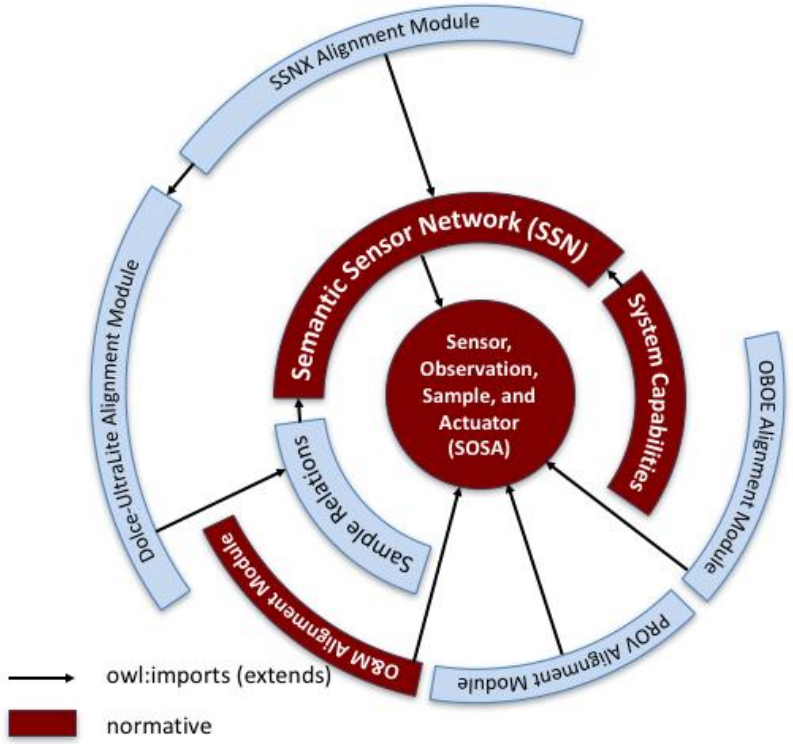


Figure 1: The Semantic Sensor Network Ontology modular architecture showing alignments to related ontologies.

Another highly relevant example of the reuse of modular ontologies comes from the W3C Data on the Web WG in the provision of a Data quality Vocabulary [10]. Data model alignments and other approaches for data integration are described in the next sections.

5.1.2 Methods and models for agricultural data access policies/rights, and federated data ownership

This section provides information regarding different technologies related to the access control which allows for a secure registration of IoT resources, as well as a secure dissemination of that information to the legitimate users. Nowadays, there are a lot of access control models that are applied to different Internet scenarios in which security is vital. In this section a brief description of the most popular models, which are commonly considered and deployed in such scenarios is presented. Regarding policies and rights, please refer to the above-mentioned IDS Information Model [11]. Few of the most relevant models are discussed below.

- The *Mandatory Access Control* (MAC) model [12] the administrator of the system can give permissions for the subject to access objects. The model assigns security labels to subjects and objects, and it is independent of the user operations, only the administrator can modify object security labels. The model puts restrictions on user actions that, while adhering to security policies, avoid dynamic alteration of the underlying policies, but it requires the isolation of the MAC system from the operating system in order to maintain the security policies and prevent unauthorized access. MAC models are not commonly used as an access control system because they are difficult and expensive to implement and maintain. Its usage is usually limited to military applications.
- The *Discretionary Access Control* (DAC) model [13] defines that the access to resources is maintained by users, who can grant permissions to their resources by being included in Access Control Lists (ACL). Each entry in the access control list gives users (or group of subjects) permissions to access resources. The permissions are usually stored by objects to

avoid having a unique and dense matrix which would imply a waste of memory and performance decline. Unlike in MAC as mentioned above, where permissions are given in predefined policies by the administrator, in DAC, permissions are given by users which decide the access rights to the resources they belong to. DAC is broadly adopted by current operation systems based on UNIX, FreeBSD, and Windows.

- The *Role-Based Access Control* (RBAC) model [14] includes features from MAC and DAC providing a more generalized framework that can be customized as per application. In RBAC, users are assigned to roles, which are maintained in a centralized way, and the security policies grant rights to roles rather than to users. Since the users are associated with roles, the user can access certain resources and perform specific tasks. Right granting and policy enforcement are carried out by the administrator and users cannot transfer permissions over their role to other users. RBAC allows creating hierarchies of permissions and inheritance, wherein more restrictive permissions override more general permissions. However, RBAC has some limitations since the administrative issues of large systems where memberships, role inheritance, and the need for fine-grained customized privileges make administration potentially cumbersome.
- The *Attribute-Based Access Control* (ABAC) model [15] in which authorization decisions are based on attributes that the user has to prove (e.g.: age, location, roles, etc.), as well as resources and environmental properties. Attributes labels can be used to describe the entities that must be considered for authorization purposes. Every attribute can consist of a key-value pair such as "Role=Manager". In ABAC, like RBAC, the privileges are usually granted to users through the usage of policies that combine attributes altogether. Thanks to the usage of ABAC, unlike in traditional RBAC models, the number of rules can be reduced, at the expense of more powerful (and complex) rules and more processing and data availability requirements. One of the main advantages of ABAC is requesters do not have to be known a priori by targets, providing a higher level of flexibility for open environments, compared to RBAC models. However, in ABAC everyone must agree on a set of attributes and their meaning while using ABAC which is sometimes hard to accomplish.
- The *AuthoriZation-Based Access Control* (ZBAC) [16] model uses authorization credentials, which are presented along a request to make an access control decision. Unlike ABAC and RBAC systems, in which the user submits an authentication along with the service request, in ZBAC systems, the user submits an authorization along with the request. ZBAC deals with authorization in distributed systems where problems like identity federated management, *Single Sign-On* (SSO) or last privilege appear. In traditional and single domain access control models, the authentication is done at request time in the system's domain; the access decision is made by using that authentication to determine the authorization. With ZBAC, users access resources in a domain other than the home user's domain, and it is based on authentication on the user's domain before the request is made. It should be noticed that this approach requires agreements between the involved domains to trust each other. As a result, users obtain authorizations, which can be represented by cryptographically bound credentials or assertions. Then, the target service or its *Policy Decision Point* (PDP) only needs to verify the validity of the authorization to make an access decision. This is a valuable feature for IoT scenarios in which constrained devices could interact with each other, since interactions with third parties are not required for each communication.

5.1.2.1 Distributed Capability-Based Access Control

According to the requirement in DEMETER the preferred access control model selected was the Distributed Capability-Based Access Control (DCapBAC) [17], because it decouples the grant of access and enforces the access control in two different phases. Firstly, allowing the device/service to perform the first phase once and employ the token as long as it is valid. Secondly presenting a token

together with the requesting message in the next phase. The following section gives a brief overview of the state-of-the-art in DCapBAC.

The technology in DCapBAC is basically an authorization method that integrates the XACML framework[1] with the concept of generating an authorization token which is presented later on to an enforcement point. The XACML framework is used for generating the authorization policies, as well as for issuing an authorization verdict thanks to the PDP (Policy Decision Point)[2] and the XACML policies generated by the PAP (Policy Administration Point)[3]. DCapBAC introduces a new component called Capability Manager (CM) which, after a positive verdict from the PDP (Policy Decision Point), issues an authorization token called Capability Token (CT) based on the authorization request and which is received by the requester entity. This CT contains information regarding the granted authorization, such as the subject, the issuer and its lifetime defined by two timestamps (beginning and end of period) among others. Therefore, the enabler comprises the following entities:

- **CM**: The entity receiving the access control request which forwards them to the XACML PDP to validate it.
- **XACML PDP**: The entity responsible for making access control decisions based on XACML policies defined by PAP.
- **XACML PAP**: The entity responsible for generating the XACML access control policies.

5.1.2.2 Capability Token

The entity Capability Token (CT) is generated by the Capability Manager (CM) after receiving a positive verdict from the PDP. The CT is basically produced as a JSON document compared over traditional formats such as XML, as JSON is getting more popularity in academia and industry in IoT scenarios, since it can provide a simple, lightweight, efficient, and expressive data representation, which is suitable to be used on constrained networks and devices. As shown below, this format follows a similar approach to JSON Web Tokens (JWTs), but including the access rights that are granted to a specific entity.

```
{
   “id”: “7g3vfT_q9vTL2aQ4”,
   “ii”: 1415174237,
   “is”: “issuer@um.es”,
   “su”: “zNwS5FetB4rwzSKsWwSBAxm5wDa=JgLjHU8zSnmeSFQgSG9HhdsJrE8=”,
   “de”: “coap://sensortemp.floor1.computersciencefaculty.um.es”,
   “si”: “SbUudG4zuXswFBxDeHB87N6t9hR=PBQqCN3gpu7nSkuPzDk7kaR3dq1=”,
   “ar”: [
        {
                “ac”: “GET”,
                “re”: “temperature”,
                “f”: 1,
                “co”: [
                        {
                        “t”: 5,
                        “v”: 25,
                        “u”: “Cel”,
                        },
                        {
```

[1] http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
[2] https://ldapwiki.com/wiki/Policy%20Decision%20Point
[3] https://ldapwiki.com/wiki/Policy%20Administration%20Point

```
                        “t”: 6,
                        “v”: 20,
                        “u”: “Cel”,
                        }
                        ]
        }
   ],
   “nb”: 1415174237,
   “na”: 1415175381
}
```

Figure 2: Capability Token Example

The figure above shows an example of the capability token, a brief description of each field is provided.

- *Identifier (id):* This field serves as the identification for a capability token. A random or pseudo-random technique will be employed by the issuer to ensure this identifier is unique.
- *Issued-time (ii):* It identifies the time at which the token was issued as the number of seconds from 1970-01-01T0:0:0Z.
- *Issuer (is):* Issuer or signer of the token.
- *Subject (su):* It refers to the subject to which the rights from the token are granted. A public key has been used to validate the legitimacy of the subject. Specifically, it is based on ECC, therefore, each half of the field represents a public key coordinate of the subject using *Base64*.
- *Device (de):* It is a URI used to identify the device to which the token applies.
- *Signature (si):* It carries the digital signature of the token. As a signature in ECDSA is represented by two values, each half of the field represents one of these values using *Base64*.
- *Access Rights (ar):* This field represents the set of rights that the issuer has granted to the subject. The Access Rights has the following classification:
    - *Action (ac):* Its purpose is to identify a specific granted action. Its value could be any CoAP method (GET, POST, PUT, DELETE), although other actions could be also considered.
    - *Resource (re):* It represents the resource in the device for which the action is granted.
    - *Condition flag (f):* It states how the set of conditions in the next field should be combined. A value of 0 means AND, and a value of 1 means OR.
    - *Conditions (co)*: Set of conditions which have to be fulfilled locally on the device to grant the corresponding action.
        - *Condition Type (t):* The type of condition to be verified.
        - *Condition value (v):* It represents the value of the condition.
        - *Condition Unit (u)*: It indicates the unit of measure that the value represents.

- *Not Before (nb):* The time before which the token must not be accepted. Its value cannot be earlier than the II field and it implies the current time must be after or equal than NB.
- *Not After (na):* It represents the time after which the token must not be accepted.

5.1.2.3 Definition of authorization policies

As mentioned in the previous section, XACML is a framework for authorization and access control that is consistent with the Attribute-Based Access Control Model. XACML uses policies to express the actions that some entity can or cannot perform.

A Policy Administration Point (PAP) generates and stores the XACML policies (in XML format) and feeds them to the Policy Decision Point (PDP).

5.1.2.4 Authorization granting process

When an entity intends to register or access a specific resource of the platform, it must be granted access first. Following the above-mentioned DCapBAC procedure, the entity must first request access to that resource, then it issues an authorization request to the CM specifying the resource. Such request is analyzed by the Capability Manager which issues an XACML authorization request to the PDP, which after validating (i.e. by checking the XACML policies generated by the PAP) responds with a verdict. The CM after receiving a positive answer, generates the corresponding CT which is attached in the final authorization response to the requesting entity.

5.1.3 Data integration in the agri-food sector

As we know that Data integration has been a hot topic of research and interest for several years. Technically data integration means combining heterogeneous data sources in order to provide a unified top-level view over them. Initially it was focused on the implementation of the Extract, Transform, Load (ETL) process [18] used in data warehousing approaches, where data from heterogeneous sources were ETL into a single view schema in order to make them compatible for integration. In the previous document concerning state-of-the-art we have discussed the disadvantages of the ETL approach for data integration and analyzed the new approach known as the semantic data integration. This new approach is more focused on providing a unified query-interface to access real time data over a mediated schema (where real data stays at their sources). We have also stated previously the basic idea behind the semantic data integration in light of the involvement of mappings between the mediated schema and the schemas of the original sources, either from entities in the mediated schema to entities in the original sources (known as Global as View approach) or vice versa (known as Local as View approach), as well as the corresponding transformation of queries. However, we also showed that the main challenge to this approach was the preservation of the semantics of the datasets, e.g. resolving semantic conflicts between heterogeneous data sources.

As Semantic data integration is commonly addressed through the use of ontologies and vocabularies, therefore it is also known as "ontology-based data access". In the previous related work, the relation/difference between semantic integration and semantic interoperability had been discussed with their respective literary definitions [19] [20]. So, in broader concepts, integration concerns data (information) and the identification of logical connections (matching) between classes, properties, and individuals across ontologies or schemas, detecting duplicates, reconciling inconsistent data values, and reasoning with semantic mapping. Whereas semantic interoperability concerns functionality (e.g., provided by services) involving semantic integration as well as, e.g., interoperability of services and tools made possible and driven by semantic integration. Anyways both may deal with the mapping, aligning, translating, and/or merging of ontologies/schemas [19]. Recent approaches have come up that not only address integration of data from relational, graph, RDF and other NoSQL databases, but also hybrid services (e.g., REST APIs). In the following sections

the state-of-the-art of Linked Data is described and its use in the process of publication and integration of heterogeneous data mainly from the agri-food area.

5.1.3.1 Linked Data

The Semantic Web, also known as the Web of Data, is a constantly growing dataspace, including not only a collection of data, but also the provision of relationships between the data. "This collection of interrelated datasets on the Web can also be referred to as Linked Data."[4]. Semantic Web standards, such as RDF [1], OWL [5], and SPARQL [2] had been developed to describe semantic information, including the relationship between data and concepts which provide the base for Linked Data. Presently a JSON-based Serialization for Linked Data (JSON-LD) [21] has also emerged based on the popular JSON format that provides a way to help JSON data interoperate at Web-scale[5].

Linked Data started as an initiative of Tim Berners-Lee[6] (the inventor of the World Wide Web) to release data from proprietary niches, is increasingly becoming one of the most popular methods for publishing data on the Web. There are different reasons involved such as:

1. Linked Data defines simple principles for publishing and interlinking structured data that is accessible by both humans and machines, enabling interoperability and information exchange [22]. For instance, improving the data accessibility lowers the barriers on finding and reusing this data, while providing machine-readable data facilitates the integration of this data into different applications.
2. Linked Data allows to discover more useful data through the connections with other datasets, and to exploit it in a more useful way through inferencing and semantic queries and rules.

As a result, there is a growing number of datasets becoming available in Linked Data format, as depicted in the Linking Open Data (LOD) cloud diagram below. The widespread use and interest in Linked Data has also resulted in the creation of guidelines [23], tutorials ([4] [24]) and best practices [3] on how to generate and publish Linked Data (e.g. [56], [25]).

5.1.3.1.1 Linked Data principles

Technically, Linked Data refers to a set of best practices for publishing, sharing and interconnecting structured data on the Web thereby enabling it to be accessed by both humans and machines. Tim Berners-Lee outlined four principles of linked data:

- Use of Uniform Resource Identifiers (URIs) for identifying entities uniquely in the world;
- Use of HTTP URIs for retrieving resources, so that entities can be referred to and looked to by others;
- Use of standards like RDF and SPARQL for structuring and linking description of resources;
- Use of links to other URIs in the exposed data to improve discovery of related information on the Web.

More in detail, RDF expresses data as triples of the form <subject, predicate, object>. A triple encodes the relation of the object to the subject through the predicate. The subject is a URI (or more generally IRI), which, as specified above, identifies a resource or a concept; the object may be either

[4] https://www.w3.org/standards/semanticweb/data.html
[5] https://json-ld.org/
[6] https://www.w3.org/DesignIssues/LinkedData.html

a literal e.g., number, string, date, or a URI which references another resource. Triples that interlink resources constitute RDF links, which construct the Web of Data.

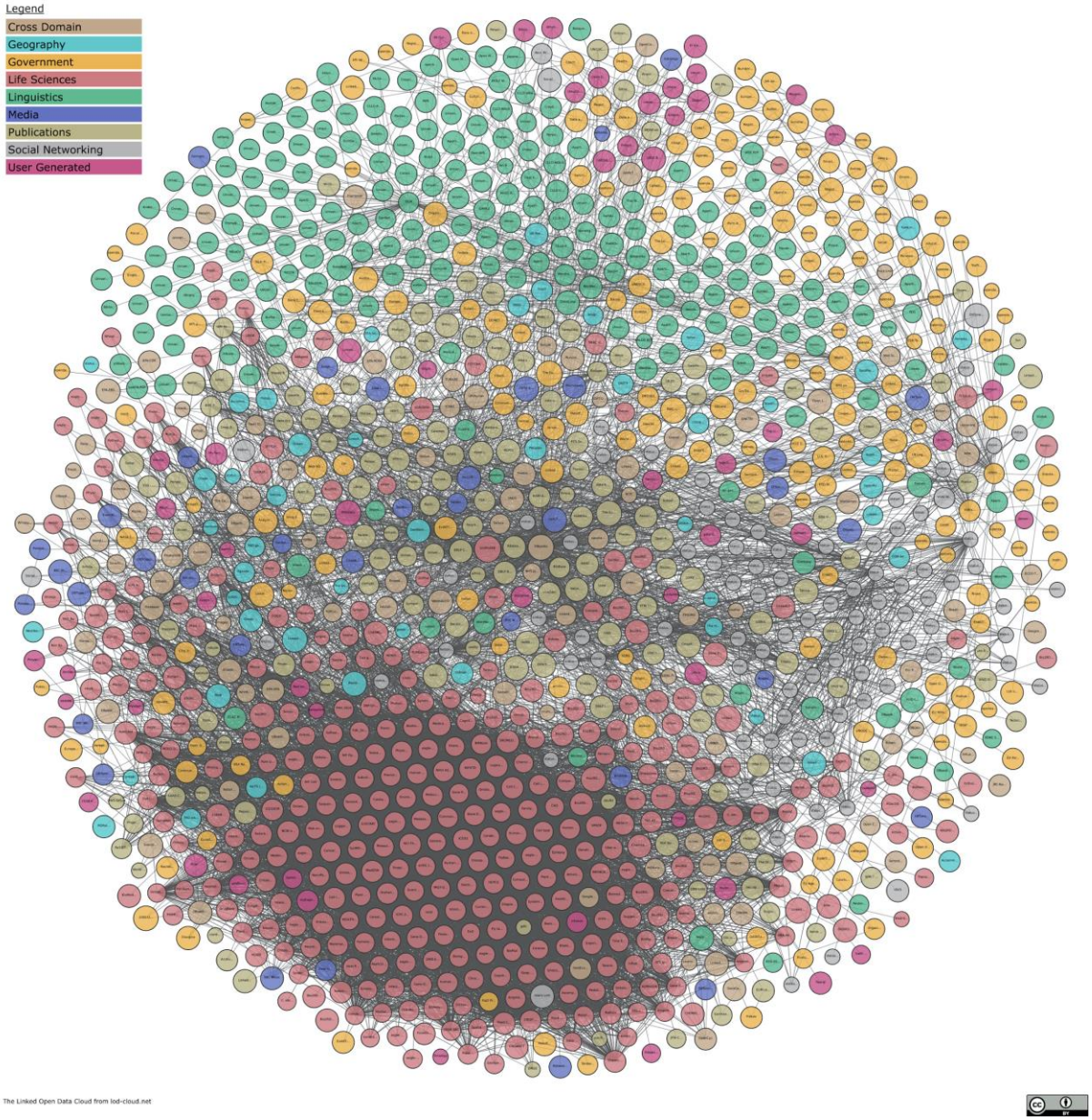


Figure 2: The Linked Open Data Cloud from https://lod-cloud.net/ (version 05/2020)

5.1.3.1.2 Linked Open Data (LOD) [cloud]

Linked Open Data (LOD) as shown in Figure 2 is Linked Data distributed under an open license, i.e., data that aims at a maximum level of reusability not only in a technical but also in a legal way. The goal of LOD is to have a world of stable and clearly scoped sets of knowledge that communicate with each other by using Web addresses (URI) as a linking mechanism. The use of RDF (which is based on URIs) supports creating a world of interconnected knowledge.

The current LOD cloud comprises in the latest version (as of May 2020) more than 1,255 datasets and 16,174 links. Additionally, there are different national knowledge infrastructures emerging that follow the LOD approach, e.g., [26], [27], [28], [29]. In the previous related work, we have discussed

how Linked Data can be used as a technical option to bridge between islands of data and create a genuine knowledge infrastructure. We have discussed that although large cross-domains datasets exist in the LOD cloud (like DBpedia[7] or wikidata[8]) and also some domains are well covered, like Geography, Government, and Bioinformatics, this is still not the case for all domains. For instance, in the agriculture domain we can find relevant thesaurus like AGROVOC[9] from FAO or the National Agricultural Library's Agricultural Thesaurus (NALT[10]), but there is still a lack of datasets related to the agricultural facilities and farm management activities. This is due to the lack of standardized models for the representation of such data, even though some efforts in this direction have been made in the past. In the past FOODIE project[11], for instance we have addressed this issue for the agriculture domain with the development of the FOODIE data model[12] [30]. We had discussed that for the purpose of ensuring the maximum degree of data interoperability, the model is based on INSPIRE generic data models, specially the data model for Agricultural and Aquaculture Facilities (AF), which was extended and specialized. FOODIE model was transformed into semantic format (owl ontology) enabling the publication of agricultural data as Linked Data [31].

Following the LOD approach, repositories have the opportunity to improve the sharing and reuse of data by ensuring that the content is discoverable and readable both by machines and humans, and by using well-known metadata standards and open data vocabularies. The LOD approach is likely to receive an extra boost from the recent emergence of the JSON-LD (Linked Data for JSON) standard [21] which makes RDF data available to web applications without the need for additional parser technologies.

Publishing existing data from repositories as LOD may require different tasks, depending on aspects like the data format, access methods available, data change frequency, etc. However, in general there will be some common tasks, such as: definition of a URI naming strategy to provide resolvable URIs to the entities in the dataset; selection/definition of models to represent the linked data; transformation of data into RDF in any of its serialization formats (e.g., RDFa, Turtle, JSON-LD, etc.); linking to other datasets; provision of machine access to the data (e.g., via a SPARQL endpoint, REST API, etc.). More information about these and other tasks can be found in the guidelines, tutorials and best practices mentioned above. Nevertheless, it is important to highlight that the linking task is ideally carried out with at least one dataset that is already part of the LOD cloud, typically using the most popular and widely used datasets, e.g., DBpedia, geonames or AGROVOC (for the agricultural field). AGROVOC as it covers all areas of interest of the Food and Agriculture Organization (FAO)[13] of the United Nations, including food, agriculture, nutrition, environment, etc., and also because it is part of the LOD cloud. AGROVOC can be used as a backbone to index and organize records of a dataset and to link them to external resources, such as the case of AGRIS (also provided by FAO). AGRIS[14] (the International System for Agricultural Science and Technology) gives access to bibliographic information on agricultural science and technology and it is also part of the LOD cloud.

[7] https://wiki.dbpedia.org/
[8] https://www.wikidata.org/wiki/Wikidata:Main_Page
[9] http://aims.fao.org/vest-registry/vocabularies/agrovoc
[10] http://aims.fao.org/news/nal-thesaurus-now-available-linked-open-data
[11] http://www.foodie-project.eu/
[12] https://github.com/Wirelessinfo/FOODIE-data-model
[13] http://www.fao.org/home/en/
[14] http://aims.fao.org/agris-network

AGROVOC is also used in the FAO geopolitical ontology (also part of the LOD cloud)[15], which provides a master reference for geopolitical information that: manages names in multiple languages (English, French, Spanish, Arabic, Chinese, Russian and Italian); maps standard coding systems (UN, ISO, FAOSTAT, AGROVOC, DBPedia, etc.); provides relations among territories (land borders, group membership, etc.) and tracks historical changes.
By linking a semantic dataset with LOD, we are able to enrich its structured data with additional data or structures from open data. Data enrichment through LOD allows metadata to connect, so that different representations of the same content can be found, and links between related resources can be established.

5.1.3.1.3 Linked Data for Data Federation

In the previous SoTa deliverable we have shown that how in recent agri-related projects (e.g., FOODIE, DATABIO, CYBELE, SIEUSOIL) Linked Data has been used as a federated layer to support a large-scale harmonization and integration of a large variety of data from different sources, providing an integrated view on top of them. During this course of some of these projects activity the PSNC triplestore has been populated with Linked Data has over 1 billion triples, which is one of the largest semantic repositories related to agriculture, as recognized by the EC innovation radar naming it the "Arable Farming Data Integrator for Smart Farming". Additionally, some of these projects also deployed different endpoints providing access to dynamic data sources in their native format as Linked Data by providing a virtual semantic layer on top of them. This has been realized through the implementation of pipelines instantiations for the publication of linked data related to the agri-food domain. The goal of these pipelines is to define and deploy (semi-) automatic processes to carry out the necessary steps to transform and publish different input datasets as Linked Data. Accordingly, they connect different data processing components to carry out the transformation of data into RDF and their linking, and include the mapping specifications to process the input datasets. Each pipeline is configured to support specific input dataset types (same format, model and delivery form). The detailed description of these pipeline instantiations is provided in the later section 9.2 of this deliverable. In the section the diagrams in section 9.2 shows a simplified generic representation of the agriculture data pipelines with the software components and interfaces involved in their life cycle view and aligned with top-level generic activities.

5.1.3.1.4 Linked Data Transformation

The transformation process depends on different aspects like the format in which data source is available, the purpose (target use cases) of the transformation, and the data volatility (how dynamic is the data). Accordingly, the tools and methods used to carry out the transformation are determined firstly by the data format. The general idea in transformation of input data into Linked data is firstly to generate an RDF mapping specification (generally in RML/R2RML format) aligning with some ontologies/vocabulary and then run the suitable component of transformation using the underlying mapping. For various input data sources, the component or methods change although the basic technology flow remains basically the same. An in-depth description of this process and the components involved such as D2RQ[16], Geotriples[17], RML Processor[18] and OpenLink Virtuoso[19]are

[15] http://aims.fao.org/vest-registry/vocabularies/geopolitical-ontology-0
[16] http://d2rq.org/

described in the section 9.2 of this document as they are also part of the data preparation and integration enabler. In this regard previously another component known as Tarql was also presented in the state-of-the-art document. Tarql is a command-line tool for converting CSV files to RDF using SPARQL 1.1 syntax[20].

5.1.3.1.5 Query translation

Query translation enables the provision of a single query interface that allows clients to send a query in a particular language, which is then translated into the appropriate language and format to query and retrieve data from different data sources to return the results. The unified data access/query interface enables access to real time data from their original source over a mediated schema, which translates queries that use terms from the mediated schema into specialized queries to match the schema of the original data source. As in the case of data transformation, the translation relies on mappings between the mediated schema and the source schemas, generally using either a Global As View - GAV [33] (from mediated to sources) (the most typical) or Local As View - LAV [34] (from sources to mediated) approach.

Data sources may support a different query language than the original query language, and so typically the original query has to be re-written in the supported target language to retrieve results. SPARQL 1.1, for example, includes the SERVICE clause that provides a convenient data retrieval formalism: a complex information request over several data sources can be expressed using a single query. However, the existing level of tools support for expressing and processing hybrid information needs using SPARQL is often limited [35], i.e., SPARQL federation implementations generally assume that federation members are data stores containing RDF triples. This situation may be acceptable if the data source is relatively static and making a physical transformation into RDF data is a viable option. However, in many real cases data changes frequently, so better approaches would be to generate RDF views (also known as virtual RDF graphs) over the source data. This is normally implemented via some kind of wrapper, which carries out the translation from SPARQL queries into SQL queries or API calls (among others). Some example tools, particularly useful for Linked Data, Translation are described in the section “9.4.1 Components” in this document, includes

D2R Server, Ephedra, OpenLink Virtuoso (open source edition of Virtuoso Universal Server) and Triplify[21], a small plugin for Web applications, which reveals the semantic structures encoded in relational databases by making database content available as RDF, JSON or Linked Data.

5.1.3.1.6 Link Discovery

As mentioned before, one of the main principles of Linked Data refers to the definition of links to other URIs to improve discovery of related information on the Web. The central idea of Linked Data is to extend the Web with a data commons by creating typed links between data from different data sources[22]. The data links that connect data sources take the form of RDF triples, where the subject of the triple is a URI reference in the namespace of one dataset, while the object is a URI reference in

[17] http://geotriples.di.uoa.gr/
[18] https://github.com/RMLio/RML-Processor
[19] http://vos.openlinksw.com/owiki/wiki/VOS
[20] https://www.w3.org/TR/sparql11-query/
[21] https://sourceforge.net/projects/triplify/
[22] https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

the other [32]. When the datasets use the same schema to identify entities the links between them can be easily made explicit, or if the datasets are relatively small links can be created manually by the dataset providers. However, as Linked Data sources have been growing, both in size and in number, the need for automated or semi-automated methods to discover and generate RDF links has also increased. There are a number of tools available for this task e.g. SILK, LIMES and Geo-L, which are described in the section "9.4.1 Components" in detail as they are also part of the data preparation and data integration enabler described in the later sections of this document.

5.1.3.1.7 Linked Data Exploitation

The resulting datasets can thereafter be exploited through SPARQL queries, or via a wide range of user interfaces (components in section 9.4.1). Some examples of these interfaces include:

- SPARQL endpoint interface In Virtuoso triplestore to execute queries: https://www.foodie-cloud.org/sparql
- Faceted search interface in Virtuoso triplestore to navigate the linked datasets http://www.foodie-cloud.org/fct/
- Map visualization via HSLayer applications, e.g., http://app.hslayers.org/project-databio/land/
- Metaphactory Linked Data exploitation platform: http://metaphactory.foodie-cloud.org/resource/Start

5.1.3.2 Data Enrichment

When using semantic technologies to their full potential, the data is expected to be modelled using ontologies that are interlinked among them. The process of semantically enriching data enables not only content reuse but also the inference of new knowledge. If the data model is developed from scratch this is not an issue. However, in many cases this is not possible because the data already exists and is stored in databases or it is provided by legacy systems that we cannot modify. For this reason, there are tools that either a) link existing structured or unstructured information to specific ontologies or b) transform/map the data to a semantic representation. In this section the focus is on the first type of tools. The second type of tools are already covered in Sections "Linked Data Transformation" and "Query Translation" respectively.

Semantic enrichment, also known as semantic annotation (or tagging), enhances the source data with a context that is linked to some structured knowledge of a domain or application (ontology), which can be then exploited by different applications and services. This is done by attaching additional information to various concepts (e.g., people, things, places, organizations, etc.) in a given text or any other content[23]. Since the newly discovered knowledge is described by standard ontologies, stored in machine-readable format and accessible through standard APIs and protocols, it can also be used for further machine processing allowing better integration with existing knowledge bases and their publication in the Linked Open Data (LOD) Cloud, discovering and understanding relations and dependencies between resources, as well as the implementation of all other kinds of user scenarios. For instance, it can support the matching discovery between data

[23] https://www.ontotext.com/knowledgehub/fundamentals/semantic-annotation/

elements, overcoming the differences among different structures and providing a solution for the (semi) automatic information integration and systems interoperability [36]. Following are few tools and methods that are used for linking existing structured or unstructured information to specific ontologies:

- **JSON-LD**[24] provides a way of linking structured information in JSON format to specific concepts in an ontology. It is a lightweight Linked Data format, easy for humans to read and write. By adding semantic annotations to JSON documents in a way that preserves their original structure, it provides a way to help JSON data interoperate at Web-scale. However, JSON-LD is just a method for encoding the linked data, not a tool to generate or discover the links between JSON elements and an ontology.
- **DBpedia Spotlight**[25] is a tool for automatically annotating mentions of DBpedia resources (i.e., almost all concepts from the domain of general knowledge, as well as some concepts from specific domains) in text by performing named entity extraction, including entity detection and name resolution. It provides a solution for linking unstructured information sources to the Linked Open Data Cloud through DBpedia, as DBpedia is a hub of the LOD Cloud having links from and to many other datasets.
- **GATE** (general architecture for text engineering) is another tool for semantic enrichment. GATE[26] is an open software consisting of a family of tools for text processing. It has a set of reusable processing resources for common NLP tasks, including the Information Extraction system (ANNIE)[27] enabling semantic enrichment of textual content.
- **COGITO**[28] is a commercial solution enabling semantic enrichment of content leveraging pre-trained vertical models and out-of-the-box and customizable taxonomies. Cogito is built on a knowledge graph (Sensigrafo)[29], where concepts (syncons) are represented as groups of lemmas with the same meaning. Syncons are interconnected through semantic and linguistic relations like hypernymy, hyponymy and other properties. Among other purposes, Cogito leverages the knowledge contained in Sensigrafo to disambiguate the meaning of a word by recognizing its context.

As a general rule for the above-mentioned approaches, the first step is the identification or definition of the ontologies that are to be used. JSON-LD is a domain independent way of embedding RDF-style semantics into JSON (depending on the context, different ontologies will be used). DBpedia Spotlight relies on the DBpedia knowledge graph, COGITO relies on their closed Sensigrafo knowledge graph, while in GATE users can specify the underlying ontology.

Besides there are also few domain-specific annotation tools. More specifically for the agri-food sector we can mention the following:

[24] https://json-ld.org/
[25] https://www.dbpedia-spotlight.org/
[26] https://gate.ac.uk/
[27] http://services.gate.ac.uk/annie/
[28] https://expertsystem.com/products/text-analytics-software-cogito-discover/
[29] http://expertsystemtraining.com/

- **AgroTagger**[30] applies text-mining on top of agri-food research outcomes. It is a keyword extractor that uses a subset of the AGROVOC thesaurus[31] (about 2,5K concepts out of the total >40K concepts of AGROVOC) as a set of allowable keywords, used for indexing information resources. A complete description of AGROVOC is also available in the SoTA for T2.1.
- **FOODIE semantic annotation service**[32] provides a simple REST API[33] designed for creating, updating and retrieving semantic annotations. The service orchestrates other components (existing annotation tools described below) in order to control and fully perform data analysis process, creates semantic form of the generated data and persists semantic data using semantic store. The current implementation of the semantic annotation service uses AgroTagger and Babelfly (described below). Annotations created by the semantic annotation service are modelled using the Modular Unified Tagging Ontology (MUTO) [37] which is designed specifically for tagging and folksonomies. MUTO allows representing public and private tagging, simple and auto generated tags and others. It is also easily extensible since all concepts defined in MUTO ontology inherits from other more general ontologies like SKOS[34] , SIOC[35] or vocabularies as RDFS[36]. FOODIE semantic annotation service is described in details in the section 9.4.1.5.
- **Babelfy**[37] is another tools example which provides a unified, multilingual, graph-based approach to Entity Linking and Word Sense Disambiguation based on a loose identification of candidate meanings coupled with a densest subgraph heuristic which selects high-coherence semantic interpretations. Babelfy is based on the BabelNet multilingual semantic network [38] and jointly performs disambiguation and entity linking. BabelNet[38] is both a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network which connects concepts and named entities in a very large network of semantic relations, made up of about 16 million entries, called Babel synsets. Each Babel synset represents a given meaning and contains all the synonyms which express that meaning in a range of different languages. Babelnet covers 284 languages and is obtained from the automatic integration of several sources including Wordnet, Wikipedia, Geonames, etc.

5.1.3.3 Data model alignment

In the context of DEMETER, model alignment refers to the process of determining correspondences between concepts in data schemas, vocabularies and/or ontologies. This can apply, for example, between different ontologies (called ontology alignment/matching), different database schemas (called schema matching), or between heterogeneous data structures (e.g., a database schema and an ontology, a JSON schema/structure and an ontology or a CSV document structure and an

[30] https://github.com/fcproj/agrotagger
[31] http://aims.fao.org/vest-registry/vocabularies/agrovoc
[32] https://www.foodie-cloud.org/semanticAnnotation/
[33] http://shorturl.at/gmGI8
[34] https://www.w3.org/TR/skos-reference/
[35] https://www.w3.org/Submission/sioc-spec/
[36] https://www.w3.org/TR/rdf-schema/
[37] http://babelfy.org/
[38] https://babelnet.org/

ontology). Note that, in this context, the correspondences are restricted to data structure elements (e.g., classes, properties, columns, etc.), not covering data elements (e.g., class instances, cell values), which is covered in Section "Data Linking". Also note that this section is concerned with the process of identifying when two objects are semantically related, and not with the mapping that refers to the transformations between the objects (discussed in Section "Data Transformation"). Correspondences are expressed using specific mapping languages like W3C R2RML[39] (for mapping from relational databases to RDF graphs), RML[40] (a generalization that also covers JSON and XML sources), Semi-Automatic Schema Matching and INTegration (SASMINT) Derivation Markup Language - SDML [1] (for expressing schema matches), Expressive and Declarative Ontology Alignment Language - EDOAL[41] (for expressing ontology alignments), as well as using axioms or constructs from general ontology languages and vocabularies (e.g., OWL[42], RDFS, SKOS[43]) or rules languages (e.g., SWRL[44]) that can interpret the correspondences.

5.1.3.3.1 (Database) schema matching

Automatic resolution of schema heterogeneity still remains a major bottleneck for provision of integrated data access/sharing among autonomous, heterogeneous, and distributed databases. In order to provide transparent access to external data and enable the sharing of information among databases, their schema heterogeneity needs to be identified and resolved and then the correspondences among schemas need to be identified. This process is called schema matching [39]. Approaches to schema integration can be broadly classified as the ones that exploit either just schema information or schema and instance level information [40]. The approaches are:

- **Schema-level matchers** which consider only schema information, not instance data. Such information may include: name, description, data type, relationship type, constraints and schema structure
- **Instance-level matchers** which use instance data to extract insights regarding the contents and meaning of schema elements, which is normally in combination to schema-level matchers to improve the accuracy of the results.
- **Hybrid matchers**, which combine multiple matching approaches to determine matches based on multiple criteria and/or information sources. They normally use additional information like dictionaries or thesauri.
- **Matching information reuse,** which exploits previous matching information as auxiliary information for future matching processes.
- **Sample prototypes**, which are typically implemented as rule-based or learner-based systems.

The quality of schema matching is commonly measured in terms of precision and recall. An example tool for (semi) automatic schema matching is **SASMIT** (Semi-Automatic Schema Matching and INTegration). SASMINT [41] follows a composite approach in schema matching by combining the results of different algorithms, and includes a Sampler component for helping the user to assign the

[39] https://www.w3.org/TR/r2rml/
[40] http://rml.io/spec.html
[41] http://alignapi.gforge.inria.fr/edoal.html
[42] https://www.w3.org/TR/owl2-overview/
[43] https://www.w3.org/TR/swbp-skos-core-spec/
[44] https://www.w3.org/Submission/SWRL/

weights to algorithms. It implements an XML-based derivation language (called SDML) to save the results of schema matching and schema integration, and defines the components of integrated schemas, to allow the automated query processing of integrated sources. Other tools include: **SEMINT** (SEMantic INTegrator) [42] system, which uses both schema and instance level information, but provides no graphical user interface (GUI); **Cupid** [43] system, which exploits a combination of name (using string similarity) and structure matcher, and provides no GUI; **S-Match** [54], which exploit different element and structure level match techniques, representing results in term of equivalences, more/less general, mismatch and overlapping, but provides no GUI; and **COMA++** [44], [45] a library of different types of matches and also a sophisticated GUI.

5.1.3.3.2 Ontology matching

Ontology matching (also called alignment) is the process of determining correspondences between concepts in ontologies, where a set of correspondences is also called an alignment. The correspondences can be used for various tasks, such as ontology evolution, ontology merging, query answering, data translation, or data integration [46]. Hence, matching ontologies enables the knowledge and data expressed with respect to the matched ontologies to interoperate [47]. A large number of solutions for ontology matching have been proposed in the last decades [46], [48], [49]. Generally, ontology matching systems implement multiple algorithms, exploiting therefore different ontology matching techniques. According to [48], these matching techniques can be classified using a top-down approach focused on the interpretation that the different techniques offer to the input information, but also bottom-up, focusing on the type of the input that the matching techniques use. In the end, however, both approaches meet at the concrete technique layer, which include techniques like: formal resource-based (using upper-level/domain ontologies, linked data), , informal resource-based (using directories, annotated resource), string-based (using name/description similarity, namespaces), language-based (using tokenization, lemmatization, etc.), constraint-based (using type similarity, key properties), taxonomy-based (using taxonomic structure), graph-based (using graph homo-morphism, path, children, leaves), instance-based (using data analysis and statistics) and model-based (using DL reasoners). Similarly, there are various systems/frameworks available for ontology alignment. The Ontology Alignment Evaluation Initiative (OAEI)[45] provides an overview of the latest systems for ontology alignment/matching and their performance. This initiative is considered as the most prominent one regarding the evaluation of different matching systems and it helps practitioners improve their works on matching techniques. Some of the most known and used tools include:

- **Alignment API and server**[46] [50]. This system is an API and implementation for expressing and sharing ontology alignments. It supports, in particular, alignment storing, correspondence annotation and sharing. It is accessible from other tools and applications through a versatile interface (HTTP, REST, SOAP, FIPA ACL). It defines and implements an alignment format and the Expressive and Declarative Ontology Alignment Language (EDOAL) to express alignments that the API can input or output.
- **AgreementMaker** [51] is a schema and ontology matching system. It allows customization of the matching process, including several matching methods to be run on inputs with different

[45] http://oaei.ontologymatching.org/
[46] http://alignapi.gforge.inria.fr/

levels of granularity, also allowing to define the amount of user participation and the formats that the input ontologies as well as the results of the alignment may be stored in.

- **RiMOM** [52] system uses three different matching strategies (name-based, metadata-based and instance-based), whose results are then filtered and combined. A similarity propagation procedure is iteratively run until no more candidate mappings are discovered and the system converges.
- **Aroma** [43] system finds equivalence and subsumption relations between classes and properties of two different taxonomies. It is defined as a hybrid, extensional and asymmetric approach that lays its foundations on the association rule paradigm and statistical measures.
- **S-Match**[47] [53] is an open source semantic matching framework that transforms input tree-like structures such as catalogs, conceptual models, etc., into lightweight ontologies to then determine the semantic correspondences between them. It includes implementation of several semantic matching algorithms, each one suitable for different purposes.

5.1.3.3.3 Schema and ontology matching

Schema and ontology matching aim at identifying semantic correspondences between data structures or models such as database schemas, XML message formats, and ontologies. Generally, many of the techniques and methods mentioned for ontology matching also apply here; however, the tool implementing those techniques must support different types of schema inputs. Some example tools with such support include:

- **S-Match** [53] semantic matching framework can take any two tree-like structures (such as database schemas, classifications, lightweight ontologies) and returns a set of correspondences between those tree nodes which semantically correspond to one another.
- **COMA** [54] (also called **COMA++** [55])[48]: COMA++, which extends COMA, is a customizable generic matching tool for both schemas and ontologies. It takes over the flexible composite match approach of COMA to combine different match algorithms, but extends its predecessor with major improvements, including a comprehensive graphical user interface, generic data model to uniformly support schemas and ontologies written in different languages, new matchers, especially for ontology matching and reusing existing match results, etc. Formats supported include XSD, XML Data Reduced (XDR), OWL, and relational schemas.
- **GeRoMeSuite** [56][49] is a framework providing an environment to simplify the implementation of model management operators. GeRoMeSuite implements several fundamental operators such as Match, Merge, and Compose. The system contains a workspace in which all models and mappings are managed, thereby making them accessible to all operator implementations in a uniform way. It is possible to import SQL, XML Schema and OWL models. The mappings supported are simple morphisms, structured intentional mappings that are used for schema integration, and executable mappings in the form of second order tuple generating dependencies that can be composed and executed.

[47] https://sourceforge.net/projects/s-match/
[48] https://sourceforge.net/projects/coma-ce/
[49] http://dbis.rwth-aachen.de/cms/projects/GeRoMeSuite

5.1.3.4 Data visualization

Data visualization is the presentation of data in a pictorial or graphical format, and a data visualization tool is the software that generates this presentation. Data visualization provides users with intuitive means to interactively explore and analyze data, enabling them to effectively identify interesting patterns, infer correlations and casualties, and supports sense-making activities. Data visualization and user interaction play a key role in exploiting Big Data sources. They support users for browsing, understanding and discovering data insights. Nonetheless, Big data characteristics, such as volume, variety and velocity pose a number of challenges for visualization. Indeed, current visualization and exploration systems should effectively and efficiently handle the following aspects:

- **Real-time Interaction:** Efficient and scalable techniques should support the interaction with billion objects datasets, while maintaining the system response in the range of a few milliseconds.
- **On-the-fly Processing**: Support of on-the-fly visualizations over large and dynamic sets of volatile raw (i.e., not preprocessed) data is required.
- **Visual Scalability:** Provision of effective data abstraction mechanisms is necessary for addressing problems related to visual information overloading (a.k.a. overplotting).
- **User Assistance and Personalization**: Encouraging user comprehension and offering customization capabilities to different user-defined exploration scenarios and preferences according to the analysis needs are important features.

Nowadays, an increasingly large number of diverse users (i.e., users with different preferences or skills) explore and analyze data in different sectors. For example, farmers need to measure and understand the impact of the huge amount and variety of data which drive agricultural production and livestock management. Big Data is expected to have a large impact on Smart Farming [57] and involves the whole supply chain. Smart sensors and devices produce large amounts of data that provide unprecedented decision-making capabilities. Big Data will influence the entire food supply chain. Big data is being used to provide predictive insights in farming operations, drive real-time operational decisions, and redesign business processes for game-changing business models. Example of Big Data visualization and analysis for agriculture include from few of the previous use cases can be listed as below:

- **Yield predictions** - The concept of yield productivity zones was introduced in the FOODIE project and further developed in the DataBio project. It aimed at the discovery, verification, and user-friendly visualization of long-term high and low yield productivity zones.
- **Analysis and visualization of Agriculture Linked Open Data** - Linked data are increasingly becoming one of the most popular methods for publishing data on the Web. There is still a lack of datasets related to the agricultural facilities and farm management activities. This is in part due to the lack of standardized models for the representation of such data. The FOODIE project addressed this issue with the development of the FOODIE data model and related ontology. A key motivation was to represent a continuous area of agricultural land with one type of crop species, cultivated by one user with one farming mode (e.g., conventional vs. transitional vs. organic farming). [58]

- **Farm machinery visualization** - Monitoring of machinery fleet movement and especially its spatiotemporal changes brings new insights about the consequences of human decisions from many areas. Economic reasons are related to economic evidence for a farmer, including fuel consumption, efficiency of trajectory etc. to revenue authority or subsidies management. On the other hand, ecologic motivations aim to decrease the environmental burden caused e.g. high CO2 emissions due to a lack of movement optimization, water pollution by nitrogen due to excessive fertilization.
- **3D interactive visualization of yield productivity zones** - Allows users to explore yield productivity zones with respect to topography (represented as DTM – Digital Terrain Model, DSM – Digital Surface Model, slope, slope orientation, and topographic wetness index). The perspective view contains a three-dimensional model of the farm plots and shows the area of interest in a perspective projection taking into account the observer position and his or her line of sight.
- **Precision Livestock Farming (PLF)** – visualization of data coming from various sensors including the number of animals, mortality, temperature and humidity, feed and water consumption, activity and distribution of the animals and number of coughs. The output can be customized for each farm, based on its specificities. It includes information on production parameters, climatic conditions as well as behavior, health and welfare of the animals. [59]

Some visualization tools that have been used in the agri-food domain include:

- **HSLayers NG**[50] is a library that extends OpenLayers 5 functionality by providing a foundation to build map GUI and extra components such as layer manager, permalink generating, styling of vector features, including OpenGIS® Web Map Service Interface Standard (WMS) layers to the map in a user-friendly way. More about this visualization tool is provided in the section “9.4.1 Components”.
- **KNOWAGE**[51] is an open source suite developed by ENGINEERING that combines traditional data and big data sources into valuable and meaningful information. Knowage provides advanced self-service capabilities that give autonomy to the end-user, who is able to build his own analysis, get insights on data and turn them into actionable knowledge for effective decision-making processes. The software is flexible because it adopts open standards and can be used in various environments without considerable requirements. Knowage follows a modular approach, features a scalable architecture, and the use of open standards to ensure easy customization and the development of user-friendly solutions
- **Grafana**[52] is an open source data visualization platform that can work with different data sources. Grafana offers multiple chart and visualization types that can be added into a dashboard for visual analysis. Main advantages include: i) dynamic and sophisticated visualization dashboards; ii) work with different data sources; iii) big volume of documentation and examples

[50] https://ng.hslayers.org/
[51] https://www.knowage-suite.com/site/home/
[52] https://grafana.com/

- **Kibana**[53] is an open source data visualization tool designed to work with information stored in ElasticSearch indexes. Kibana has a lot of different chart types, tables and maps for data visualization and real time analysis of streaming data. Main advantages include: i) multiplatform; ii) visualizations and searches are stored as objects that can be used in more than one dashboard; iii) sophisticated visualization dashboards; iv) big volume of documentation and examples
- **Apache Zeppelin**[54] is an open source web-based notebook designed for interactive data analysis and visualization. Zeppelin works with different platforms and interpreters, and can be used for example as front-end for Apache Spark. Main advantages include: i) multi-language backend; ii) angular JS and Bootstrap based UI; iii) big volume of documentation and examples
- **Metaphactory**[55] is an end-to-end Knowledge Graph platform for Knowledge Graph management, rapid application development and end-user oriented interaction. Metaphactory makes authoring, building, curating, editing, integrating, linking, searching, and visualizing Knowledge Graphs easy, fast and affordable. Metaphactory is a commercial product, but also features academic licenses. Main advantages include: i) standard connectors for a variety of data formats to integrate and interlink heterogeneous data sources; ii) graphical SPARQL Endpoint UI for easy querying and management of saved queries in a query catalog; iii) declarative Web components for end-user interaction; iv) end-user friendly navigation, exploration and visualization of Knowledge Graphs; v) rich semantic search with visual query construction and faceting. More about this tool is provided in the section “9.4.1 Components”.

5.2 Analytics and knowledge extraction

This section presents the State of the Art on tools for (big data) analytics and knowledge extraction as well as database management of such data; while also considering data quality issues and also being able to explain the result of a particular tool.

5.2.1 Big data tools in support of data processing and analytics

Machine learning (ML) and knowledge extraction (KE) techniques often require data processing tools and big data frameworks and libraries. The scope of this section is to describe the available Big Data Platforms, the Big Data File / Storage Systems, as well as the existing Big Data Batch / Stream Processing approaches and the connectors in order to present a system in its entirety which we can store, send and analyze data.

5.2.1.1 Big Data Platforms and Infrastructure

Frameworks such as Hadoop, based on MapReduce[56] [60], are the baseline for many systems dealing with big data. In [61] a survey of tools and frameworks based on Hadoop for ML can be found. Apache Mahout and Apache Samsara are examples of ML libraries used on top of the Hadoop ecosystem. Some well-known frameworks are briefly introduced in the following:

[53] https://www.elastic.co/products/kibana
[54] https://d3js.org/
[55] https://www.metaphacts.com/product
[56] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

Cloudera Enterprise - Cloudera Enterprise (Cloudera) includes CDH, an open source Hadoop-based platform. CDH is Cloudera's 100% open source platform distribution, including Apache Hadoop and built to meet enterprise demands. By integrating Hadoop with more than a dozen other critical open source projects, Cloudera has created a functionally advanced system that helps in performing end to end Big Data workflows. Cloudera Enterprise is available in three editions, each offering varying levels of service management capabilities. The basic edition provides management capabilities to support cluster running core CDH services that include HDFS, Hive, MapReduce, Oozie, YARN and ZooKeeper. Cloudera's Benefits are as follows:

- One integrated system, bringing diverse users and application workloads to one pool of data on common infrastructure; no data movement is required
- Perimeter security, authentication, granular authorization, and data protection
- Enterprise-grade data auditing(control), data lineage, and data discovery
- Native high-availability, fault-tolerance and self-healing storage, automated backup and disaster recovery, and advanced system and data management -
- Apache-licensed open source to ensure the user's data and applications from misappropriation, and an open platform to connect with all of the user's existing investments in technology and skills
- One massively scalable platform to store any amount or type of data, in its original form, for as long as desired or required
- Integrated with the user's existing infrastructure and tools
- Flexible to run a variety of enterprise workloads - including batch processing, interactive SQL, enterprise search and advanced analytics

Cloudera's Disadvantages

- Not fully Open Source, couple of components of the distributions are privately owned, meaning with public contributions are not welcome
- More Up to date technologies
- Improvements to Cluster Management tool is required, which are already available to its competitors

Hortonworks - Architected, developed, and built completely in the open, Hortonworks Data Platform (Hortonworks) provides Hadoop designed to meet the needs of enterprise data processing. HDP is a platform for multi-workload data processing across an array of processing methods - from batch through interactive to real-time - all supported with solutions for governance, integration, security and operations. Furthermore, Hortonworks is a massive contributor to the open-source Hadoop community focused on evolving it into a broadly capable data-management platform. Everything in the Hortonworks Data Platform (HDP) is freely available as open-source software. Hortonworks distribution has master-slave architecture and it can support among others MapReduce, YARN, Spark, Kafka, Flink and HBase. Furthermore, Hortonworks has no proprietary software, uses Ambari for management and Stinger for handling queries, and Apache Solr for searches of data.

Hortonworks's Benefits

- Completely Open: HDP is the only completely open Hadoop data platform available. All solutions in HDP are developed as projects through the Apache Software Foundation (ASF). There are no proprietary extensions or add-ons required.
- Fundamentally Versatile: At its heart HDP offers linear scale storage and compute across a wide range of access methods from batch to interactive, to real time, search and streaming. It includes a comprehensive set of capabilities across governance, integration, security and operations.
- Wholly Integrated: HDP integrates with and augments your existing applications and systems so that you can take advantage of Hadoop with only minimal change to existing data architectures and skillsets. Deploy HDP in-cloud, on-premise or from an appliance across both Linux and Windows.
- Hortonworks provides you with the flexibility to run the same industry-leading, open source platform to gain data insights in the data center as well as on the public cloud of choice.

Hortonworks's Disadvantages:

- Installation can be complex and needs to be streamlined
- There exist minor stability issues with the platform. As a remedy for that someone may choose not to follow the latest releases to make sure more stable versions are used.
- Upgrading from lower versions is at the moment feasible but demands effort.
- There is room for improvement in monitoring. The Ambari Management interface on HDP is just a basic one and does not have many rich features.

MapR - The MapR Distribution (MapR) including Apache Hadoop[57] provides an enterprise-grade distributed data platform that can reliably store and process big and fast data. MapR Distribution gives a good foundation for running batch, interactive, and real-time applications. With an open choice approach to open source, MapR gives you a broad range of technologies (multiple projects for SQL-on-Hadoop, NoSQL databases, execution engines such as Spark, etc.) to choose from, so that the right tool is employed for a specific need. Furthermore, MapR M7 Hadoop distribution addresses weakness in HBase by doing away with region servers, table splits and merges, and data-compaction steps. MapR has also implemented its own architecture for snapshotting, high availability, and system recovery. With M7, MapR also introduced optional LucidWorks Search software on top of Hadoop for building out recommendation engines, fraud-detection, and predictive applications.

The three platform services offered (MapR-FS, MapR-DB, and MapR Streams), are unified by common core capabilities built into the underlying platform such as high availability, real-time access, unified security, multi-tenancy, disaster recovery, a global namespace, self-healing, and management and monitoring. The MapR Converged Data Platform allows you to quickly and easily build breakthrough, reliable, real-time applications by providing:

[57] http://hadoop.apache.org

- Single cluster for streams, file storage, database, and analytics.
- Persistence of streaming data, providing direct data access to batch and interactive frameworks, eliminating data movement.
- Unified security framework for data-in-motion and data-at-rest, with authentication, authorization, and encryption.
- Utility-grade reliability with self-healing and no single point-of-failure architecture.

MapR's Benefits are as follows:

- Unified Big Data Platform: Capable of creating a complete picture of all data, including high-velocity, real-time data, to find previously unidentifiable insights. Process more data types with the schema-less flexibility and the high-velocity read/write capabilities of the integrated in-Hadoop online database platform.
- Proven Production Readiness: Ability to get continuous value from your data with the technology proven in production to meet strict service level agreements. Deploy 24x7 online applications with enterprise-grade capabilities to achieve zero downtime. Run HBase-compatible applications with zero database administration.
- Consistent High Performance at Any Scale: Ability to get faster results on larger data sets to respond more quickly to more complete data. Achieve quicker application responsiveness for an enhanced user experience. Easily load and process high volumes and high velocities of incoming data.

Apache Ambari - Ambari (Apache Ambari)[58] is a completely open source management platform for provisioning, managing, monitoring and securing Apache Hadoop clusters. Apache Ambari can help in taking the guesswork out of operating Hadoop. Apache Ambari, as part of the Hortonworks Data Platform, allows enterprises to plan, install and securely configure HDP making it easier to provide ongoing cluster maintenance and management, irrespective to the size of the cluster. Ambari makes Hadoop management simpler by providing a consistent, secure platform for operational control with an intuitive Web UI as well as a robust REST API, which is particularly useful for automating cluster operations. The tools allow Hadoop operators get the following core benefits:

- Simplified Installation, Configuration and Management
- Centralized Security Setup.
- Full Visibility into Cluster Health
- Highly Extensible and Customizable

[58] https://hortonworks.com/apache/ambari/

5.2.1.2 Big Data File / Storage Systems

In general, Data stores are grouped according to their data model, i.e. SQL vs. NoSQL (Cattell et al. 2011):

- Key-value Stores: These systems store values and an index to find them, based on a programmer defined key.
- Document Stores: These systems store documents, as just defined. The documents are indexed, and a simple query mechanism is provided.
- Extensible Record Stores: These systems store extensible records that can be partitioned vertically and horizontally across nodes. Some papers call these “wide column stores”.
- Relational Databases: These systems store (and index and query) tuples

Key Value Stores: The simplest data stores use a data model similar to the popular memcached distributed in-memory cache, with a single key-value index for all the data. These systems are called key-value stores. Unlike memcached, these systems generally provide a persistence mechanism and additional functionality as well: replication, versioning, locking, transactions, sorting, and/or other features. The client interface provides inserts, deletes, and index lookups. Like memcached, none of these systems offer secondary indexes or keys. Some noticeable Key Value Stores include:

- Project Voldermort
- Riak
- Redis
- Tokyo Cabinet
- Document Stores

Document stores support more complex data than the key-value stores. Although termed “document store” and these systems could store “documents” in the traditional sense (articles, Microsoft Word files, etc.), a document in these systems can be any kind of “pointerless object”. Unlike the key-value stores, these systems generally support secondary indexes and multiple types of documents (objects) per database, and nested documents or lists. Like other NoSQL systems, the document stores do not provide ACID transactional properties. Here are some of the Document Stores:

- SimpleDB
- CouchDB
- MongoDB
- Terrastore

Extensible Record Stores: The extensible record stores seem to have been motivated by Google’s success with BigTable. Their basic data model is rows and columns, and their basic scalability model is splitting both rows and columns over multiple nodes:

- Rows are split across nodes through shading on the primary key. They typically split by range rather than a hash function. This means that queries on ranges of values do not have to go to every node.

- Columns of a table are distributed over multiple nodes by using “column groups”. These may seem like a new complexity, but column groups are simply a way for the customer to indicate which columns are best stored together.

Extensible Record Stores include:

- Hadoop Distributed File System (HDFS)
- HBase
- Cassandra

Hadoop Distributed File System - The Hadoop Distributed File System (HDFS)[59] is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. Hardware failure is the norm rather than the exception. An HDFS instance may consist of hundreds or thousands of server machines, each storing part of the file system’s data. The fact that there are a huge number of components and that each component has a non-trivial probability of failure means that some component of HDFS is always non-functional. Therefore, detection of faults and quick, automatic recovery from them is a core architectural goal of HDFS. HDFS has been designed to be easily portable from one platform to another. This facilitates widespread adoption of HDFS as a platform of choice for a large set of applications. An HDFS has a master/slave architecture and an HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are several DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system’s clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode. HDFS is part of the Apache Hadoop Core project. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

HBase - Apache HBase (Jiang 2012) is the Hadoop database, is a type of NoSQL database, a distributed, scalable, big data store. When there is a need for random, real-time read/write access to your Big Data, Apache HBase is befitting. This project's goal is the hosting of very large tables (billions of rows X millions of columns) using clusters of commodity hardware. Apache HBase is an open-source, distributed, versioned, non-relational database model. Some key features include the following:

[59] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

- Linear and modular scalability.
- Strictly consistent reads and writes.
- Automatic and configurable sharing of tables
- Automatic failover support between RegionServers.
- Convenient base classes for backing Hadoop MapReduce jobs with Apache HBase tables.
- Easy to use Java API for client access.
- Block cache and Bloom Filters for real-time queries.
- Query predicate push down via server-side Filters
- Thrift gateway and a RESTful Web service that supports XML, Protobuf, and binary data encoding options
- Extensible jruby-based (JIRB) shell
- Support for exporting metrics via the Hadoop metrics subsystem to files or Ganglia; or via JMX

Cassandra - Apache Cassandra[60] (Rabl et al. 2012) is a free and open-source distributed NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers robust support for clusters spanning multiple datacenters, with asynchronous master less replication allowing low latency operations for all clients. Data is automatically replicated to multiple nodes for fault-tolerance. Replication across multiple data centers is supported. Failed nodes can be replaced with no downtime. There are no network bottlenecks. Every node in the cluster is identical. Every node in the cluster has the same role. There is no single point of failure. Data is distributed across the cluster (so each node contains different data), but there is no master as every node can service any request. Read and write throughput both increase linearly as new machines are added, with no downtime or interruption to applications. Cassandra is not row level consistent, meaning that inserts and updates into the table that affect the same row that are processed at approximately the same time may affect the non-key columns in inconsistent ways. One update may affect one column while another affects the other, resulting in sets of values within the row that were never specified or intended.

Relational Databases: Unlike the other data stores, relational DBMSs have a complete pre-defined schema, a SQL interface, and ACID transactions. Traditionally, RDBMSs have not achieved the scalability of some of the previously described data stores. It appears likely that some relational DBMSs will provide scalability comparable with NoSQL data stores, with two provisions:

- Use small-scope operations: As we've noted, operations that span many nodes, e.g. joins over many tables, will not scale well with sharing.
- Use small-scope transactions: Likewise, transactions that span many nodes are going to be very inefficient, with the communication and two-phase commit overhead.

It should be noted that NoSQL systems avoid these two problems by making it difficult or impossible to perform larger scope operations and transactions. Relational Databases are:

- MySQL Cluster
- VoltDB

[60] http://cassandra.apache.org/

- Clustric
- ScaleDB
- ScaleBase
- NimbusDB

SQL vs NoSQL: Key Differences (SQL vs. NoSQL)

- One of the key differentiators is NoSQL supported by column-oriented databases where RDBMS is a row-oriented database.
- NoSQL seems to work better on both unstructured and unrelated data. The better solutions are the crossover databases that have elements of both NoSQL and SQL.
- RDBMSs that use SQL are schema–oriented which means the structure of the data should be known in advance to ensure that the data adheres to the schema. For example, predefined schema-based applications that use SQL include Payroll Management System, Order Processing and Flight Reservations.
- SQL Databases are vertically scalable – this means that they can only be scaled by enhancing the horsepower of the implementation hardware, thereby making it a costly deal for processing large batches of data.
- NoSQL databases give up some features of the traditional databases for speed and horizontal scalability. NoSQL databases on the other hand are perceived to be cheaper, faster and safer to extend a pre-existing program to do a new job than to implement something from scratch.
- More importantly, data Integrity is a key feature of SQL based databases. This means, ensuring the data is validated across all the tables and there's no duplicate, unrelated or unauthorized data inserted in the system.

Advantages of SQL databases are that they are typically more performant when dealing with more complex queries. Users cite the relational nature of SQL DBs as encouraging a well-structured database.

5.2.1.3 Big data processing

Hadoop and many of the frameworks based on it, are good for processing and analyzing data at rest, but not so much to deal with streaming data or data-in-motion. *Spark*[61], based in Hadoop, improves speed and resource consumption and it is open to use different programming languages (Java, Python, Scala and R). Spark comes with a ML library called MLib that provides ML methods for both batch and stream analytics. Spark is not purely a stream data analytics engine, as it uses micro-batches to simulate streams, but it is useful and fast in many situations. Other frameworks such as *Apache Flink*[62] have been developed to be native in-memory stream processing engines. Flink also comes with its own ML library, FlinkML, which provides support for many ML methods both for streaming and static data.

[61] https://spark.apache.org/
[62] https://flink.apache.org/

H2O[63] is another open source project that can be considered as a complete analytical product on its own. H2O provides a distributed processing engine, data pre-processing, analytics, math, ML libraries and evaluation tools. Similar to Flink, H2O processes data in memory and has its own ML library (Sparking Water). As well as Spark, it offers support for the same programming languages and it can execute Spark processes by providing integration with Spark processing framework through its ML library, or its own processing models on top of Spark and Storm.

The Apache project[64] provides many OS projects (49 listed under this category up to October 2019) dealing with data processing and analytics for multiple purposes. Among them it is worth mentioning *Apache Kafka*[65], *Apache Flume*[66] and *Apache NiFi*[67]. Kafka provides a powerful publish-subscribe mechanism to handle data streams. This can be very useful to get incoming streaming data (i.e. from sensors in the field) that can be consumed by one or several programs or applications. Flume is a distributed streaming system that is very useful to collect, aggregate, and move large amounts of data, as well as allowing analytic applications. Apache NiFi is a tool to create data-driven pipelines. It uses a graphical tool to design the dataflows which allow easy to configure, extensible and secure data processing and analytical pipelines. Previous projects worked on providing docker containers with all these frameworks and tools. In this scope is worth mentioning BigDataEurope[68], TOREADOR where ATOS provided more than 70 docker containers with combinations of many of these tools and frameworks, or QROWD[69], where the backbone of data processing is based on Apache NiFi pipelines developed by ATOS.

Based also in docker containers is the OS *Acumos AI*[70] project from the Linux Foundation. Acumos AI provides a framework to ease the process of building, sharing, and deploying AI models and apps. Acumos packages the most common AI frameworks, such as TensorFlow or Scikit Learn, and programming languages, such as Python, Java or R. Acumos is implementing an automatic onboarding process for AI models developed with these tools and languages to dockerize the models. It also provides a federated AI model marketplace to share and publicize the models. A potential use of Acumos in DEMETER could be precisely the creation of an AI model marketplace for the agriculture domain. An extra benefit of this approach is that Acumos is at the core of the European AI on demand platform currently under development in the scope of the AI4EU project. Due to the federated nature of Acumos, the AI model marketplace developed in DEMETER could be federated to the one in AI4EU, making the models accessible form the on-demand platform and vice versa. In the following sections, we describe frameworks for batch processing and stream processing.

5.2.1.3.1 Batch processing

[63] https://www.h2o.ai/products/h2o/
[64] https://projects.apache.org/
[65] https://kafka.apache.org/
[66] https://flume.apache.org
[67] https://nifi.apache.org/
[68] https://www.big-data-europe.eu/
[69] http://qrowd-project.eu/
[70] https://www.acumos.org/

Apache Hadoop - Apache Hadoop (Hadoop) is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS) and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

Apache Spark - Apache Spark (Spark) has as its architectural foundation the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines, that is maintained in a fault-tolerant way. RDD is a fundamental data structure of Spark. This processing tool is a programming abstraction that represents an immutable collection of objects that can be split across a computing cluster. Operations on the RDDs can also be split across the cluster and executed in a parallel batch process, leading to fast and scalable parallel processing. RDDs can be created from simple text files, SQL databases, NoSQL stores (such as Cassandra and MongoDB) among others. Much of the Spark Core API is built on this RDD concept, enabling traditional map and reduce functionality, but also providing built-in support for joining data sets, filtering, sampling, and aggregation.

Hadoop MapReduce - Hadoop MapReduce [60] is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically, both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Typically, the compute nodes and the storage nodes are the same, that is, the MapReduce framework and the Hadoop Distributed File System are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster. The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.

At their minimum, applications specify the input / output locations and supply map and reduce functions via implementations of appropriate interfaces and/or abstract classes. These, and other job parameters, comprise the job configuration. The Hadoop job client then submits the job (jar/executable etc.) and configuration to the JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them,

providing status and diagnostic information to the job-client. Although the Hadoop framework is implemented in Java, MapReduce applications need not be written in Java.

5.2.1.3.2 Stream processing

Apache Flink - Apache Flink (Apache Flink) is an open source stream processing framework developed by the Apache Software Foundation. The core of Apache Flink is a distributed streaming dataflow engine written in Java and Scala. Flink executes arbitrary dataflow programs in a data-parallel and pipelined manner. With its pipelined runtime system, it enables the execution of bulk/batch and stream processing programs. Furthermore, Flink's runtime supports the execution of iterative algorithms natively. Flink provides a high-throughput, low-latency streaming engine as well as support for event-time processing and state management. Flink applications are fault-tolerant in the event of machine failure and support exactly-once semantics. Programs can be written in Java, Scala, Python, and SQL and are automatically compiled and optimized into dataflow programs that are executed in a cluster or cloud environment. It is worth mentioning that Flink does not provide its own data storage system and provides data source and sink connectors to systems such as Amazon Kinesis, Apache Kafka, HDFS, Apache Cassandra, and ElasticSearch. In general, Flink: provides results that are accurate, even in the case of out-of-order or late-arriving data; is stateful and fault-tolerant and can seamlessly recover from failures while maintaining exactly-once application state; performs at large scale, running on thousands of nodes with very good throughput and latency characteristics.

Spark Streaming - Spark Streaming (Spark Streaming) makes it easy to build scalable fault-tolerant streaming applications. Spark Streaming brings Apache Spark's language-integrated API to stream processing, letting you write streaming jobs the same way you write batch jobs. It supports Java, Scala and Python. Stateful exactly-once semantics out of the box. Combine streaming with batch and interactive queries. Spark Streaming extended the Apache Spark concept of batch processing into streaming by breaking the stream down into a continuous series of microbatches, which could then be manipulated using the Apache Spark API. In this way, code in batch and streaming operations can share (mostly) the same code, running on the same framework, thus reducing both developer and operator overhead. A criticism of the Spark Streaming approach is that micro batching, in scenarios where a low-latency response to incoming data is required, may not be able to match the performance of other streaming-capable frameworks like Apache Storm, Apache Flink, and Apache Apex, all of which use a pure streaming method rather than microbatches. Instead of processing the streaming data one record at a time, Spark Streaming discretizes the streaming data into tiny, sub-second micro-batches. In other words, Spark Streaming’s Receivers accept data in parallel and buffer it in the memory of Spark’s workers nodes. Then the latency-optimized Spark engine runs short tasks (tens of milliseconds) to process the batches and output the results to other systems. Note that unlike the traditional continuous operator model, where the computation is statically allocated to a node, Spark tasks are assigned dynamically to the workers based on the locality of the data and available resources. This enables both better load balancing and faster fault recovery.

Kafka Streaming - Kafka Streams (Kafka Stream) is a client library for building applications and microservices, where the input and output data are stored in a Kafka cluster. It combines the simplicity of writing and deploying standard Java and Scala applications on the client side with the benefits of Kafka’s server-side cluster technology. Some benefits of Kafka include:

- Elastic, highly scalable, fault-tolerant

- Deploy to containers, VMs, bare metal, cloud
- Equally viable for small, medium, & large use cases
- Fully integrated with Kafka security
- Write standard Java applications
- Exactly-once processing semantics
- No separate processing cluster required

Apache Storm - Apache Storm (Apache Storm, 2016) is a free and open source distributed real-time computation system. Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop did for batch processing. Storm is simple, can be used with any programming language. Storm has many use cases: real-time analytics, online machine learning, continuous computation, distributed RPC, ETL, and more. Storm is fast: a benchmark clocked it at over a million tuples processed per second per node. It is scalable, fault-tolerant, guarantees your data will be processed, and is easy to set up and operate. Storm integrates with the queueing and database technologies you already use. A Storm topology consumes streams of data and processes those streams in arbitrarily complex ways, repartitioning the streams between each stage of the computation however needed.

5.2.1.4 Connectors

Kafka Connect - Kafka Connect (Kafka Connect) is a framework for scalable and reliably streaming data between Apache Kafka and other data systems. Connect makes it simple to use existing connector implementations for common data sources and sinks to move data into and out of Kafka. Kafka Connect's applications are wide ranging. A source connector can ingest entire databases and stream table updates to Kafka topics or even collect metrics from all of your application servers into Kafka topics, making the data available for stream processing with low latency. A sink connector can deliver data from Kafka topics into secondary indexes like Elasticsearch or into batch systems such as Hadoop for offline analysis. Kafka Connect is focused on streaming data to and from Kafka. This focus makes it much simpler for developers to write high quality, reliable, and high-performance connector plugins and makes it possible for the framework to make guarantees that are difficult to achieve in other frameworks. The main benefits of using Kafka Connect are:

- Data Centric Pipeline – use meaningful data abstractions to pull or push data to Kafka.
- Flexibility and Scalability – run with streaming and batch-oriented systems on a single node or scaled to an organization-wide service.
- Reusability and Extensibility – leverage existing connectors or extend them to tailor to your needs and lower time to production.

Spark Connectors - Both Spark and HBase are widely used, but how to use them together with high performance and simplicity is a very challenging topic. Spark HBase Connector (SHC) provides feature rich and efficient access to HBase through Spark SQL. It bridges the gap between the simple HBase key value store and complex relational SQL queries and enables users to perform complex data analytics on top of HBase using Spark. SHC implements the standard Spark data source APIs and leverages the Spark catalyst engine for query optimization. To achieve high performance, SHC constructs the RDD from scratch instead of using the standard HadoopRDD. With the customized RDD, all critical techniques can be applied and fully implemented, such as partition pruning, column

pruning, predicate pushdown and data locality. The design makes the maintenance easy, while achieving a good tradeoff between performance and simplicity. Furthermore, Spark is collaborating with MongoDB, Cassandra, Solr, Elasticsearch etc. in creating connectors between them (Spark packages). There is also the ability to define custom and purpose-built connectors depending on the needs (Connector Devel.). The community offers a big selection of implementations that work with specific technologies and tools. There also exists a Connect Developer Guide so as to create connectors if necessary although in general it is preferable to use already existing connectors and it is a solution when something specific is required.

5.2.1.5 Data pipelines for data aggregation and processing

Apache NiF[71] is a data platform developed by the Apache Software Foundation created to move data between different systems both in real time and scheduled. It offers a GUI where data pipelines can be drawn and programmed. It is very useful to integrate data coming from different sources and systems. NiFi's base unit of work is called a FlowFile, an encapsulation of the processed data that can be updated and routed using the so-called Processors. Some of the NiFi features are its high horizontal scalability (each flowfile can be processed in a different system), handling of back pressure if the amount of data is difficult to handle in real time and able to record provenance. NiFi provides many out-of-the-box data ingestors, connectors with many existing sources and systems (i.e. connectors to Apache Kafka and Apache Flume) as processors, which make the overall idea quite reusable and easier to use over time. On top of this engine, Apache NiFi provides the possibility to create custom processors. For instance, in DEMETER we could develop typical NiFi processors to handle typical agricultural data ingestion from known sources, as well as processors to handle data preprocessing tasks or even analytical tasks. This way data flows in DEMETER integrated in the NiFi platform will allow developers not to worry about coding most of the functionalities that NiFi offers because they are added seamlessly to the custom processor.

5.2.2 Data Analytics (incl. Machine Learning) in the agri-food domain

In this section the state-of-the-art of different available components involved in the data analytics in the agri-food domain are discussed.

- **Caffe** is a popular deep learning framework which can be used in various experimental concepts. In [62] and [63], the pre-trained deep CNN is utilized and combined with traditional features for leaf classification, achieving a dominant accuracy of at least 97% and outperforms the traditional method. Similarly, some groups exploit various DL architectures (ConvNet, AlexNet, GoogLeNet) to detect plant diseases after applying affine [64] and perspective [65] [66] transformation on the leaf imagery and exceeding 95% accuracy. The framework is not used solely on classification problems. CNNs can be applied in datasets for various computations, like the estimation of crop yields [67] and actually outperform classic algorithms like Support Vector Machine Regression (SVR).
- TensorFlow platform also gained a lot of ground recently and is expected to increase its share in research activities with the stable version that was recently released. In [68], they introduce Long Short-Term Memory (LSTM) networks by basically adding a level of feedback in the classic feedforward RNN model, which was enough to exceed the state-of-the-art

[71] https://nifi.apache.org/

performance in land cover classification. A modified Inception-ResNet architecture version is used over a typical CNN for crop yield estimation [69] exceeding the threshold of 90% in real image test accuracy. The application of deep CNNs is broad in the field of agricultural robotics, both on UAVs and UGVs, with modifications like deep lightweight models combination [70], [71], [72].

- **Keras and Theano** are two open-source libraries for Python that are usually deployed together and are specially created to run machine learning algorithms optimally. Modern earth observation is a field where computational needs are huge, as deep CNNs are used for land cover classification. A similar approach to the aforementioned one with LSTM networks is implemented using Keras and Theano [73],[74]. Another use of LSTMNs for improving traditional methods is for general plant classification [75]. In [76] the quality of vegetables during winter months is examined by processing radar images using deep Recurrent NN – based classifiers which outperformed the classical SVM and random forest classification algorithms. Theano-based *Lasagna* module also provides reliable solutions when implementing deep CNNs, as it offers a GPU-accelerated differentiation platform which allows faster gradient computation even for complex systems [77],[78].

Apart from the aforementioned popular frameworks and platforms, other tools make their way into machine and deep learning applications and implementations. A reliable choice for JVM languages, *deeplearning4j* is used for plant disease detection, successfully tested on banana leaves [79]. Pylearn2 is another library built mostly on top of Theano that can be used to implement deep CNNs, e.g. for plant classification [80]. Deep learning consists of mostly mathematical functions, so MATLAB is far from absent from the process. MatConvNet is a toolbox that implements CNNs for computer vision purposes. The problem of segmentation of root and soil from X-ray tomography can be effectively solved using these tools [81]. Deep Learning Toolbox provides similar aid in building variations of CNNs using MATLAB for classification purposes, as in cattle race detection [82].

Image Processing is vastly used for Quality of Services (QoS) applications and custom models are sometimes preferred against library-implemented solutions. In [83], Artificial Neural Networks are deployed in order to detect some kinds of diseases and grade fruit using image processing algorithms based on mathematical functions using features like color, texture and morphology. On quality grading as well, image processing is enriched with fuzzy logic rules in order to classify pineapples into quality classes [84]. A combination of IoT data and image processing is introduced in [85] for the assessment of plants' health, as soil and other data are used in a complementary Data manner with periodical image recording to evaluate the impact of possible changes in the environment. Smartphones are part of our daily routine, but they can also be a rich source of images to be processed, according to [86]. An irrigation sensor is buried near the roots of the crops and through a mobile app it is possible to compute the water contents of the soil. Weed monitoring is one of the biggest challenges for farmers and UAVs can provide the necessary imagery to handle this. A semi-supervised approach proposed in [87] proved very effective in crop row detection.

Last, classical machine learning algorithms are still popular and do not lack in accuracy. In [88], a variety of ML methods are compared in order to decide which should be used in a number of tasks, including obtaining missing values or forecasting future data. Algorithms like Linear and Polynomial Regression, KNN and Decision Trees can be evaluated for these tasks. The above algorithms

combined with a distributed file system like Hadoop can reduce the time and space needed to perform forecasting and other big data analytics, as highlighted in [89]. ANNs is another tool that is used for the estimation of food availability in sub-Saharan Africa [90]. Unsupervised learning is also a great asset in data-driven agriculture. K-means algorithm is applied [91] so as to identify the zones in which a cotton field should split for optimal management.

5.2.3 Explainable AI

While agriculture plays a critical role in the global economy, agri-technology and precision farming drive agricultural productivity, covering numerous aspects. The multitemporal remote sensing data, satellite imagery, and machinery data generated in modern agricultural operations enable a better understanding of the operational environment, e.g., an interaction of dynamic crop, soil, and weather conditions. Machine learning (ML) plays an important role in precision farming, leading to more accurate and faster decision making. Subsequently, the applications of ML have emerged, ranging from yield prediction, livestock management, animal welfare, livestock production, water management, soli management, species recognition, weed detection, weed detection, crop quality monitoring, to disease detection:

- Yield prediction is one of the most significant topics in precision agriculture and is of high importance for yield mapping, yield estimation, matching of crop supply with demand, and crop management to increase productivity. ML can help provide an efficient, low-cost, and non-destructive method, e.g., to automatically count coffee fruits on a branch by categorizing them into different categories (e.g., harvestable, not harvestable, and fruits with disregarded maturation stage) as well as measuring the weight and the maturation percentage of the coffee fruits.
- One of the most significant concerns in agriculture is pest and disease control in open-air and greenhouse conditions. The most widely used practice in pest and disease control is to uniformly spray pesticides over the cropping area. For example, ML models trained on satellite imagery are successfully applied for the detection and screening of Bakanae disease in rice seedlings, classification of parasites, and the automatic detection of thrips in strawberry greenhouse environments. Nevertheless, such systems are employed in the detection and discrimination of healthy Silybum marianum plants and those infected by smut fungus Microbotyum silybum during vegetative growth.
- Weed detection and management is another significant problem in agriculture. Many producers indicate weeds as the most important threat to crop production. The accurate detection of weeds is of high importance to sustainable agriculture, because weeds are difficult to detect and discriminate from crops in which ML algorithms in conjunction with sensors can lead to accurate detection and discrimination of weeds with low cost and with no environmental issues and side effects. In particular, DNN models trained on hyperspectral and multispectral images from unmanned aircraft systems have been able to identify Silybum marianum- a weed that is hard to eradicate and causes major loss on crop yield.
- ML played a key role in the automatic identification and classification of plant species in order to avoid the use of human experts, by reducing the classification time. For example, via leaf vein patterns, it is now possible to identify and classify different legume species, e.g., white beans, red beans, and soybean.

To solve these problems, both machine learning and deep learning-based approaches such as support vector machines (SVM), decision trees, tree-ensemble (e.g., random forest) self-organizing map (SOM), and deep neural network (DNN) are pervasive and widely used in literature. However,

with the availability of multimodal data (e.g., satellite imagery, sensor, and tabular data, or satellite imagery and received crop growth characteristics fused with soil data) usages of DNN architectures are getting very popular in smart and precision farming, because of their proven success and robustness to handle hyperspectral reflectance imaging, yielding more accurate prediction. Although approaches based on ML and DNN have shown promising success in precision farming, recent technological advances rely on accurate decision support systems that are often perceived as black boxes due to their overwhelming complexity and not well-understood internal functioning. They not only suffer from a lack of transparency but also cannot reason about their underlying decisions. This lack of transparency can lead to several technical, ethical, legal, and trust issues. In some other cases, the decision system may reflect unacceptable biases that can generate distrust.

The General Data Protection Regulation (GDPR), approved by the European Parliament in 2018, suggests that individuals should be able to obtain explanations of the decisions made from their data by automated processing, and to challenge those decisions. In particular, article 22 states that individuals “have the right not to be subject to a decision based solely on automated processing “and “whenever human subjects have their lives significantly impacted by an automatic decision-making machine, the human subject has the right to know why the decision is made," i.e., “right to explanation." All these reasons have given rise to the domain of interpretable AI, including in precision farming. In other words, the GDPR prohibits the use of ML for automated decisions unless a clear explanation of the logic used to make each decision is well explained. Although not every prediction made by an ML algorithm needs to be explained, in many cases the ML models itself must have interpretable logic embedded. Higher interpretability of an ML model means easier comprehension and explanation of future predictions for end-users. Interpretability is important to provide reasoning of the recommendations given by any agro-decision support system.

5.2.4 Data Quality

5.2.4.1 Definition

In an ecosystem with different stakeholders (as within the context of DEMETER), different relationships exist between the involved parties. In the following, our focus will be on the data related aspects. Some of these stakeholders will act as data suppliers, some as data consumers and/or data service providers and others might interact with a combination of these perspectives. This means that there may be a large number of interested parties who can benefit from the data provided for their individual application. On the other hand, an interested party may have to check the data of a number of different providers in advance with regard to their content and quality suitability for their own purpose.

In general, quality is an abstract concept and each person perceives and defines it differently depending on the point of view, i.e., it is often subjective [92]. Moreover, two views have been established in the field of engineering disciplines: (1) quality is the satisfaction of explicit as well as implicit needs of the user and (2) quality is the compliance with given specifications. [93]

- Data quality is the “degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions” [94].
- Data quality is the “degree to which data meets user requirements” [95].

Therefore, data quality is not an independent concept. It can only be assessed meaningfully by considering the intended usage of the data and the context, in which the data is applied. Thus, the starting point for a quality assessment are quality needs, which depend, among other things, on the user and his or her information needs (such as analysis questions).

5.2.4.2 Standards

In the following, we will list several standards available in the context of data quality:

- ISO/TS 8000
  - -1:2011, Data quality — Part 1: Overview
  - -2:2018, Data quality — Part 2: Vocabulary
  - -8:2015, Data quality — Part 8: Information and data quality: Concepts and measuring
  - -6x: Data quality management (2016-2019)
    - Part 60: Data quality management: Overview
    - Part 61: Data quality management: Process reference model
    - Part 62: Data quality management: Organizational process maturity assessment: Application of standards relating to process assessment
    - Part 63: Data quality management: Process measurement
  - -1xx: Master data quality (2009-2018)
    - Part 100: Master data: Exchange of characteristic data: Overview
    - Part 110: Master data: Exchange of characteristic data: Syntax, semantic encoding, and conformance to data specification
    - Part 115: Master data: Exchange of quality identifiers: Syntactic, semantic and resolution requirements
    - Part 116: Master data: Exchange of quality identifiers: Application of ISO 8000-115 to authoritative legal entity identifiers
    - Part 120: Master data: Exchange of characteristic data: Provenance
    - Part 130: Master data: Exchange of characteristic data: Accuracy
    - Part 140: Master data: Exchange of characteristic data: Completeness
    - Part 150: Master data: Quality management framework
  - -311:2012, Data quality — Part 311: Guidance for the application of product data quality for shape (PDQ-S)
- ISO/IEC 25000 series
  - 25012:2008 Data quality model
  - 25020:2019 Quality measurement framework
  - 25024:2015 Measurement of data quality
- ISO/TR 21707:2008 Data quality in ITS systems
- ISO 19157:2013 Geographic information — Data quality

For example, [94] defines elements required to specify data quality requirements and evaluating data quality. According to that standard, a *data quality model* provides a framework with a set of characteristics to address these both tasks. In general, data quality can be represented by different *data quality characteristics* (i.e., "categor[ies] of data quality attributes that bears on data quality" [94]). For example, the ISO Standard 25024: 2015 defines the following fifteen data quality

characteristics: Accuracy, Completeness, Consistency, Credibility, Correctness, Accessibility, Compliance, Confidentiality, Efficiency, Precision, Traceability, Understandability, Availability, Portability, Recoverability. Furthermore, *data quality measures* are defined to measure and assess these data quality characteristics. Data quality measure is a "variable to which a value is assigned as the result of measurement of a data quality characteristic "[94].

5.2.4.3 Approaches

The amount of data consistently increases and data is also collected within different (business) contexts, e.g., domains, processes, products, etc. In general, data is a key part of gaining and increasing business value. On the other hand, not all data is providing the same level of benefits and value. The impact of wrong results received from bad data can additionally be expensive and can lead to wrong consequences. So, data quality can impact the operative business processes as well as strategic decision making. For example, companies lose up to 25 percent of their operative gains [96] based on bad data quality or in other words, data quality issues can produce additional around 600 billion US dollars per year in the USA [97].

Thus, handling data quality is an important activity and there exist several approaches to support that. Batini et al. [98] provide an overview of different approaches and classify them based on different perspectives:

- *"*phases and steps that compose the methodology;
- strategies and techniques that are adopted in the methodology for assessing and improving data quality levels;
- dimensions and metrics that are chosen in the methodology to assess data quality levels;
- types of costs that are associated with data quality issues including:
    - *costs associated with poor data quality*, that is process costs caused by data errors and opportunity costs due to lost and missed revenues; these costs are also referred to as indirect costs;
    - *costs of assessment and improvement activities*, also referred as direct costs;
- *types of data* that are considered in the methodology;
- *types of information systems* that use, modify, and manage the data that are considered in the methodology;" [98]

Therefore, the authors distinguish three phases:

1. *State Reconstruction*: this phase captures the data collections, the organizational and contextual constraints under which the data is processed and collected, as well as quality issues and costs.
2. *Assessment*: in this phase, the quality of the data is measured and compared with reference values.
3. *Improvement*: this phase includes all steps, techniques and strategies that are used to achieve quality goals.

To briefly summarize the paper by Batini et al., the figure below provides an overview of the compared methodologies. The methodologies are distinguished between (1) *complete methodologies* ("which provide support to both the assessment and improvement phases, and

address both technical and economic issues"), (2) *audit methodologies ("*which focus on the assessment phase and provide limited support to the improvement phase"), (3) *operational methodologies* ("which focus on the technical issues of both the assessment and improvement phases, but do not address economic issues"), and (4) *economic methodologies ("*which focus on the evaluation of costs"). The name and references for each of these methodologies for data quality are provided in the following Table 1.

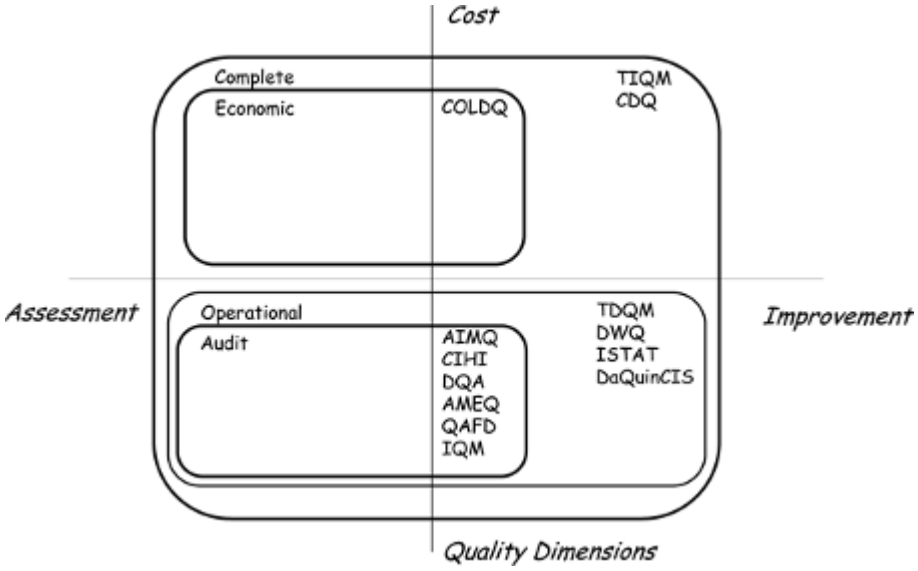


Figure 3: A classification of methodologies by Batini et al.

Methodology Acronym	**Extended Name**	**Main Reference (origin)**
TDQM	Total Data Quality Management	Wang 1998
DWQ	The Data Warehouse Quality Methodology	Jeusfeld et al. 1998
TIQM	Total Information Quality Management	English 1999
AIMQ	A methodology for information quality assessment	Lee et al. 2002
CIHI	Canadian Institute for Health Information methodology	Long and Seko 2005
DQA	Data Quality Assessment	Pipino 2002
IQM	Information Quality Measurement	Eppler and Münzenmaier 2002

Methodology Acronym	Extended Name	Main Reference (origin)
ISTAT	ISTAT methodology	Falorsi et al. 2003
AMEQ	Activity-based Measuring and Evaluating of product information Quality (AMEQ) methodology	Su and Jin 2004
COLDQ	Loshin Methodology (Cost-effect Of Low Data Quality	Loshin 2004
DaQuinCIS	Data Quality in Cooperative Information Systems	Scannapieco et al. 2004
QAFD	Methodology for the Quality Assessment of Financial Data	De Amicis and Batini 2004
CDQ	Comprehensive methodology for Data Quality management	Batini and Scannapieco 2006

Table 1: Methodologies considered by Batini et al.

5.2.5 Domain independent quality assessment and data cleaning

5.2.5.1 Summary

Previous research of Fraunhofer FIT staff and their collaborators contributes to the T2.3 objective of "filtering of irrelevant/outdated/low-quality data" on a general level – not specific to the agricultural domain but adaptable to any domain. This is achieved by languages for representing data quality metrics and the quality of a given dataset, and by tools that compute the quality of linked datasets.

Fraunhofer FIT is experienced with two tools for automatically assessing the quality of big linked datasets (i.e., RDF graphs), both providing implementations for many of the domain-independent linked data quality metrics collected in an earlier, widely cited survey [99]: the Luzzu quality assessment framework focuses on easy definition of domain-specific quality metrics, on supporting the complete quality assessment workflow, and on some aspects of scalability. SANSA emphasizes scalability and compatibility with big data architectures even more.

Besides, previously we developed some Python scripts to not just detect and report quality problems, but to actually fix certain inconsistencies (e.g., syntax errors, spaces in URIs, removing trivial literals, broken links/IRIs) in an RDF knowledge graph. These tools can be used to fix the inconsistencies before or after passing through the quality checking and assessment based on Luzzu or SANSA frameworks.

5.2.5.2 W3C Data Quality Vocabulary (DQV)

The W3C Data Quality Vocabulary (DQV)[72] supports the definition of custom quality metrics (on a descriptive, not computational level) and the annotation of datasets with their quality as measured by computing given metrics. On the level of entire datasets, DQV's ability to annotate is independent

[72] https://www.w3.org/TR/vocab-dqv/

of the data model/format; more fine-grained annotation support is available for datasets represented as RDF graphs, such as data in terms of DEMETER's AIM.

The main entities of the W3C Data Quality Vocabulary (DQV) include measurements of a dataset according to quality metrics, i.e., concrete means of measuring abstract quality *dimensions*, as well as annotations of datasets with the results of such measurements. The DQV reuses many existing standard vocabularies; for example, it represents quality measurements as observations in a data cube having the dimensions (not to be confused with the notion of "*quality* dimension") dataset, quality metric, and time [100], using the W3C Data Cube Vocabulary[73]. Figure 4 shows the overview of the DQV model.

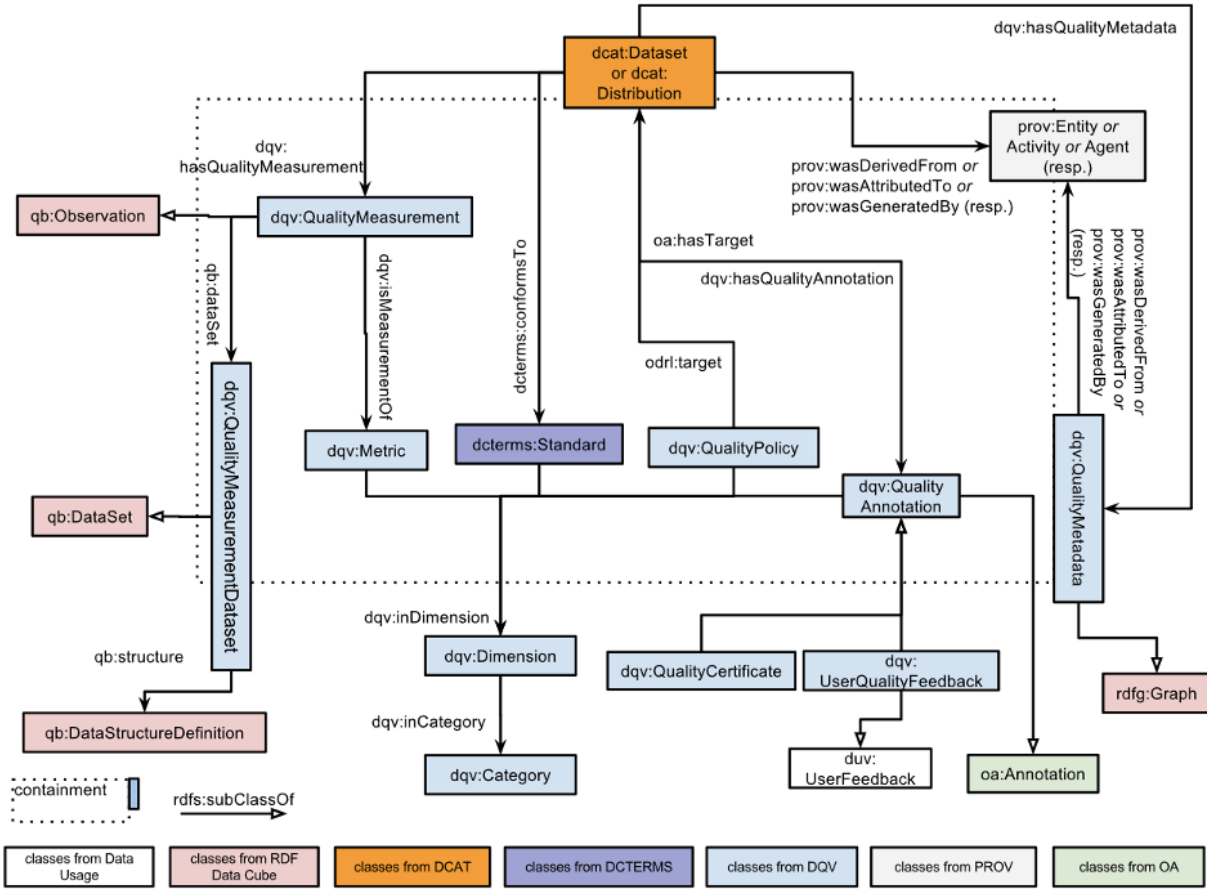


Figure 4: Overview of the W3C DQV data model

5.2.5.3 Luzzu Linked Data Quality Assessment Framework

The conceptual foundations of the Luzzu Linked Data Quality Assessment Framework[74] have influenced the W3C DQV [101]. Luzzu focuses on an easy definition at least of simple domain-specific quality metrics by domain experts without a programming background. Luzzu aims at supporting the complete quality assessment workflow; to this end it includes a web-based graphical frontend for reviewing the results of assessing different versions of a dataset over time w.r.t. the combination of

[73] https://www.w3.org/TR/vocab-data-cube/

[74] https://luzzu.github.io/Framework

different metrics. Luzzu achieves scalability, if the definition of the respective metric allows, by several means:

- by streaming the input dataset triple by triple,
- by distributed loading using Apache Spark (however, the available metrics' implementations are not currently optimized for distributed computing; therefore, see SANSA below), and
- by implementing probabilistic approximations of certain metrics [102]

Luzzu natively uses the Data Quality Ontology daQ as its data model; however, this predecessor of DQV can easily be converted into the latter.

5.2.5.4 SANSA Semantic Analytics Stack

The Semantic Analytics Stack (SANSA)[75] includes support for distributed quality assessment[76] of RDF graphs, based on Apache Spark and its concepts of data transformation and action[77] [103]. A comparative evaluation against Luzzu proves the high scalability and applicability of SANSA for quality assessment. As a result, SANSA can handle large-scale RDF knowledge graphs with billions of triples. Like the rest of SANSA, the quality assessment functionality is accessible through a web-based Zeppelin notebook interface (using the Scala programming language) and deployed as Docker containers conforming to the Big Data Europe platform. Further, SANSA quality assessment functionality can check for data availability, completeness, conciseness, and interlinking to ensure the consistency between RDF triples.

5.2.5.5 Semi-automatic quality assessment: cleaning and correcting RDF triples

Although SANSA's distributed quality assessment provides quality assessment metrics in a scalable manner, some issues cannot be tackled in order to ensure the quality of the RDF triples. For example, removing trivial triples, syntax errors, broken links, unnecessary space and non-escaped forbidden characters in URIs, missing or unmatched brackets, missing end of statement characters in between triples, missing end tags, wrongly placed spaces and characters etc. are important. Therefore, detecting and resolving syntactic errors in RDF knowledge graphs in a semi-automatic fashion is important, in case of large-scale graphs, where doing the same with manual human-intervention is difficult. Next two subsections provide some insights on how we'd overcome such issues as part of the data quality assessment.

5.2.5.6 Cleaning RDF knowledge graphs

In order to ensure the quality of a knowledge graph, often trivial information needs to be removed. For example, knowledge graph embeddings methods such as RDF2Vec, SimpleIE, TransE, KGloVe, CrossE, and ComplEx embed nodes and entities into a lower dimensional space. The model learns latent numerical representations of entities in an RDF graph, where the neighborhood of a node and the relations that exist to the neighboring nodes preserve the semantics. Such embeddings can be exploited for various tasks like link prediction, entity and document modeling, and content-based recommender systems. Since existing graph embedding methods do not incorporate literal

[75] http://sansa-stack.net/

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

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

information into the embedding, literals need to be removed from the KG9. Previously developed Python based tools to remove trivial triples didn't carry very significant information.

5.2.5.7 Fixing inconsistencies in RDF triples

Further, often RDF triples contain issues (bad string escape, misuse of keywords, bad language tags, bad local namespace in IRI, bad prefix label in IR, broken IRIs, spaces in URI, bad syntax in different RDF serializations, bad numbers as literal, bad string, bad directives etc.), which creates parsing by the RDF-triple stores. As a result, those triples (or the whole) won't be uploaded in the triple store for querying. Therefore, such broken links and inconsistencies will be fixed as part of the data quality assessment. Previously, we developed a Python based tool[78], which can fix inevitable syntax errors in RDF triples.

5.3 Data Protection, Privacy and Traceability

This section presents the State of the Art on keeping track of data provenance and the traceability of data as well as privacy and security relevant technologies including protocols for authentication and authorization to access data.

5.3.1 Data provenance

The pervasive nature of IoT raises serious security and privacy concerns, since highly sensitive information is exchanged continuously, even without user awareness. Personal data and individual identities are getting more and more vulnerable in a digital world with European stakeholders interacting in globalized scenarios. The on-going lack of trust derives from the current deficiency of solutions, including consistently applied technologies and processes for trusted enrolment, identification and authentication processes and, in particular, the use of online credentials with low levels of authentication assurance.

Managing the data provenance in these scenarios becomes vital. Data provenance is a process that determines the history of a data product, starting from its original sources. Assured provenance data can help detect access violations within the IoT infrastructure. However, developing assured data provenance remains a critical issue. Besides, provenance data may contain sensitive information about the original data and the data owners. Hence, there is a need to secure not only the data but also ensure integrity and trustworthiness of provenance data.

In that sense, when a privacy-preserving approach is needed, the data provenance metadata should allow unveiling the real identity of the owner associated with the exchanged IoT data, when the inspection grounds are met (e.g. identity theft or associated crimes). Besides, the data provenance information should be stuck to the data, enabling the tracking and auditing of it, wherever the data is stored or shared, in transient or at rest. However, currently there is a lack related to the application of proper privacy-preserving data provenance mechanisms for IoT scenarios that meets these requirements.

In this context, the work done on Towards Privacy Preserving Data Provenance for the Internet of Things [104], based on the ReliAble euRopean Identity EcoSystem (ARIES) [105] H2020 European research project, which aims to provide means for stronger and more trusted authentication, in a user-friendly and efficient manner and with full respect to data subject's rights for personal data protection and privacy, relies on ARIES mobile vIDs and Anonymous Credential Systems, to sign, in a privacy-preserving way, the IoT exchanged data, ensuring the ownership, anonymity, integrity and

[78] https://git.rwth-aachen.de/sb/rdfConverterCorrector

authenticity of the IoT data, prior its outsourcing. Concretely, using a Non-Interactive Zero Knowledge Proof (NI-ZKP), to sign the IoT data in a privacy preserving way, and then, outsourcing the data provenance metadata attached to the data.

Likewise, the rise of blockchain technologies has attracted interest due to a shared, distributed and fault-tolerant database where every participant in the network can share the ability to nullify adversaries by harnessing the computational capabilities of the honest nodes and information exchanged is resilient to manipulation.

The blockchain network is a distributed public ledger where any single transaction is witnessed and verified by network nodes. Its decentralized architecture makes it a possible solution for the development of an assured data provenance network. In the blockchain decentralized architecture, every node participates in the network for providing services, thereby providing better efficiency. Availability is also ensured because of blockchain's distributed characteristics.

5.3.2 Traceability requirements for data

Traceability has been a well-known aspect in numerous industries for decades. In logistics, traceability refers to the capability for tracing goods along the distribution chain, from the suppliers to the retailers. There are many benefits that come from track-and-trace. In the case of this project, it is crucial to count on it in the smart farm and agri-food value chain for letting all stakeholders know all the information about a product from farm to market. There are several solutions for tracing goods and their information, such as DLTs, CAS, bar coding, etc.

Regarding DLTs (Digital Ledger Technologies), many approaches have been studied [106], [107], [108], [109], [110], [111] for traceability in agri-food and in general, aiming to explore the advantages of having a cryptographically secure and immutable record of transactions. One of the solutions can be integrating the data into a public blockchain, which offers stakeholders a more transparent supply chain. It has the advantage of being globally accessible with real-time data and its analysis, providing relevant or useful information for all stakeholders (including for consumers, because they gain more confidence due to transparency and access to full information). This, in combination with IoT, can reduce redundancies in the supply chain. And, most importantly, in the case of a virus or bacteria outbreak, suppliers can identify and take care of unsafe products very quickly.

In the case of [112], the authors propose a food traceability solution using smart-contract token Ethereum, which covers the advantages mentioned above. Nevertheless, the drawbacks of this approach, according to the authors, are that any data chosen to be stored on the blockchain should not be sensitive or detrimental to the source, and that latency and the corresponding bottleneck fees are still a problem. There are also similar solutions using the Bitcoin blockchain [113] for supply chain management and discussions on using DLTs such as SOFIE for federated IoT [114]. Also, a blockchain solution that is gaining strength in the last years is Hyperledger, having implementations in Agri-Food supply chain management traceability [109]. Lastly, the DLT Directed Acyclic Graphs (DAG) also offer a scalable, fast and cheap solution for tracing, in opposition to possibly slow and expensive blockchain approaches [115]. Nevertheless, the problem with DAGs is that energy consumption still is not low enough to be implemented in IoT [116]. To overcome this issue with DLTs in general, there is currently work in progress on using an Ethereum gateway to act as blockchain node and then have a messaging mechanism with all the IoT devices [117].

Moreover, traceability is also crucial in this project regarding access control, to keep track of who enters what and when. In Demeter, all stakeholders will need to share information and analyses, and

that requires saving a log for possible audit logs or security breaches handling. In [118], they propose a traceable access control scheme with key and ciphertext delegation, to avoid the overhead on mobile computing, which would be useful for the case of IoT too.

Proposal and reference	Pros	Cons
UAV + Ethereum + RFID [106]	Process automation. Faster inventory than human operator. Able to estimate position of items.	Not mentioned.
DLT, specifically blockchain (SotA analysis) [107]	Cryptographically secure and immutable record of transactions. Improves speed and fidelity of traceability.	Need of data standardization in the food domain. Hard entry barriers for the food supply chain. No scalability (blockchain). No privacy mechanisms to protect users.
Blockchain + IoT, smart contracts, smart agriculture, RFID [108]	Trusted, self-organized, open and ecological food traceability system.	Not mentioned.
Ethereum, Hyperledger Sawtooth for Agri-Food [109]	Fault-tolerance, immutability, transparency and full traceability. **Ethereum**: scalability, reliability, mature, free if a private network is used **Hyperledger**: low latency, more appropriate for IoT, customization of records, fast implementations, easier integrations,	**Ethereum**: high latency, single language and fixed structure for smart contracts (constrained business logic), CPU-intensive (barrier for constrained devices). **Hyperledger**: low scalability, immature.
Agri-food, IoT, blockchain (Hyperledger Fabric) [110]	Multiple organizations in consortium chain -> reduced operating costs	Not mentioned.
Blockchain, agricultural supply chain [111]	Integration between the blockchain technology and IoT in order to improve the safety, security and quality of food products	Not mentioned.

Proposal and reference	Pros	Cons
Ethereum, smart contracts, IoT, food traceability [112]	Transparency and access to full information. Reduce redundancies in the supply chain.	Any data chosen to be stored on the blockchain should not be sensitive or detrimental to the source. Latency and the corresponding bottleneck fees are still a problem.
Bitcoin, blockchain, supply chain management [113]	Open, secure, tamperproof and completely decentralized.	Only theory, not a practical implementation. Transaction costs. Poor transaction performance.
Security and privacy challenges for IoT + DLT (analysis) [114]	DLT role should be diminished to that of a traditional trusted third party and/or for storing fingerprints of data, with smart contract oracles (critical data should be stored off-chain, in more traditional and separately protected systems, using open DLTs only to facilitate interoperability by providing distributed trust anchors).	Unwise to use (open) DLTs directly with IoT devices or for storing IoT related data.
RFID, DAG [115], [116]	Scalable, fast and cheap solution for tracing.	Energy consumption still is not low enough to be implemented in IoT.
Ethereum gateway (as node, smart contracts) + messaging mechanism with all IoT [117]	Independent of their computing and storage capabilities, it is possible to integrate a blockchain client to any IoT device.	Immature for now.

Proposal and reference	Pros	Cons
Traceable access control scheme with key and ciphertext delegation [118]	Avoids the overhead on mobile computing, which would be useful for the case of IoT. Performed key delegation without loss of the traceability with the same computation overhead.	Immature for now.

Table 2: Solutions and their pros and cons.

5.3.3 Authentication protocols

Authentication ensures that an identity of a subject (user or smart object) is valid, i.e. that the subject is indeed who or what claims to be. It allows binding an identity to a subject. The authentication can be performed based on something the subject knows (e.g. password), something the subject possesses (e.g. smart cards, security token) or something the subject is (e.g. fingerprint or retinal pattern). An authentication component enables authenticating users and smart objects based on the provided credentials. The credential can be in the form of login/password, shared key or digital certificate. As a result of the authentication process, an assertion is generated to be used afterwards, in order to declare that a specific subject was authenticated successfully by the Issuing authority. Below are the protocols maintained in authentication.

5.3.3.1 Open Authorization (OAuth)

OAuth is a scalable delegation protocol (i.e., you delegate someone to do something with somebody on your behalf). OAuth allows a user to permit access to an application to accomplish authorized tasks on behalf of the user [119]. Therefore, it allows a third-party program to gain restricted access to an HTTP service. This API authorization process can be securely implemented by a range of desktop, web and mobile applications. It introduces the concept of an authorization token that states the right of the client application to access authorized services on the server. Access to authorized services on the server is controlled using an authorization token. Nonetheless, it does not override any access control decisions that the server-side program may make. The OAuth 2.0 core authorization framework, described by IETF in RFC-6749 [120] defines the following specifications:

- OAuth 2.0 Core Spec describing the interactions between client, resource owner and server.
- OAuth 2.0 Core Spec and Bearer Spec describing the use of bearer tokens respectively. The bearer token is a large random number and a symbol of authorization. Since the number is large, then the probability of guessing the correct number is very small. This token is easier to process and use than the signature but requires SSL. Bearer tokens are the default type of access tokens.
- OAuth 2.0 MAC Spec describes the HTTP MAC access authentication scheme, an HTTP authentication method using a MAC algorithm to provide cryptographic verification of portions of HTTP requests. This token securely authenticates users without encrypting all traffic. Therefore, it is the most suitable option for APIs that require the security of OAuth and handle very large requests or responses where SSL is inefficient.

- OAuth 2.0 JWT Spec describes the use of a JWT Bearer Token as a means for requesting an OAuth 2.0 access token as well as for client authentication. JWT is a JSON-based security token encoding that enables identity and security information to be shared across security domains.
- OAuth 2.0 SAML Spec describes the use of a SAML2.0 Bearer Token (Assertion) as a means for requesting an OAuth 2.0 access token in addition to use as a means of client authentication. It is available in OAuth 2.0. It extends the support to the SAML-based operations. This facility of OAuth made it more popular among the SAML community and the universal open standard.

OAuth assumes four key roles in any authorization process:

- *Resource Server (RS)*: It hosts user data that is protected by OAuth.
- *Resource Owner (RO)/User*: It user of the application and owner of data.
- *OAuth Consumer/Client (OC)*: It application which makes an API request to get protected resources on behalf of the resource owner.
- *Authorization Server (AS)*: It authorizes the consumer after getting permission from resource owner and issues access token to the consumer for accessing protected resources available on the resource server.

OAuth offers the flexibility and leaves it up to server implementers to decide how the actual authentication and authorization are to be done.

5.3.3.2 OpenID Connect (OIDC)

OpenID Connect is a group of lightweight specifications that afford a framework for transmitting digital identity via RESTful APIs. OpenID Connect is seen as the evolution of OpenID 2.0, and is built as a profile of OAuth 2.0 rather than a completely distinct protocol foundation. OpenID Connect 1.0 is just another identity layer on the top of the OAuth 2.0 protocol. It facilitates clients to confirm the identity of the user depending on the authentication made by an Authorization Server, in addition to acquiring simple profile information about the user. OpenID Connect uses two main types of tokens: an access token and an ID token. The ID contains information about the authenticated user, and it is a JWT (JSON WebToken). This token is signed by the identity provider and can be read and verified without accessing the identity provider. OIDC assumes five key roles in any authentication and authorization process:

- *End User*: user of the application and owner of the information.
- *Relying Party (RP)*: application which makes API requests to get protected resources on behalf of the end user.
- *Authorization Endpoint (AE)*: only endpoint where the end-user needs to interact if they are not already logged in. It validates the identity of the end-user and obtains the consent and authorization from the end-user if the client has not been pre-authorized. It returns an authorization grant to the end-user or client depending on the use case. Sometimes, this authorization grant can then be passed in a request by the client to the token endpoint in exchange for an ID token, access token, and refresh token.
- *Token Endpoint (TE):* handles requests for retrieving and refreshing access tokens, ID tokens, refresh tokens, and other variables. It accepts a request from the client that includes an authorization code that is issued to the client by the authorization endpoint directly or via the end user. When the authorization code is validated, the appropriate tokens are returned in response to the client.

- *User Info Endpoint (UIE)*: OAuth 2.0 protected resource that the client application can retrieve consented claims, or assertions, about the authenticated end user. The client should present a valid access token to retrieve only those User Info claims that are scoped by the presented token.

OpenID also offers some flexibility in the implementation; however, it standardized many parameters such as instance scopes, endpoint discovery, and dynamic registration of clients, which were left up to implementers in the OAuth 2.0 implementation.

5.3.3.3 Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML) was developed by the Security Services Technical Committee of OASIS (Organization for the Advancement of Structured Information Standards) [121]. SAML is an XML-oriented framework for transmitting user authentication, entitlement, and other attribute information[79]. This framework provides two federation partners to select and share identity attributes using a SAML assertion/message payload, on the condition that these attributes can be expressed in XML. SAML assumes three key roles in any transaction Identity Provider (IDP/IdP), Service Provider (SP) and User:

- *Identity Provider (IDP/IdP)*: trusted organization that authenticates and authorizes users. It issues security assertion tokens for authentication and authorization services.
- *Service Provider (SP)*: organization that provides Web and other services. A SP relies on a trusted IDP for authentication and authorization services. It acts on information encoded in assertion tokens to determine whether a user is to be allowed access to a resource or not.
- *User*: entity that initiates a sequence of protocol messages and consumes the service provided by the SP. A user may be an application program that is requesting access to a resource.

The latest version of the SAML specifications is SAML 2.0, which describes the following components:

- Assertions state how identities are represented.
- Protocols represent a sequence of XML messages designed to achieve a single goal.
- Bindings describe how protocol messages are transported over a lower-level protocol such as HTTP.
- Profiles combine several bindings to describe a solution for a use case.

The SAML assertion is the main notion in SAML. It is a claim, statement, or declaration of a digital identity which is made by the IDP and trusted by the SP. The identity information required by the SP, is usually agreed in advance by the IDP and SP. However, there is a provision after the initial transaction to request additional information.

5.3.4 Authorization Protocols

5.3.4.1 Context-aware privacy policies

Given the pervasive, distributed and dynamic nature of IoT, context should be a first-class security component in order to drive the behavior of devices. This would allow smart objects to be enabled with context-aware security solutions, in order to make security decisions adaptive to the context in which transactions are performed. At the same time, context information should be managed by considering security and privacy considerations. Current IoT devices (e.g. smartphones) can obtain

[79] https://wiki.oasis-open.org/security/FrontPage

context information from other entities of their surrounding environment, as well as to provide contextual data to other smart objects. These communications can be performed through lossy networks and constrained devices, which must be secured by suitable security mechanisms and protocols. Additionally, trust and reputation mechanisms should be employed to assess the trustworthiness of data being provided by other entities in the environment. In this way, smart objects can discard information that comes from less reliable devices. Moreover, high-level context information can be reasoned and inferred by considering privacy concerns. Thus, a smartphone could be configured to provide information about a person's location with less granularity (e.g. giving the name of the city where s/he is, but not the GPS coordinates), or every long periods of time in order to avoid daily habits of that person could be inferred by other entities.

5.3.4.2 Capability-Based Access Control

The inherent requirements and constraints of IoT environments, as well as the nature of the potential applications of these scenarios, have brought about a greater consensus among academia and industry to consider access control as one of the key aspects to be addressed for a full acceptance of all IoT stakeholders.

The IoT ecosystem requires novel authorization mechanisms to tailor its limitations. The computing limitations of IoT devices and the distributed nature of IoT ecosystems must be considered by novel techniques for the access control of the IoT resources (i.e. sensing data and actuation operations). Currently, the most commonly accepted access control models, such as RBAC [123] or ABAC [124] have been widely used in a multitude of security scenarios. However, the suitability of these models for a distributed access control approach on IoT scenarios has not been demonstrated. In fact, the application of these models is not trivial since resource-constrained devices may not have enough processing resources to implement a complex access control mechanism. These models usually require a consistent definition of the meaning of roles and attributes, as well as complex access control policies, which makes challenging a straightforward application in resource-constrained devices.

Moreover, most of recent access control proposals have been designed through centralized approaches in which a central entity or gateway is responsible for managing the corresponding authorization mechanisms, allowing or denying requests from external entities. Since this component is usually instantiated by unconstrained entities or back-end servers, standard access control technologies are directly applied. However, significant drawbacks arise when centralized approaches are considered on a real IoT deployment. On the one hand, the inclusion of a central entity for each access request clearly compromises end-to-end security properties, which are considered as an essential requirement on IoT, due to the sensitivity level of potential applications. On the other hand, the dynamic nature of IoT scenarios with a potential huge number of devices complicates the trust management with the central entity, affecting scalability. Moreover, access control decisions do not consider contextual conditions which are locally sensed by end devices.

These issues could be addressed by a decentralized approach, in which IoT devices (e.g. smartphones, sensors, actuators, etc.) are enabled with authorization logic without the need to delegate this task to a different entity when receiving an access request. In this case, end devices are enabled with the ability to obtain, process and transmit information to other entities in a protected way. However, in a fully distributed approach, the feasibility of the application of traditional access control models, such as RBAC or ABAC, has not been demonstrated so far. Indeed, as previously mentioned, such models require a mutual understanding of the meaning of roles and attributes, as well as complex access control policies, which makes challenging the application of them on IoT

devices. Moreover, the impact of the potential applications of IoT in all aspects of our lives is shifting security aspects from an enterprise-centric vision to a more user-centric one. Therefore, usability is a key factor to be considered, since untrained users should be able to control how their devices and data are shared with other users and services.

As already mentioned, DCapBAC has been postulated as a feasible approach to be deployed on IoT scenarios [125], [17] even in the presence of devices with tight resource constraints. Inspired by SPKI Certificate Theory and ZBAC foundations, it is based on a lightweight and flexible design that allows authorization functionality to be embedded on IoT devices, providing the advantages of a distributed security approach for IoT in terms of scalability, interoperability and end-to-end security. The key element of this approach is the concept of capability, which was originally introduced by [126] as "token, ticket, or key that gives the possessor permission to access an entity or object in a computer system". This token is usually composed by a set of privileges which are granted to the entity holding the token. Additionally, the token must be tamper-proof and unequivocally identified in order to be considered in a real environment. Therefore, it is necessary to consider suitable cryptographic mechanisms to be used even on resource-constrained devices which enable an end-to-end secure access control mechanism. This concept is applied to IoT environments and extended by defining conditions which are locally verified on the constrained device. This feature enhances the flexibility of DCapBAC, since any parameter which is read by the smart object could be used in the authorization process.

In DEMETER, an innovative access control mechanism is proposed, according to the combination of several authorization technologies in order to enable a scalable solution for distributed IoT scenarios. Two technologies are used: (1) authorization policies for performing access control decisions, and (2) distributed access control tokens as an authorization technique to be validated by IoT constrained sensors, as well as ICT systems. DEMETER will provide two security mechanisms for distributed capability-based authorization based on access control policies presented in [17]. DEMETER will support capability-based tokens based on JSON format and ECC optimized signature to achieve distributed authorization of constrained IoT devices.

5.3.4.3 Capability Token

The format of the capability token is based on JSON. Compared to more traditional formats such as XML, JSON is getting more attention from academia and industry in IoT scenarios, since it can provide a simple, lightweight, efficient, and expressive data representation, which is suitable to be used on constrained networks and devices. A detailed description of Capability token is mentioned in the section “Distributed Capability-Based Access Control”.

5.3.4.4 DCapBAC scenario

In a typical DCapBAC scenario, an entity (subject) tries to access a resource of another entity (target). Usually, a third party (issuer) generates a token for the subject specifying which privileges it has. Thus, when the subject attempts to access a resource hosted in the target, it attaches the token which was generated by the issuer. Then, the target evaluates the token granting or denying access to the resource. Therefore, a subject which wishes to get access to certain information from a target, requires sending the token together with the request. Thus, the target device that receives such a token can know the privileges (contained in the token) that the subject has, and it can act as a Policy Enforcement Point (PEP). This simplifies the access control mechanism, and it is a relevant feature on IoT scenarios since complex access control policies are not required to be deployed on end devices.

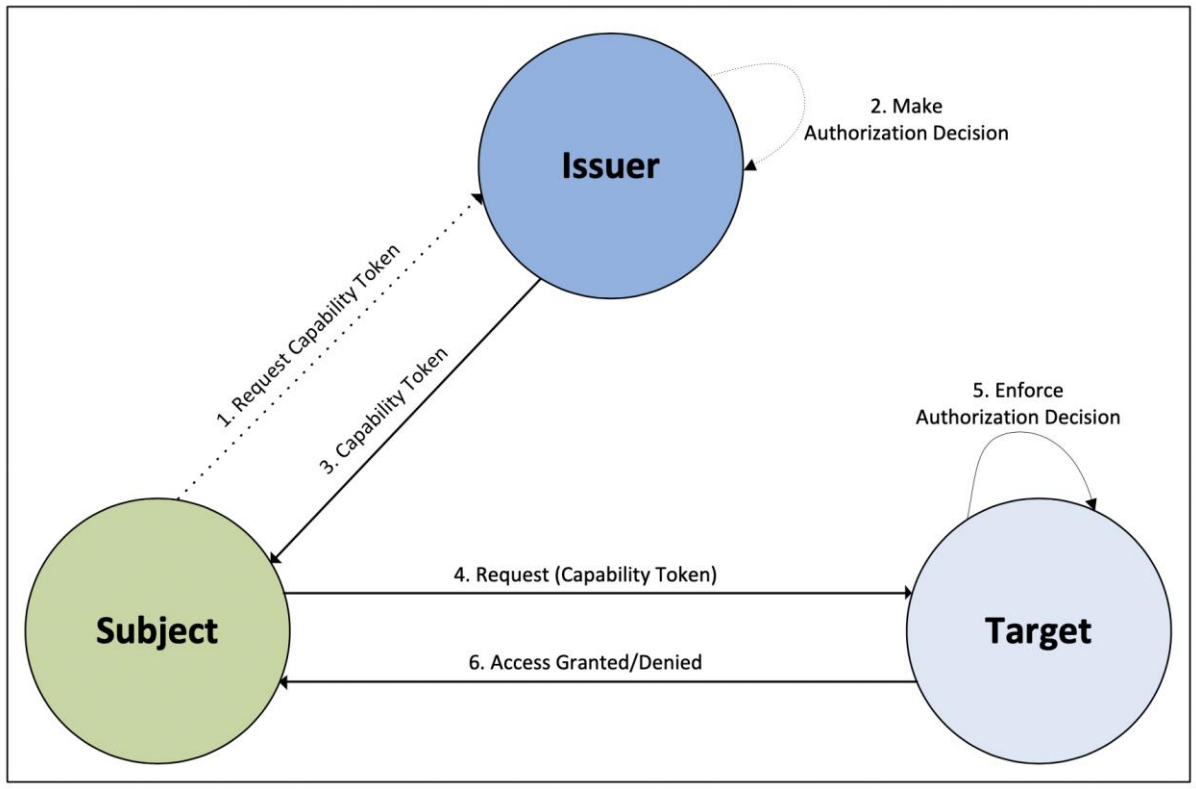


Figure 5: DCapBAC scenario overview

The basic operation of DCapBAC is shown in the Figure 5. As an initial step, the Issuer entity, which could be instantiated by the device owner or another entity in charge of the smart object, issues a capability token to the Subject to be able to access such device. Additionally, in order to avoid security breaches, such a token is signed by the Issuer. Therefore, in the case of a "Permit" decision, a capability token is generated with that specific privilege. In addition, XACML Obligations can be used in order to embed contextual conditions to be locally verified by the target device. Once the Subject has received the capability token, it attempts to access the device data. For this purpose, a request is generated in which the token is attached. This request does not have to be read by any intermediate entity. When the Target receives the access request, the authorization process is carried out. First, the application checks the validity of the token (i.e. if it has expired) as well as the rights and conditions to be verified. Then, the Issuer signature is verified with the corresponding public key. Depending on the specific scenario, this key can be delivered to smart objects during the commissioning or manufacturing process, or it can be recovered from a predefined location. Finally, once the authorization process has been completed, the Target generates a response based on the authorization decision.

Additionally, this approach provides support for advanced features, such as access delegation. In this case, a subject S (acting as a delegator) with a capability token CT can generate another token CT' for S' (acting as a delegated), in which a subset of the privileges of CT are embedded. Consequently, CT' can be used by S' to get access to a resource in a target smart object. Furthermore, S can grant the right to S' for additional delegations. This feature is valuable in order to address the dynamic and pervasive nature of IoT scenarios and everyday life. For example, elderly people can provide temporary privileges or delegation of them to home help personnel to get access to their homes in

case of an emergency. In the case of delegation, it is necessary to sign each new capability token with the corresponding subset of privileges, in order to allow a full auditability of access and avoid security breaches.

5.3.5 Privacy and Security By-Design Technologies

In DEMETER, we consider the following security/privacy by-design technologies: identity management and privacy-preserving group communication.

5.3.5.1 Identity Management (IdM)

For identity management, traditional solutions lack proper features to manage preservation of privacy and the minimal disclosure in a heterogeneous IoT environment. Traditional solutions based on credentials (e.g. X.509 certificates) require a centralized storage of the identity information in the service provider. In these cases, the service provider has all linked information about the users, and the users cannot control their private data to disclose in some contexts. To enable minimal disclosure, novel IdM technologies have been proposed to control partial identities in a private way based on the context in order to allow anonymity.

In DEMETER, IdM will provide novel technologies and operations for managing the secure access to the identity data in order to protect the privacy of the entities (i.e. smart devices and ICT services). IdM oversees controlling some entities' information such as identities, credentials and pseudonyms. IdM has interfaces to enable the modification of entities information from system administrators. For IoT environments, the IdM system enables distributed and scalable deployment to achieve high-performance with vast numbers of devices and identity data. Moreover, the IdM system must support limited computing resources to allow its deployment in constrained IoT gateways.

5.3.5.2 Privacy Group Sharing Communication

Dynamic IoT environment with heterogeneous entities needs distributed and scalable solutions for data exchanging based on privacy group sharing techniques. The reason behind is that interactions among IoT entities are usually based on short associations without the previous establishment of trusted links. Moreover, data exchanges must preserve the privacy of the involved entities in order to enable more flexible data sharing models. Traditionally, IoT solutions for constrained devices are based on Symmetric Key Cryptography (SKC). However, SKC needs that data producer and consumer must know a shared key. For this reason, these solutions are not able to allow enough scalability and flexibility in IoT networks with vast amounts of IoT devices and ICT services. To cope with this problem, Public Key Cryptography (PKC) was developed. But PKC requires high capacities in terms of memory and computing, so as the usage of specific certificates. Both SKC and PKC enable a data producer to encrypt information to be shared only by a unique consumer. The ubiquitous and distributed nature of the IoT environment requires privacy group sharing techniques to allow a data producer to encrypt information to be accessible by a group of consumers or unknown receivers. As an alternative to PKC certificates, Identity-Based Encryption (IBE) [127] was developed to enable data sharing with a group of consumers based on an identity string. In that sense, Attribute-Based Encryption (ABE) [128] was designed to extend the IBE string to a set of attributes according to the identity. In ABE, the information can be encrypted and accessible to a group of entities according to certain attributes, although their identities are likely unknown. The ABE scheme enables high scalability and flexibility in comparison with PKC and SKC schemes. In ABE, there is an Attribute Authority (AA) that controls the cryptographic credentials based on sets of attributes.

In the DEMETER system, a novel scheme called Ciphertext-policy Attribute-based Encryption (CP-ABE) [129] will be integrated. This enables that ciphertext can be encrypted according to a policy of

attributes, meanwhile the credentials of involved entities are associated with groups of attributes. So, data producers do not have to control the dissemination of the encrypted information to consumers, while consumers can access the information according to credentials based on their authorization attributes. In addition, this CP-ABE scheme can be employed in constrained IoT devices by the combination with Symmetric Key Cryptography (SKC). In CP-ABE, the information can be encrypted with a symmetric key according to a specific policy based on a set of attributes of consumers. CP-ABE can rely on Identity Management (i.e. anonymous credentials) where entities can request private keys based on their attributes. So, only consumers with the attributes defined in the policy can obtain the private key to decrypt the information. If consumers have high-constrained sensors, then SKC encryption and decryption functions can be performed by more powerful devices (i.e. trustworthy gateways).

5.3.6 Risk analysis and estimation

Currently, organizations have an increasing dependence on information systems, which makes them essential. Thus, being indispensable turns them into the aim of direct or unintentional attacks, and these incidents can jeopardize the mission of the organization. The way for different entities to organize and make decisions on how to act towards them is performed by risk analysis and risk management. Currently, there are standards and good practices on this matter, both at national and international levels. On the latter, the most used one is ISO 27005 - Security techniques - Information security risk management. Risk management aims to reach a realistic knowledge about circumstances that could affect processes or services, causing damage or losses. It allows priorities and security requisites to be established to cope with those situations. For that end, risk management relies on risk analysis, that is, the process that permits identifying, studying and evaluating, through the implicated variables, the potential events that could affect the objectives of an organization, as well as their consequences.

Risk analysis and estimation have been extensively studied in general, but also several works have been carried out focusing on the IoT paradigm. On [130], they propose a distributed risk management system for IoT specifically for water management. On [131], they develop a risk analysis for a smart home automation system, which could be adapted to general IoT. In combination with access control, in [132] they create a model that performs a risk analysis to estimate the security risk associated with each access request and uses the estimated risk to make the access decision. To make matters more automatic, in [133] they propose a methodology aimed at automating the threat modeling and risk analysis processes for an IoT system. It relies upon an open catalogue (from EU projects), for gathering information about threats and vulnerabilities. Moreover, using standard methodologies is also necessary to aim for interoperability among systems and that includes risk assessment interoperability. For that end, in [134] they offer a standardized model that includes a design process with risk assessment vectors, specific for IoT cyber risk, complemented by a comparative empirical analysis of multiple cyber risk assessment approaches [135]. The same organization completes this approach by defining design parameters for a decision support system for visualizing cyber risk of an IoT supply chain [136].

6 Technical requirements (all, previous contribution)

This section presents the technical requirements for specific tasks in WP2 from DK3 to DK7, which are:

- **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 presented in the next subsections. This template will be extended by an additional field to capture the relevant Use Cases, once the pilot use cases are finalized.

6.1 Data integration: Semantic Interoperability/integration Requirements

Requirement ID	DK3.1	**Version**	0.2	**Last Update Date**	12/12/2019
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). 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. • Mobile crowdsourced data collection tools (see comments) • IoT and sensor management systems, collecting real-time data from IoT devices (sensors and weather stations)
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	No reference to “Mobile crowdsourced data collection tools” either in D5.1 or in the DOW Removed some innovations, which is not directly addressed by this requirement (such as 8 and 9)

Requirement ID	DK3.2	**Version**	0.2	**Last Update Date**	12/12/2019
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 take into account 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 • Mobile crowdsourced data • Statistical data
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.				
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				
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				
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				
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				
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.				
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				
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.				
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				
Comments/Remarks	This requirement merges DK3.7,8,9				

6.2 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.				
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).				
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.				
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**	0.3	**Last Update Date**	25/01/2020
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, MQTT), as well as using standard input and output formats for services (e.g. JSON or XML).				
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.				
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.				
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.				
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.				
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	DK4.09	**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.				
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.				
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	DK1.11	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.				
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	13/12/2019

				Date	
Title	Data synchronization (batch and real-time)				
Description	DEMETER needs to guarantee data synchronization in batch, and should guarantee data synchronization in real-time.				
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).				
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				
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.				
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.				
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.				
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.				
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.				
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.				
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.				
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: o Sum o Average o Maximum o Minimum o Count o Count distinct ● The system must implement advanced functions that allow data to be aggregated: o Windowing Functions o Ranking functions ● The system should persist the aggregated value returned to the query in a database				
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				
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				
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	

6.3 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.				
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.				
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.
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]				
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.				
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.				
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.				
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**	0.1	**Last Update Date**	03/12/2019
Title	Perform measures/actions based on assessment results				
Description	Data quality components should, based on the assessed data quality (the "status of data"), perform measures to improve the data quality or provide quality assessment results to the analysis component that shall act on it.				
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	

Requirement ID	DK5.12	**Version**	0.1	**Last Update Date**	03/12/2019
Title	Support selection of appropriate data analysis techniques				
Description	Data quality components should support the selection of appropriate data analysis techniques. E.g., predictive algorithms running on raw data received from platform systems and devices should foresee the quality of data in order to extract a specific data model.				
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 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					

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.				
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).				
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).				
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 metrics for measuring data quality				
Description	Data quality components should utilize the appropriate quality metrics for each type (or source if appropriate) of data; e.g. different quality metrics might apply to image data and others to weather data etc.				

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.				
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).				
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.				
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.				
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)				
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.				
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.				
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.				
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					

6.4 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.				
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.				
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.				
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	12/12/2019
Title	Support for Exploratory Numerical Data Analysis				
Description	DEMETER Data Analytics must support methods for exploratory data analysis (EDA) on given data- or querysets independent of the deployed software solution. This can either be achieved by granting specific secure (i.e. guaranteeing that the data is accessed only by those roles eligible) access to the DEMETER data storage layer based on a distinct 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 queryset with different methods (e.g. Pearson, Spearman)				
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**	27/1/2020
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 queryset features/fields between analyses of business domains that do not share the exact same namespace but, in reality, share comparable concepts (e.g. analyses 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.				
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					

6.5 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.				
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				

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.				
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**	0.1	**Last Update Date**	09/12/2019
Title	Tools/application requirements: Encryption should begin at sensor level.				
Description	Each installed sensor shall be equipped with lightweight cryptography in order to assure encryption on such low level, as well.				
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	0.1	Last Update Date	09/12/2019
Title	Tools/application requirements: Secure way to handle resource constrained devices.				
Description	Resource constrained devices must communicate in a secure way.				
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	0.1	Last Update Date	09/12/2019
Title	Tools/application requirements: Monitoring of intrusion detection.				
Description	Lightweight cryptography on sensors shall allow detection and monitoring on intrusions.				
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.				
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.				
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.				
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.				
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				
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 authorization, traceability: Secure transport layer for authn/authz				
Description	The transport layer for all matters regarding authentication and authorization must be secure.				
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.				
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**	11/12/2019

				Date	
Title	Authentication and authorization, 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.				
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 authorization, 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.				
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 authorization, traceability: Appropriate traceability for heterogeneous datasets				
Description	It is necessary to establish suitable traceability for heterogeneous datasets.				
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.				
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.
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.				
Relevant Pilot(s)	ALL				
Relevant Task(s)	T2.4				
Relevant Objective(s)	1				
Relevant Innovation(s)	9. Secure Agricultural data sharing services				
Involved	Developers				

stakeholders/actors	
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.				
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.				
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.				
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	0.1	Last Update Date	11/12/2019
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.				
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	

7 Data & knowledge handling in DEMETER architecture

In this section, we present a synopsis of the data and knowledge handing infrastructure provided by the DEMETER Reference Architecture (RA). The DEMETER RA aims to facilitate the collection, processing and usage of the data used by DEMETER enabled pilots and to enable the capabilities regarding the extraction tools that are needed as identified by the technical requirements presented in the previous section. These requirements are handled by the components regarding the data management, fusion and analytics presented in this deliverable (starting from the next section), and it is the job of the RA to facilitate the integration of all these components into the DEMETER system.

As described in detail in deliverable D3.1, the DEMETER RA is a modular architecture using an overarching approach that 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. An important part of this process is the 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. The AIM common data model has been created to be interoperable with (and using/aligning data models from) a number of well-known ontologies and systems, such as FIWARE, Saref4Agri, FOODIE etc. For more information about the common data model and semantics, refer to deliverable D2.1.

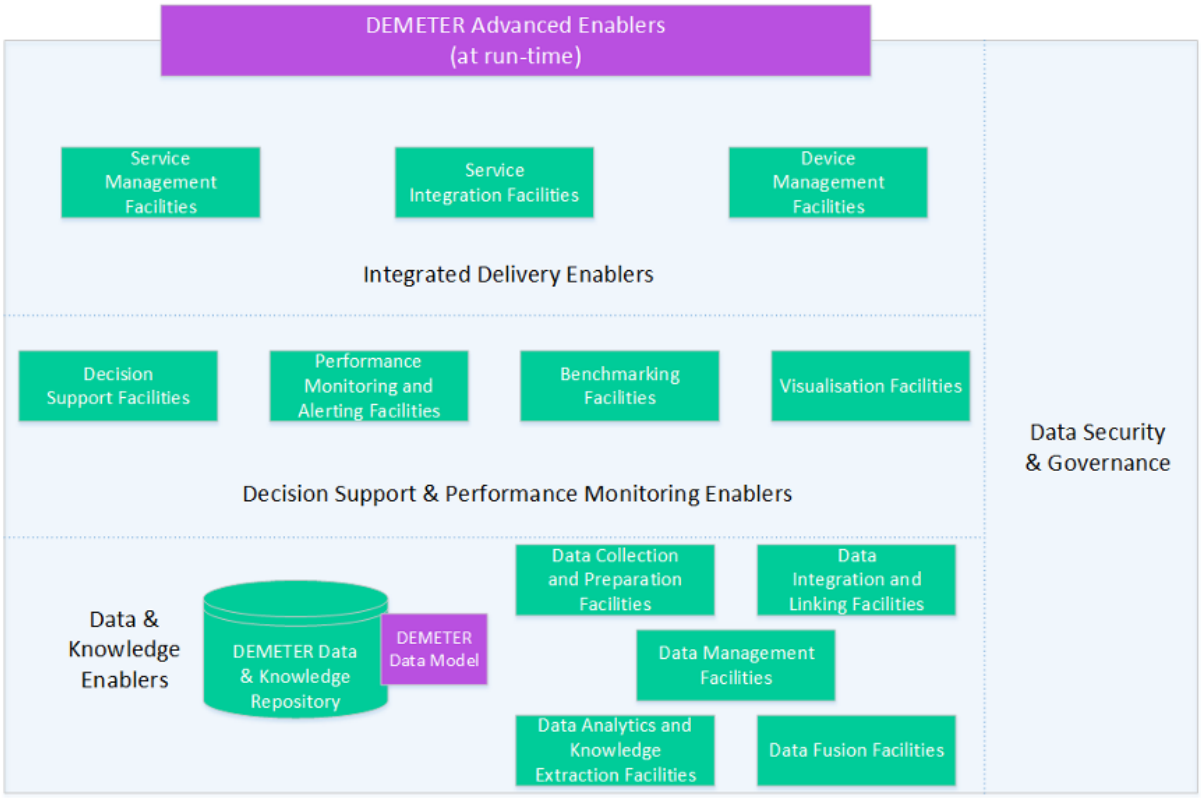


Figure 6: Advanced Enablers offered by DEMETER (Figure 35 in D3.1)

DEMETER offers a set of Core Enablers needed for creating any DEMETER applications that are mandatory for any interested stakeholder who wishes to expose or share her own resources. DEMETER also offers another type of enablers: Advanced Enablers that are optional and are discoverable and accessible through the Hub. They are depicted in the figure above and fall under several different categories. The Data & Knowledge Enablers presented in this deliverable 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. More specifically, the Data Collection & Preparation enablers collect, curate and prepare the data obtained, while the Data Integration & Linking together with the Data Fusion enablers integrate and fuse the data collected from heterogeneous sources. Furthermore, Data Management is guaranteed according to the users' stated preferences, while Data Analytics & Knowledge Extraction enablers are made available to any apps developed allowing for further processing of the fused data. Finally, split among the Core and the Advanced Enablers (vertical layer on the right) there lie the security protection facilities, which aim for example to ensure secure transfer of sensitive data or to prevent access to unauthorized entities. This document elaborates on all aforementioned enablers.

The DEMETER RA enables a data representation that is indeed linked to a series of relevant concerns, such as allowing interoperability between internal and external processes in DEMETER. Other concerns include 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.

Furthermore, the definition of the scope and requirements as well as a clear understanding of the needs from the system is the first step in order to build a model that supports all platform functionalities. The model is used to implement the entities that will enable the internal and external processes of the DEMETER Reference Architecture with the possibility of being supported in the representation of AIM through the use of a modular, flexible and extensible approach. Given these assumptions, different kind of data spaces have been identified in the RA and defined which will have to support, on the one hand, the interoperability between the resources generated by the DEMETER Enablers, on the other, the need to support the representation of the Stakeholders who will take part in the platform. The identified data spaces prescribed by the RA are a semantic database or **DEMETER Data & Knowledge Repository**, and a DEMETER **User Registry & DEMETER Resource Registry**, as shown in Figure below[80]. The data part of the RA follows an iterative approach, where complete and comprehensive descriptions of the relationships among the identified entities are refined; and in this deliverable we are able to do just this by defining the specific tools and components that have been developed (as outlined in the remaining sections of this report). It also focuses on the point of view of the Stakeholders (i.e., users accessing the information via the DEMETER Dashboard), as they are the main consumers (can also be suppliers too) of services and data.

[80] This figure is initially presented in D3.1 (Figure 41), and presents the main data flows between the various components highlighting also the stakeholders involved.

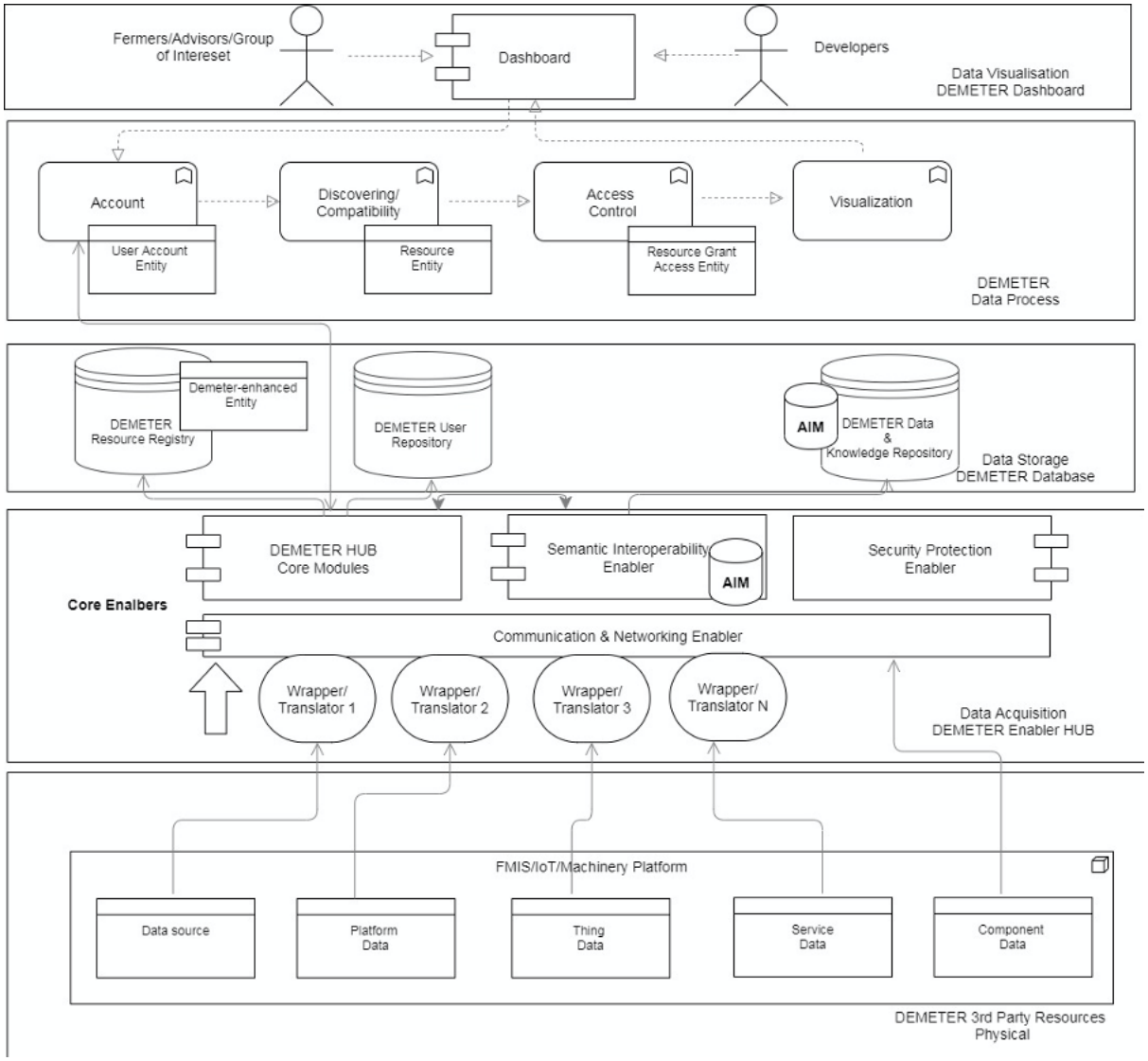


Figure 7: DEMETER Main Data Flows (Figure 41 in D3.1)

In more detail, the **DEMETER Data & Knowledge Repository** carries any data or extracted knowledge that may eventually be stored by DEMETER locally. It is based on the DEMETER AIM described in detail in D2.1 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** carries information concerning the DEMETER user accounts, such as personal data, credentials, and so on, while the **DEMETER Resource Registry** records all resources (i.e., data (sources), things, services or even entire platforms) that are registered to DEMETER by their owners (or parties responsible for making them available and exploiting them). Of course, security matters and protection of the data stored or references in the aforementioned DEMETER repositories or registries need to be carefully addressed. The respected data security mechanisms are outlined in Section 12 of this deliverable.

As depicted in Figure 7, 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.

As already mentioned, the data acquired and translated to AIM can also be used internally by DEMETER itself. The data encapsulated in a DEMETER enhanced-entity format which contains the semantic description, the metadata of each platform, thing, service or application are made available through the DEH APIs to all DEMETER Enablers. Selected data operations are made available to external stakeholders via the DEMETER Dashboard (UI) offering also data visualization facilities.

Central to the data handling facilities of the DEMETER RA are modules enabling the data preparation, integration, fusion and analytics; allowing also for data quality handling and data security protection. The respective modules and mechanisms are presented in the following sections of this deliverable.

8 Data management components

8.1 Overview

DEMETER will supply a series of software components or enablers dedicated to data management, to allow interoperability between and with modules/enablers which will be made available at the Pilots level. Data Management (DM) integrates architectural practices and techniques and the tools necessary to obtain consistent access and delivery of data to meet the data consumption requirements of all applications and activities processes. Data management is also an enabling discipline, in which technology and business work together to ensure uniformity, accuracy, management, governance, semantic consistency for data in an application or set of applications. The data represent the whole consistency of attributes managed and/or used within an application or information system. Examples of such entities in DEMETER include users, services, applications, dataset (business or related to specific processes in the Pilots).

DEMETER data management should allow a constant flow of data, either without interruptions or a recovery capacity if the communication with the data source is interrupted abruptly, and load capacity by the central components of these systems or technologies. DEMETER Enablers for data management will rely on these physical characteristics to respond to Pilots who intend to store their resources in the DEMETER Enabler HUB. Before analyzing the functional solution of the data management system, however, we must consider the DEMETER context. The aim of many of the involved Pilots in DEMETER is to increase productivity by acquiring and interpreting data relating to climate, weather, soil, water quality and crop status. Small and large farms use information systems to improve crop management and increase productivity. These systems include sensors for crop monitoring and for tracking agricultural products from production to final distribution. This resulted in a considerable increase in data. In this context, which poses many challenges such as the organization and management of the farm, product traceability, environmental requirements, decision support to increase performance, data synchronization between sensors and agricultural machinery, and systems that are able to interpret them represents one of the fundamental aspects of a system that must be able to support them.

The DEMETER project is very ambitious: the common approach to data management will embrace a myriad of technologies of communication from the underlying Pilot systems, offering the existing DEMETER's actors an interesting set of tools and technologies to strengthen their business and

innovation. Federated data management for the Agri-food sector as well their requirements has been imposed by many Pilot sites very heterogeneous with each other both in the facts (multiple technologies and systems) and in the objectives (companies that feed different businesses in different Agri-food sectors). The result is therefore clear and evident: a simple data management solution is not sufficient to meet the technological needs of each DEMETER Pilot site. Indeed, the data management must be connected and harmonized to the ability to analyze that a well-structured functional solution (before) and covered by adequate technologies (after) could support and manage these needs. Therefore, in this section of the document a complete data management solution for DEMETER is defined (which will be completed with regard to some technical specifications in D3.2), which includes design strategies through UML diagrams, to support data management from the Pilots to the DEMETER Enabler HUB.

DEMETER data management Enablers will therefore assume the following characteristics:

- will allow the storage, access, processing and delivery of data taking into account all the specifications of the DEMETER Pilot sites;
- will not represent a specific type of technology, but will make use of a complete set of components made up of many different technologies in combinations able to face the great heterogeneity and complexity of the project. In essence, show the ability to provide access to data management through open access tools;
- will support the availability of data to DEMETER front-end applications or enablers independently by including mechanisms to isolate the workload requirements and control various end user parameters;
- will support the originality in data formats, using common technologies capable of standardizing them according to standard protocols and common information model or AIM;
- finally, it will ensure the high configurability of these components, guaranteeing common technologies that comply with the guidelines and design in activities such as provisioning and delivery of data management software.

Finally, there is no doubt that having an overview of the information lifecycle management in DEMETER can represent at least in part introduction of the data system is a fundamental point, as it not only identifies the ways in which to ensure correct storage of data in a technologically infrastructure advanced as a DEMETER, but it helps to reduce the complexity of exposing how this system actually manages data management. It comes below provided a general view on the meaning and on some fundamental concepts of Data Lifecycle management, focusing attention on the approach used in the project through a holistic representation of the components or functional modules that will contribute to forming this system.

The ILM (Information Lifecycle Management) strategy that focuses mainly on the management of physical storage systems and on applying certain management techniques to information (as described at the Wikipedia page[81]) is insufficient for modern systems, since it does not take into account certain aspects such as property, value or efficiency of the information. Instead, a DLM (Data Lifecycle Management) strategy that considers these aspects as well as the flow of information throughout its lifespan will be applied.

The points covered by the Data Lifecycle Management that DEMETER will use are the following:

- Data generation/gathering.

[81] https://en.wikipedia.org/wiki/Information_lifecycle_management

- Data storage/upkeep
- Data usage/security
- Data final disposition

Figure 8 shows an outline of such DLM:

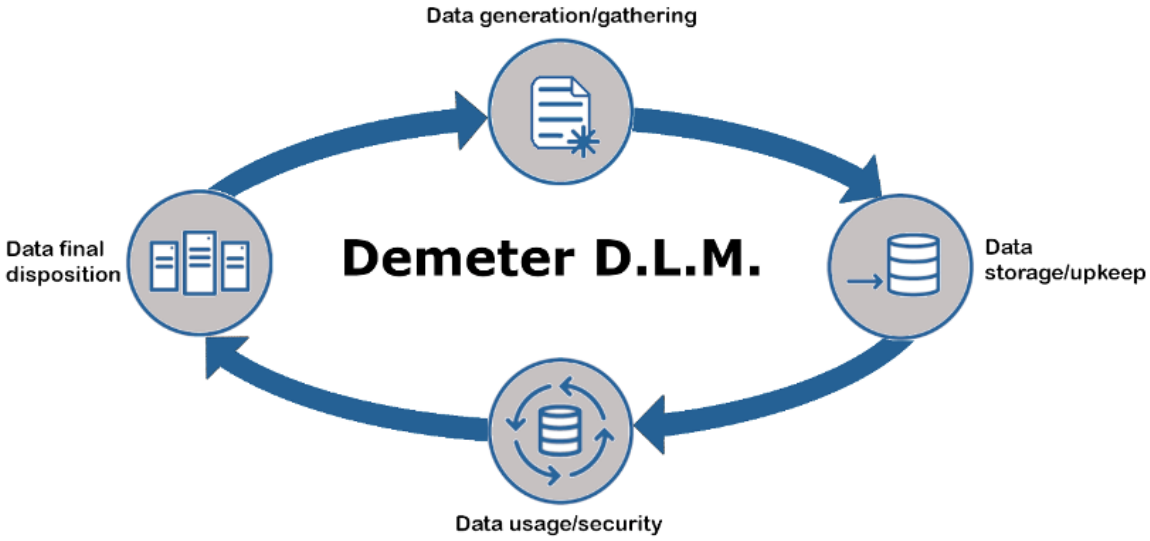


Figure 8: DEMETER Data Lifecycle Management Schema

The first step of the D.L.M. process is to perform the data gathering. Projects participating in DEMETER generate very heterogeneous data from a wide variety of sensors / devices, as can be seen in the following list:

- Irrigation control devices
- Meteorological stations
- ERP´s
- IoT sensors
- Electric Valves Control
- Drones
- GPS Tracking devices
- Pest Control traps
- Satellite TimeSeries
- Third party devices

These devices may be from third parties or open source so, to achieve a smooth exchange of information, DEMETER will implement its Enabler HUB to offer a set of services/interfaces (described in detail in D3.2) to allow data flow between all the stakeholders.

The Enabler HUB is the entry point to the Agricultural Interoperability Space (AIS) which will implement a semantic interoperability system that will allow data sent through these services/interfaces to be normalized and used. Such system, after analyzing in deep different protocols/data models as the FIWARE NGSI, NGSI-LD, FIWARE Agri-food, INSPIRE, rmAgro or Saref4agri, will be based in the AIM model as described in the deliverable D2.1, which uses JSON-LD format for data serialization. That model will be utilized to standardize the data in a homogeneous way so that it can be used by DEMETER. This normalization process will be performed taking into account the quality/consistency of the datasets and, to this end, DEMETER will use tools such as:

- LUZZU quality assessment framework
- SANSA Semantic Analytics Stack

These tools provide many quality assessment metrics in charge of correcting errors, e.g. syntax errors, forbidden characters, wrong triples, missing tags/brackets, broken links, etc. Thanks to this approach, data from projects can be corrected/adapted to DEMETER's needs, improving the overall result of its usage thus avoiding costs associated with poor data quality.

Figure 9 shows the data relationship between the projects and DEMETER through the services/interfaces as well as the data standardization process that will be performed to all datasets:

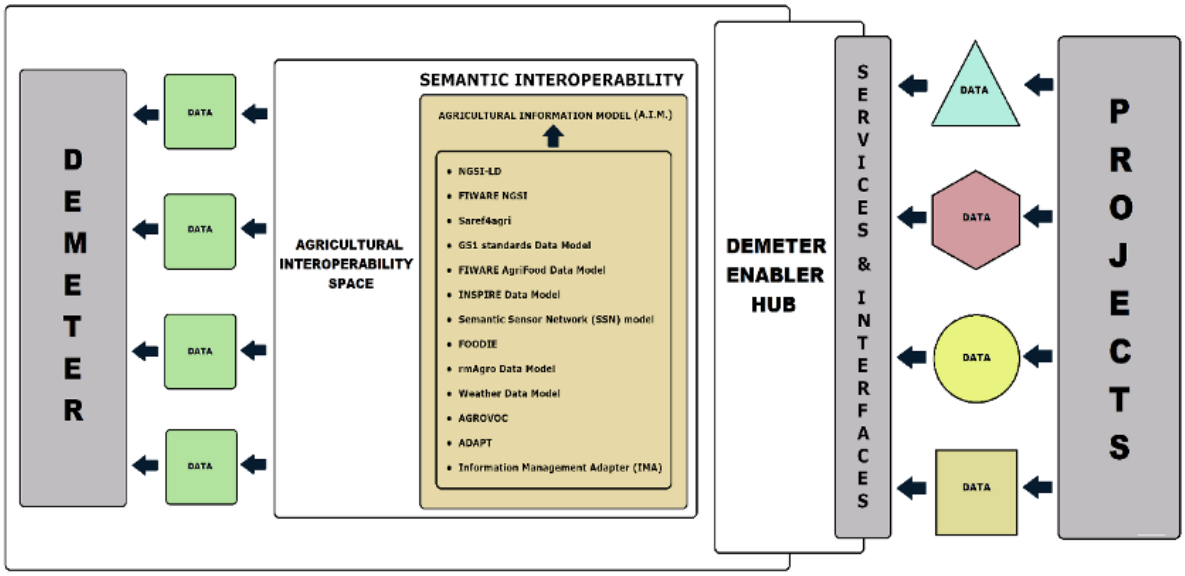


Figure 9: DEMETER Data Standardization Schema

Thanks to the strategy described above, DEMETER's core will receive normalized data which will be used in the next phases of the Information Lifecycle Management in a coherent way, avoiding the problems of technological heterogeneity of the devices available in the agricultural ecosystem.

The next step in the Data Lifecycle Management strategy consists in storing/maintaining the standardized data collected in the previous point. Such data, which is composed of text, images, videos and audios, supposes a huge volume of information and will be processed according to DEMETER's Reference Architecture, which is based on the analysis performed to the following frameworks:

- IoT-A
- AIOTI
- BDVA
- NIST
- RAMI4.0
- IIRA
- DataBio
- IOF2020
- FIWARE
- IDSA
- AfarCloud

According to such analysis, DEMETER's database will be divided in two main sections to perform the data management:

- Data & Knowledge Repository
- User & Resource Registry

The first one is in charge of storing the data generated by all the components of DEMETER's ecosystem and the second one stores the users, personal data and permissions as well as some data entities. The data flow for both of them is described in detail in the Figure 10 that links with the data flow picture from previous D.L.M. phase:

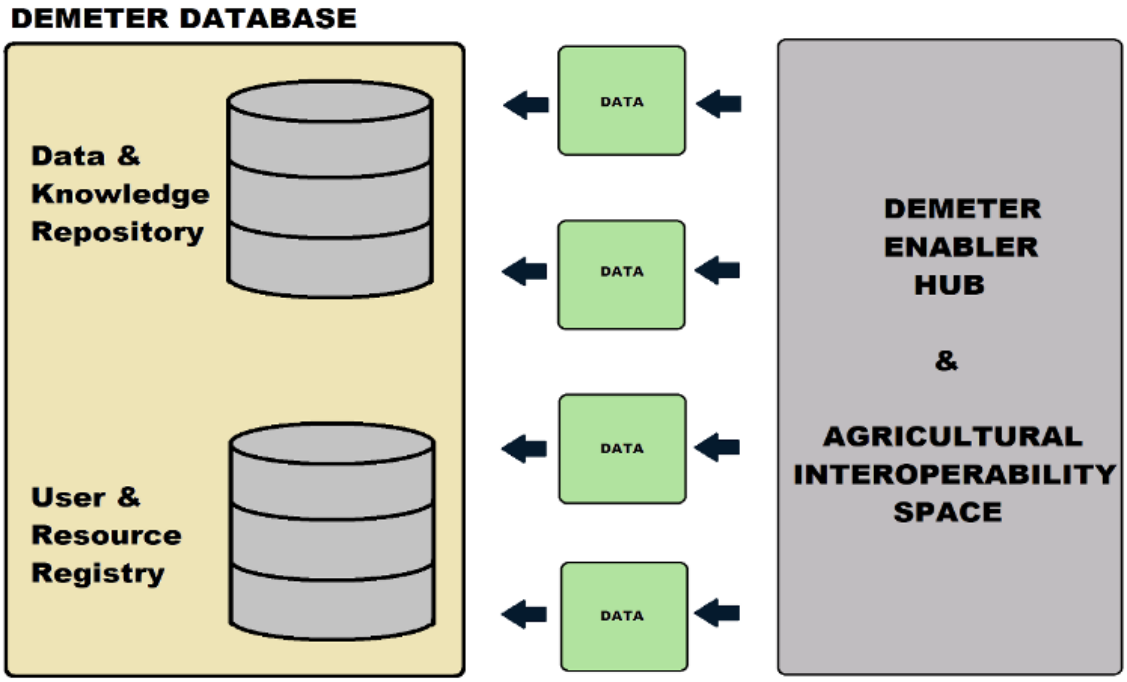


Figure 10: DEMETER Database Schema.

Once the data is stored in the database, it is ready to be used by any internal services from DEMETER Enabler Hub. One of the main consumers of the data is the Dashboard, which is in charge of acting as an interface for the use cases defined. It will be implemented using the DYMER open source visualization tool, through such an interface layer, data will be accessible to all users wishing to send/receive information from the DEMETER database.

Another main actor of the data usage are the services that allow the input from the devices used in the projects. Such data income will be performed in two ways:

- Access through wrappers and interfaces for third party or legacy infrastructure.
- Direct access through D.E.H. for compatible devices.

All the described data usage will be secured using an identity manager enabler based on KeyRock and an access control manager enabler based on Distributed Capability-Based Access Control (DCapBAC). The first one will have a Restful API based on OAuth2 and will manage the users, roles

and permissions. The second one will apply Context-aware Access Control Policies using XACML framework, a capability manager and a PEP-Proxy.

Any user/device wishing to access the Dashboard or use the services will have to login and, once logged and according to their granted permissions, will be able to recover a determined type or amount of information.

In addition to these security measures, most of the data stored in the database, as for example user passwords, will be encrypted using the latest technologies (all security related issues are described in detail in task T2.4 and T3.4).

Figure 11 shows how the data stored by DEMETER is accessed by the users and the devices:

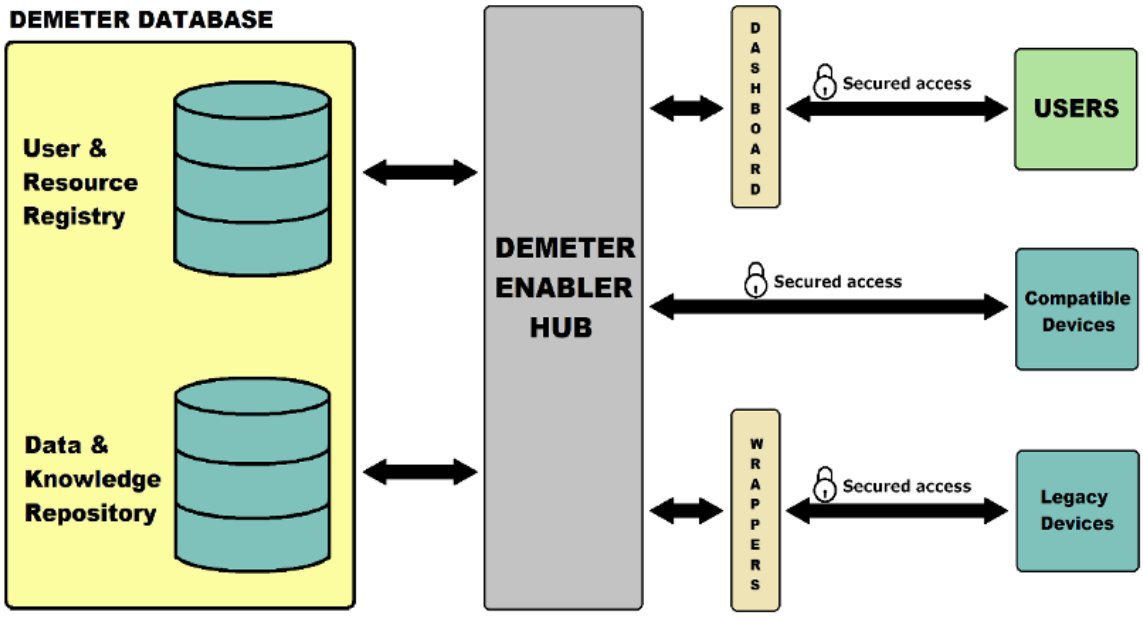


Figure 11: DEMETER Data Access Schema.

If personal data or other sensitive data are to be archived, the project will do so with the right security measures in place. This foresees that the data management system must be able to integrate with the DEMETER components that will take care of this task. Any processing of personal data will be subject to appropriate technical and organizational security measures against unauthorized access and changes, taking into account the nature, scope and context. The enablers who will take care of this task will also implement security measures in data management to control access to information and to manage authorization

The implementation of the data management components can be found at the WP2 Data Management folder in DEMETER GitLab: https://gitlab.com/demeterproject/wp2/datamanagement

8.2 Design/approach (including UML diagrams)

This section provides a high-level design of the entire stack of data management block components (or DEMETER Enablers), outlining its system during development and validation activities within WP5. These results have been guided by previous work and by the results of WP2, in particular the analysis of the requirements and the reference application scenarios. A UML[82] diagram as Figure 12

[82] https://www.uml.org/

is used to identify the main Enablers in the data management block, the relationships between them, and the application flows. Below, a holistic approach of the data management block which shows the relationships between them, the data flow and how these modules will interface with the data providers:

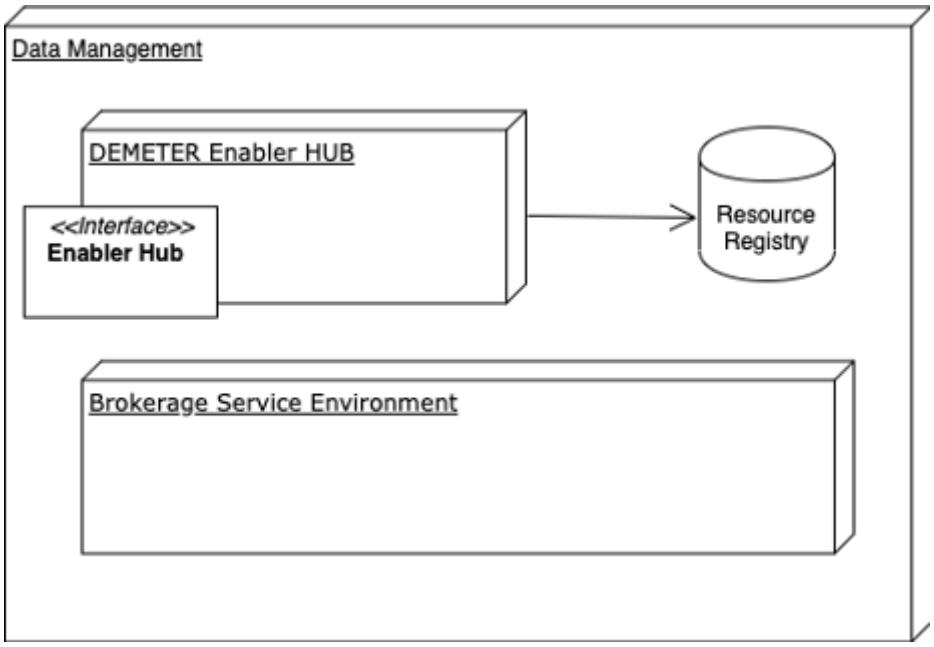


Figure 12: Enablers block for data management in DEMETER Project

The component diagram represents a very high-level view of the blocks or Enablers that will become part of the data management process, allowing their acquisition, transport and finally storage. As can be seen from the figure, the purpose is not so much to represent the intrinsic characteristics (such as modules, technologies and relationships between them) of each Enabler (which will be done in detail in another Project Deliverable or D3.2) but only to show the flow of information from heterogeneous data sources to DEMETER, and the main components involved in this process.

The diagram, show the followed DEMETER Enabler or group of Enablers:

- BSE - Brokerage Service Environment
- DEH - DEH Enabler HUB

As already described in D3.1, each platform, thing, service or application is represented by a DEMETER-Enhanced Entity. These resources will be made available through the DEMETER Enabler Hub; the resource once in the HUB could be consumed by other services and therefore produced. The instances of the DEMETER Enhanced Entities such as resource, service and application will be annotated with metadata describing their characteristics, information relating to their ownership and the restrictions of their locations; this information will be transported from the BSE to the HUB which will store this information in the Resource Registry. The following Figure 13, only by way of example but not limited to, it shows through a sequence diagram how the three components will interact with each other, when it will be necessary to register a resource and its metadata in DEH.

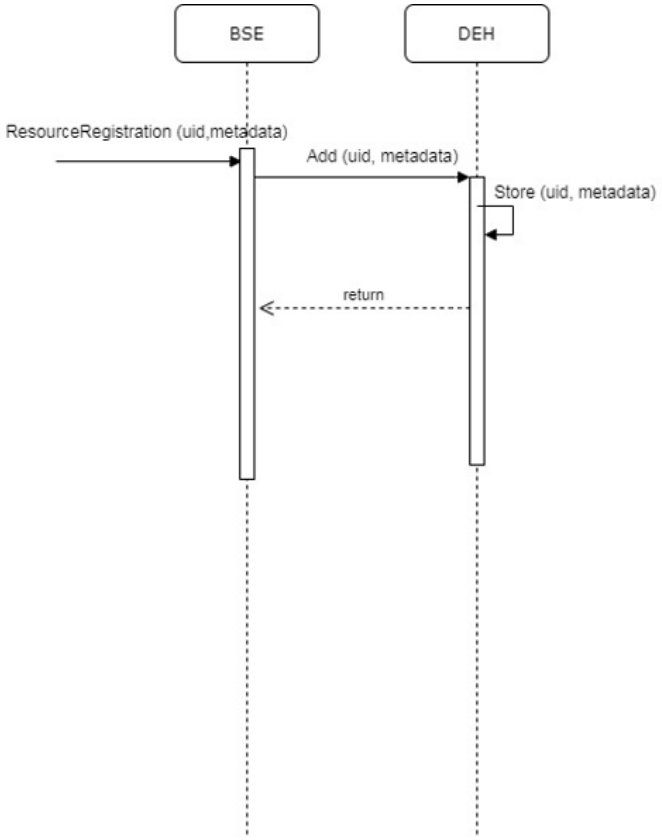


Figure 13: Data management resource registration sequence diagram

DEMETER APIs will offer a whole series of services or enablers such as those that will deal with the transport of information or the interoperability of semantics, security and the client APIs of the HUB which will then allow both data discovery and registration in the DEMETER Hub.

The BSE and DEH Enablers, drawn in the diagram don't show their internal technology characteristics, as they will be detailed defined in D3.2, but they outline the fact that these Enablers will use a SaaS (Software as a Service) approach and then expose interoperability APIs such as RESTful Services, in order to manage the data coming from DEMETER-enhanced Entity.

8.2.1 Multi-tenancy

Since its early years, the SaaS (Software as a Service) and cloud market have evolved from a single tenancy approach where each user had its own database and software instance to a multi-tenancy approach where hardware, software and databases are shared among many users. The benefits of such strategy are evident as shown below:

- **Lower hardware costs**: sharing the same physical infrastructure allows dramatic cost cutting.
- **Decrease software development costs**: as all users work with the same software, no personalized developments must be performed, thus this fastens the development process and lowers costs.
- **Faster software/data updates**: due to the fact that only one software/database needs to be maintained, the updates can be performed sooner, which results in security and reliability gains.

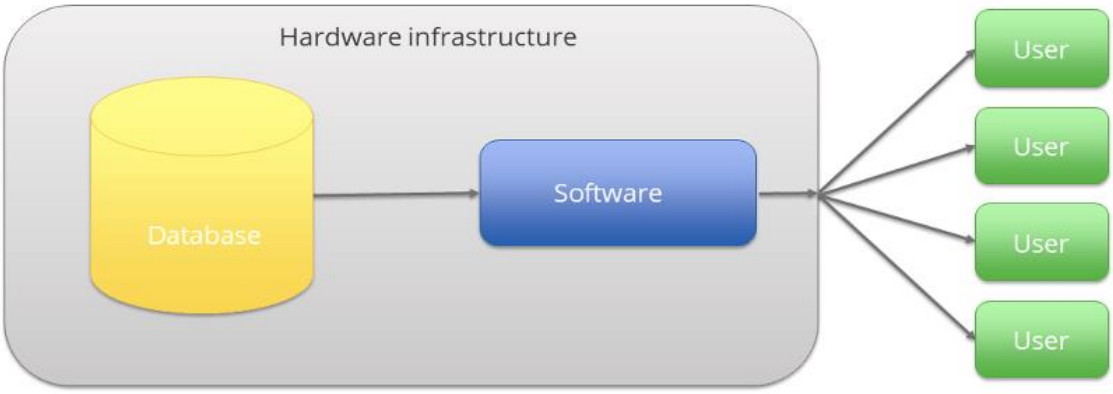


Figure 14: Multi-tenancy schema

DEMETER, as any other modern SaaS software/platform, will implement a multi-tenancy data management approach that will allow sharing the same data between a big number of users. In order to do so, it will be managed by BSE (Brokerage Service Enabler) component, that will expose many DEE (Demeter Enhanced Entity) to connect with.

Each of such Demeter Enhanced Entities will offer some core enablers:

- Communication & networking enabler
- Semantic interoperability enabler
- Security protection enabler
- DEH Client enabler

Any Pilot wishing to connect with DEMETER will be able to do so through the available core enablers from the enhanced entities or directly with the BSE in case of being compatible with it, as shown in Figure 15:

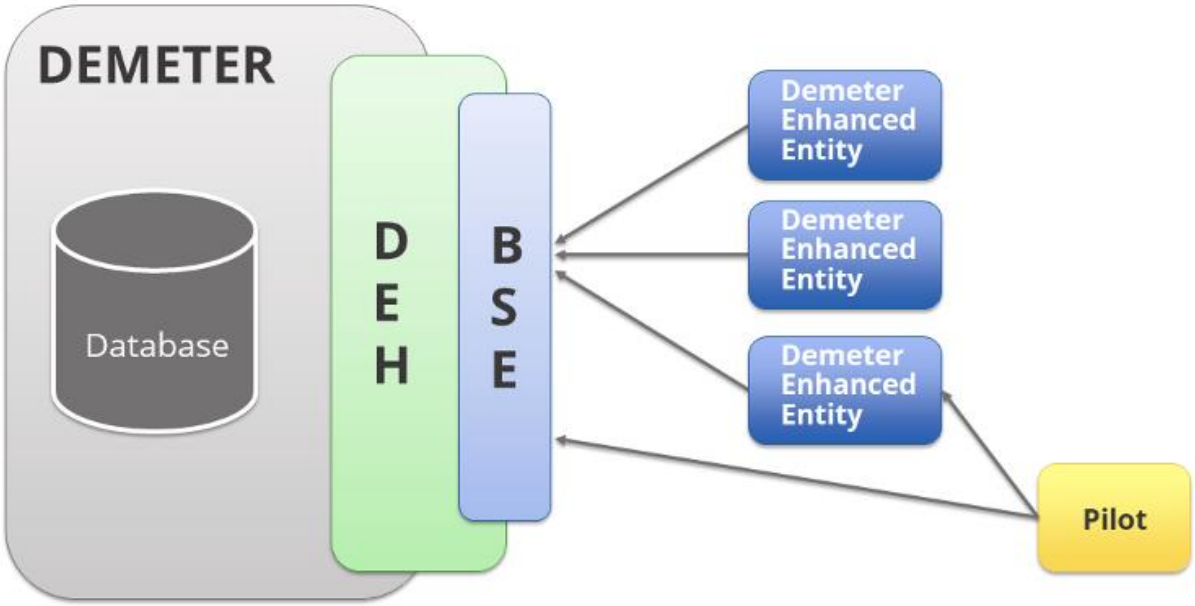


Figure 15: DEMETER Pilots connection to DEH

The DEMETER solution (at least for the execution of the Project) will provide a cloud approach for implementing the DEMETER Enabler HUB. From a technological point of view, the creation of flexible

cloud architectures was enabled by the affirmation of the mechanisms of multitenancy (single instance, multi tenants) which have enabled the provision of scalable services thanks to the dynamic sharing of resources. Multitenancy is an architectural principle whereby a single instance of a software service can be activated without the need to replicate architectures dedicated isolates, thus providing a service to many client organizations (tenants). To achieve scalability in the cloud, other physical instances are typically added. This is known as horizontal scalability. For example, in web applications, to manage more traffic, it is possible to add other VMs of the server and insert them in a balancing service of the load. Each VM runs a separate physical instance of the web app. The client requests can be directed to any instance. Collectively, the system functions as a single logical instance. It is possible to interrupt a VM or add a new one, without this having any impact on users who use the system, but this also if it is data supply services that are connected to a software infrastructure. In this architecture, each physical instance is of the multi-tenant type and capacity can be increased by adding more instances. If an instance is deactivated, it has no impact on tenants.

When designing a multi-tenant SaaS application, you need to carefully choose the tenancy model that best suits the needs of the application. This model determines how each tenant's data is mapped to archiving. The choice of the tenancy model affects the design and application management. In many cases, switching to a different model at a later time would be very costly because the redesign and use of some technologies don't support this model. In general, the tenancy model if chosen immediately at the design stage, doesn't affect the operation of an application, but it is likely to affect other aspects of the overall solution.

The discussion of tenancies focuses on the data level. Considering only the application level which in some respects represents a monolithic entity, if the application is divided into several smaller components, the choice of the tenancy model may change. It is possible to treat some components differently than others in terms of tenancy and the storage technology or platform used.

There are different approaches therefore, or models that can support multi-tenancy, considering the application level only for DEMETER, he could think of using a multi-tenant approach for applications or services, with single tenant databases. In this model, the DEMETER solution represented by Core Enablers should be installed several times in each Pilot, while the DEH could be installed on the cloud in a single, shared server instance. Any instance of the Core Enablers would represent an autonomous instance. Each instance of the app has only one tenant and therefore needs a single centralized database of resources, such as the Resource Registry offered by DEMETER Enabler HUB.

The simplest database policy uses a single database to host data from multi-tenant services. This could then be improved by adding more tenants; in this way the database is increased with more storage and processing resources. This increase may be sufficient, although there is always a scalability limit. However, long before this limit is reached, the database may become complex to manage and this may cause slowness in the operations normally performed on an archive. It will therefore be the responsibility of the project partners to always check the availability of resources and prevent this approach from becoming a bottleneck for all applications that will refer to it for saving data from Pilots.

The logic relating to multitenancy in DEMETER is achieved simply by sharing this criterion of the design with the data management Enablers, but also in the technological choice that is made in implementing each Enabler. Obviously, this choice must be directed towards those technologies that can support multi-tenancy, but which also have the flexibility to change models quickly enough and without additional costs or effort. There are different approaches to implementing an architecture

of this type: however, in DEMETER a possible viable solution could be to have a shared database and shared schema approach for the Demeter Enabler HUB.

8.2.2 Availability, scalability & QoS

The Scalability, High-Availability and QoS or Quality of Service are fundamental requirements in cloud-based system infrastructures. The acquisition and support of provisioning of cloud-based systems (whether public, private or hybrid) requires additional resources, as the number of users increases use of these systems.

In information technology, **High Availability (HA)** refers to a system that is continuously operational for a desirably length of time. Since the computer system consists of many parts in which all parts usually need to be present in order for the whole to be operational, much planning for high availability centers around backup and failover processing and data storage and access[83]. The following procedures may be implemented to achieve HA operations:

- Physical data redundancy
- Backup procedures
- Service Level Agreement

Physical data redundancy requires that each virtual machine where services/applications are deployed has a virtual disk assigned from a mass storage array. Each virtual disk is protected by redundancy provided by the storage array.

This is achieved by adding VMs to a **Monitoring System** like Nagios[84]. Nagios offers monitoring and alerting services for servers, switches, applications, and services, alerting administrators when things go wrong and when the problem has been resolved.

In particular, the Nagios instance can check servers live status via ICMP ping. It can also monitor the ssh service for linux based systems and RDP for windows-based systems. We could add other services, like database or web applications monitoring. As Nagios sends email to administrators as soon as it detects some problem with services, we will be able to address the issues timely, avoiding long downtime of the services provided.

Another fundamental aspect to consider in the implementation of a high-performance system are the **Backup Procedures**. Virtual machines with Linux systems are usually backed up at the files level. The main goal of this backup is to store a copy of all files on another storage system. The copy can be used to restore data in case of accidentally removing or unaware modification. The backup script creates a compressed archive file (with tar and gzip programs) and stores it on a local disk. Each virtual machine has a virtual disk assigned from a mass storage array. Virtual disk is protected by redundancy provided by storage arrays. After completion of this process the archive file should be uploaded to some persistent storage service, e.g., Platon-U4[85] via sftp connection. Backups are normally made once a week, e.g., every Sunday at 04:00:00 (CET).

All databases should be backed up, e.g., every week. Backups should also be stored in some persistent storage service like Platon-U4 storage service. For other backups, by default, the VM administrators would be responsible to take care of the data stored on their machine.

[83] http://searchdatacenter.techtarget.com/definition/high-availability

[84] https://www.nagios.org/

[85] https://storage.pionier.net.pl/

SLA (Service Level Agreement) is a part of a service contract where particular aspects of the service (e.g. scope, quality, responsibilities) are agreed between the service provider and the service user. A common feature of an SLA is a contracted delivery time[86].

The most important objectives of SLA are as follows:

- Creating an environment that ensures effective support of end users
- Describing the responsibilities of all parties of the agreement
- Defining the commencement of the agreement, its initial term and the provision for reviews
- Defining all details of the service to be delivered thus reducing the risk of misunderstanding
- Providing a common understanding of service requirements/capabilities and of the rules involved in the measurement of service levels

The SLA contract that will be signed by all involved parties shall include the following assumptions:

- The agreement will commence on agreed date and will continue until the end of the project;
- The agreement will be reviewed at an agreed date by all involved parties. The review will cover services provided, service levels and procedures. Changes to this agreement must be approved by all parties;
- The SLA summary reports will be delivered on a regular basis;
- The service provider shall not be liable for any loss, which are consequences of natural disasters, as well as network outages that are consequences of problems in another ISP's network. Responsibility is limited to the actual loss. Parties shall not be responsible for the loss of benefits as a result of the incident;
- The service provider reserves the right to immediately disconnect any or all of the devices from the power supply in case of any risk associated with continued operation (especially fire hazard);
- The service provider reserves the right to block network traffic to any or all of the systems in case of a reasonable suspicion of an attack carried out from one of the systems;
- The service provider will accept incident requests 24/7. Standard priority requests will be handled during business hours. The service provider will make every reasonable effort to resolve any "high priority" (e.g. complete inaccessibility of hardware or loss of network connectivity) issues as soon as possible.

Hardware related issues are usually resolved on the best effort basis, however, if the issue resolution involves actions covered by the supplier warranty package, the resolve time is not less than the support request resolve time offered by hardware supplier support package and described in support package warranty terms and conditions.

The provider hosts the aforementioned hardware in its premises, providing also redundant and uninterruptible power supply and network infrastructure. The provider shall not be liable for network and power outages which are consequences of natural disasters, as well as network outages which are consequences of problems in another ISP's network.

The goal of the service is to ensure availability of aforementioned hardware at the agreed level not counting planned maintenance times. The availability metric will be measured by a rolling, e.g., a 6 months period. The administrative tasks and responsibilities of provider's admin group include:

- Supervising and monitoring the running servers,
- Identifying issues and problems related to hardware and network connections

[86] http://en.wikipedia.org/wiki/Service-level_agreement

In case of maintenance hardware and software activities these will be announced to project community:

- Two days in advance if the maintenance action expects the reduction of available resources only
- Seven days in advance if all of the equipment will be inaccessible during the maintenance.

The **System's Scalability** generally represents the ability to increase or decrease scale according to needs and availability. A system that implements this ability is considered scalable. The most common use of this property of a system is very often translated into load scalability, that is, the ability of a system to increase its performance if new resources are provided to it (e.g. greater computing power such as adding die Hardware as additional processors). A system is more scalable if its software architecture and/or hardware architecture is able to manage one or more bottlenecks through the authorization of the overall computing power. This intrinsic characteristic of the system architecture applies in general terms to the other meanings in which the term "scalable" can be understood or referred to.

The Scalability needs for cloud-based systems can be classified into two main categories:

- scalability requirements associated with identifying system usage and bottlenecks so that the application (an estimate) can be adapted to the necessary resources.
- autonomous resource acquisition requirements along with resource acquisition rules so they can be the resources of the cloud themselves to adapt upon request to new scalability needs.

The most common monitoring metrics are those that base their logic on monitoring the CPU of the machines or VMs of the entire hardware industry and software that hosts these systems, but also an active profiling and identification of bottlenecks associated with response time through use of special algorithms. Cloud resources can be acquired from private, public and hybrid clouds depending on data and service constraints. For a more competitive monitoring of the approach to acquiring resources, different schemes can be used to determine whether availability of resources is sufficient or insufficient. Vertical scaling (when only the scalability model is adapted to a layer of a based system cloud) can also be adopted in cloud-based systems using private, hybrid and public IaaS (Infrastructure as a Service) clouds. Therefore, the scalability requirements for cloud-based systems should consider all the factors mentioned above.

Another important factor to consider in the DEMETER cloud infrastructure is represented by **QoS or Quality of Service**. In the networking, QoS indicates the priority with which applications can access network resources. Quality of Service refers to a series of indicators that help establish the overall quality and performance of a network (such as a LAN or the Internet). In particular, this evaluation is made from the "user point of view". For some types of connections, QoS is a determining factor. Some web applications particularly expensive in terms of bandwidth require that the cloud operator guarantees precise quality standards for the execution, without interruptions or any kind of related problems.

QoS determines the different priority levels with which different applications, users or data flows will be able to access the available network resources: for some programs or protocols, for example, certain levels of bit rate (bit transmission frequency), delay (delay in bit transmission), jitter and probability of errors in data transmission. The ability to set different priorities is crucial especially if the network resources are scarce or in any case insufficient to guarantee the transmission of all the data flows that need it; in cases like these, to respect the minimum limits of QoS, the applications

that needing more bandwidth, will necessarily have a preferential channel than to other applications. A network or in any case a data transport protocol that supports the QoS may enter into a specific contract traffic (traffic contract) with software and applications that require guarantees on the bit rate, delay and jitter in order to function properly.

Some elements more than others are used to define QoS:

- ***Out-of-order delivery:*** In packet-switched networks, packets may not arrive at their destination in the same order with which they left. This happens because the data can follow different paths and take different times to "complete the path". So that the destination node is able to correctly interpret the received data, the protocol used in the transmission must take care also of the "reconstruction" of the original message, causing a general delay in communication between the two nodes. From a quantitative point of view this indicator consists of an assessment of the probability that a data packet will arrive out of order.
- ***Delay:*** The transit of a packet within a network usually takes a few milliseconds (abbreviated in ms, equivalent to a thousandth of a second). A problem is considered the case in which the delay in communication exceeds ten ms. For this purpose they are generally considered two indicators: the "average delay", which indicates the average time for data packets to cross the network, and the "percentile delay", which indicates the percentage of data packets that reach the destination within a certain maximum time (which, in general, is decided based on the type of application involved).
- ***Packet loss:*** It is not always said that all packages find their way. For several drawbacks, in fact, it can happen that one or more packets are lost, causing a general delay in communication. In these cases, the communication protocol asks for the sending of the packets not received at the sending node, so that we can complete the message and interpret it correctly. Again, the factor that defines the QoS indicates the maximum percentage of packets lost out of the total packets sent.
- ***Transmission error:*** Due to signal interference or problems in the communication medium, packets can arrive at their destination damaged. The knot recipient, by means of the communication protocol used, must request that they be sent again or, as happens more and more often, try to reconstruct the message itself based on the packets already received, interpolating the missing information. Again, the value that defines this factor represents the maximum percentage of damaged packets compared to the total sent.
- ***Throughput:*** QoS also depends on the bandwidth actually available to the user. The throughput represents precisely this factor: the amount of bandwidth that can actually be used by the user for data transmission. The maximum available bandwidth is established upstream by the ISP (Internet service provider) but it may vary, moment by moment, according to the network resources available to general level (for example, it can drop even a lot in case of high network congestion), consequently the throughput can reach saturation completely the transmission capacity available at that moment. Therefore, the higher the throughput guaranteed by the ISP, the more the. QoS associated with it will be high.

Considering DEMETER's Frontend applications like DEH Dashboards, it can be safely said that they do not have particular needs and can also work on low networks performance. When, on the other hand, we refer to the internal or backend services that will allow communication between DEMETER Pilots and Enablers, we could say that these modules are more rigorous in operational terms and therefore would require more guarantees on the available bandwidth, on the average delay (or percentage) and on the percentage of error admitted by the network: therefore a QoS that guarantees their correct functioning. Finally, it is fair to say that there are no unique tools that we

could suggest here, capable of managing QoS or scalability in DEMETER, but useful monitoring criteria to be taken into consideration; these monitoring criteria could help regulate the flow of data of applications or services (from the cloud to a client and vice versa) in order to guarantee an efficient service, without network interruptions or errors of any kind.

However, there are real monitoring tools that have implemented these criteria. There are consolidated tools like Apache JMeter[87], which could guarantee the monitoring of DEMETER data management components in order to ensure their functioning in the operating environment of the project execution. These tools are designed for application scalability, basing their logic on performance testing. However, there are tools that can monitor and manage software but also network resources such as Hyperic HQ [88], an application monitoring and performance management tool for virtual, physical, and cloud infrastructures.

In addition, simply measuring the QoS of cloud computing is not particularly demanding, since for example Microsoft Azure or AWS offer paid services that allow monitoring appropriately for the desired degree/type of QoS/IT services.

8.3 Implementation (including interfaces)

This section contains a high-level detail of the components or Enablers proposed for blocking data management: these represent the core of the solution, with different but contiguous purposes to make the solution for enabling data management solid and efficient inside the DEMETER Project. The effort here is to highlight the potential of each block, the main features such as the data communication interfaces, which each component has and which allow it to cooperate with the other components within the data management block, but more in general with all the other DEMETER enablers.

Each component, or group of enablers, is/are described in a dedicated section in the Data Management section of this document. Each section contains:

- A general description of the component
- Main Enabler interface

A general overview of the interface exposed by each component, are depicted in the Figure 16:

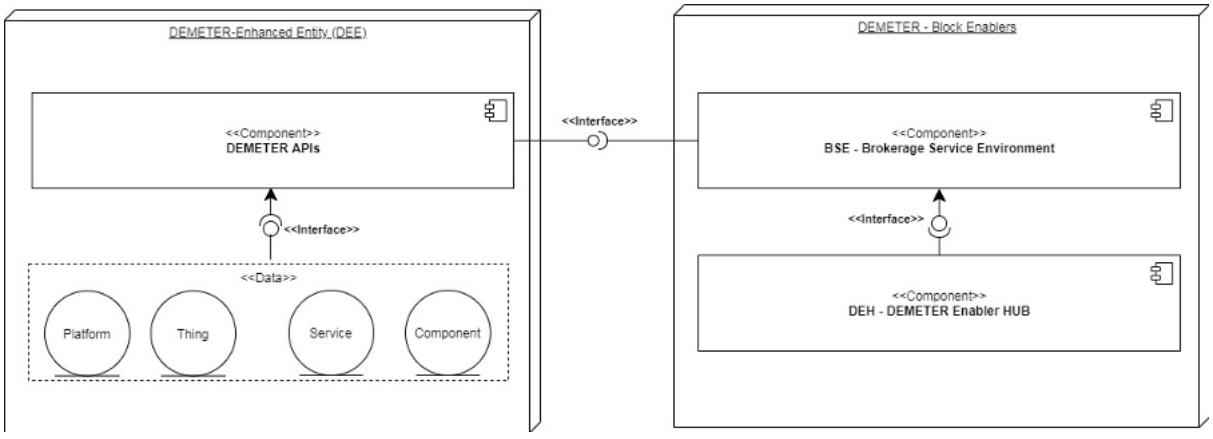


Figure 16: Main interfaces of Enablers block for DEMETER data management

[87] https://jmeter.apache.org/

[88] https://sourceforge.net/projects/hyperic-hq/

Each interface of the block represents a connector to the DEMETER world and enables a specific service or API. These services will cooperate together in data management. The resources, coming from data providers such as Pilots, through Case-specific API will provide data to DEMETER through the DEMETER API block consisting of a DEMETER Core Enablers which will enable a whole series of mechanisms:

- Communication and networking
- Semantic Interoperability
- Security
- DEH client

The latter component of the Enablers block will take care of the saving of the Resources in the AIM format in a Resource Registry Store (it should also be remembered here that all these components or Enabler will be detailed in their entirety within the D3.2), through the BSE Brokerage service Environment component. The Core Enablers combined will form DEMETER-enhanced Entities as described in the Reference Architecture document.

8.3.1 Brokerage Service Environment (BSE)

One of the core components of DEMETER platform is the Brokerage Service Environment (BSE). BSE plays an integral role for the interconnection of the providers and consumers of the various DEMETER compliant resources. In deliverable D3.1 the deployment diagram was presented in Figure 17 where the AIS module was depicted; BSE is part of the AIS.

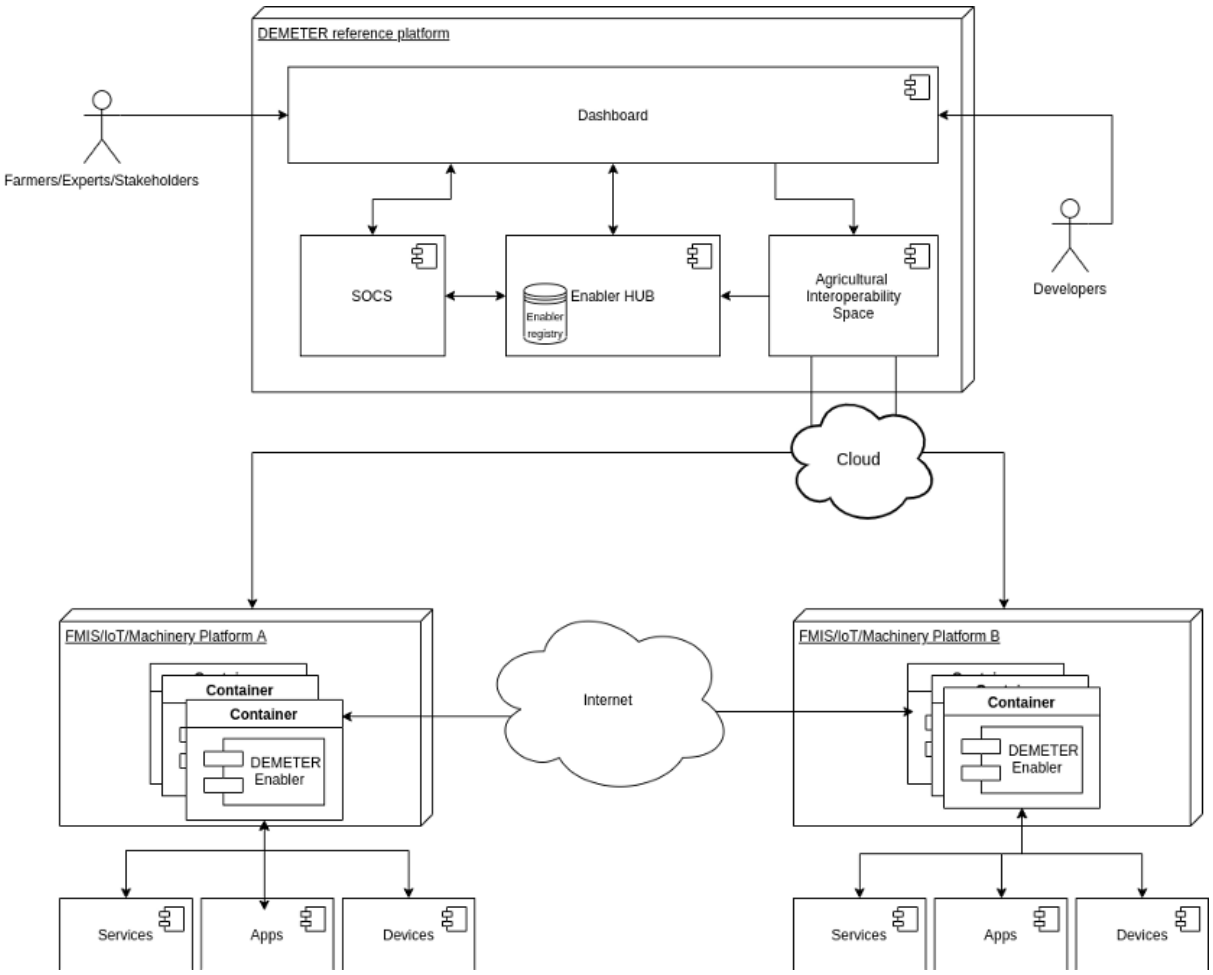


Figure 17: Brokerage Service Environment

BSE provides three basic services, registration, discovery, provisioning through APIs (e.g., REST, pub/sub). As depicted in the diagram above, data from various resources (services, platforms, devices) flow via DEMETER Enablers to DEMETER's platform components. Data are structured according to their owner needs and related needs at the resources level. However, DEMETER focuses on interoperability of resources; hence, data needs to be translated into a DEMETER-compliant data model if they are to be provided to consumers. This data model is derived from the work performed and presented in the context of D2.1 where AIM (Agriculture Information Model) was defined. Therefore, data is translated into a DEMETER-compliant form after going through the translation process within a DEMETER Enabler. Enablers might offer different interfaces to the resources, e.g., REST APIs, MQTT endpoints. In combination with the rest of DEMETER Enablers that are deployed within the same DEMETER Enhanced Entity, data flows to BSE in order to consume the services offered:

1. To get registered and advertise their features using metadata
2. To get available for discovery from interested consumers
3. To get provisioned to consumers

On the other data flow direction, data might need to be translated from DEMETER's data model to a format that fits the consumer's needs. In this case, another DEMETER Enabler has the responsibility of providing this translation.

The result of the above procedure is that data are translated from/to DEMETER data model at the Enabler level, thus achieving interoperability between heterogeneous data providers and consumers.

BSE as well as the Core Enablers, will be described in detail within D3.2 (DEMETER Technology Integration Tools - Release 1) which is due M10.

8.3.2 DEMETER Enabler HUB (DEH)

DEMETER Enabler HUB as one of the crucial components of DEMETER project involves communication between various DEMETER components. Core services included in DEH are Compatibility Checker, Resource Registry Management, Discovery Management. Demeter Dashboard will be provided as an external component outside of Core APIs and it's an existing solution - DYMER, which is also part of DEH Enabler. In order to provide a secure connection, DEH client enabler will communicate with Security Protection Enabler, specifically with Identity Manager, which relies on FIWARE IDM[89] and Resource Access Control components. All listed components inside Core DEH API will be available as SaaS (Software as a Service). Communication between core services will be provided through APIs which will be producing and consuming AIM model information (based on JSON-LD interoperability exchange format), since the focus is to rely on the AIM model as much as possible. DEH Client Enabler will expose respected functionalities implementing a set of API Clients capable of communicating a DEMETER-Enhanced Entities with the DEH Enabler HUB. Data entities from any Platform, Thing, Application, Service will be managed through these APIs, but for the sole purpose of discovery and management of the registry by DEH.

Data gathered from DEMETER Pilots will be published at Brokerage Service Environment, and it will be exposed with API, which will communicate with DEH Enabler Client. First data should be checked with Compatibility Checker Component: If data satisfies all necessary requirements, it will be passed

89 https://fiware-idm.readthedocs.io/

to Resource Registry Management and stored into DEMETER Resource Registry. Resource Registry Management module is in charge of all operations related to storing and managing DEMETER Entities. At the end. a display module or DYMER, will be able to show these resources in resource register mode to the end users of DEMETER, who intend to view them. This module will be able to use the same module like Resource Registry Management in order to fetch a resource from Demeter Resource Registry.

Due to an overlap with the D3.2 deliverable, the internal modules of the DEH, the technologies used to implement them and all the technical details on the services will be fully reported in that document.

8.4 Data management components in DEMETER-enhanced entities

Various types of sensor data provide valuable information to farmers. The development of SensLog [90]within the Demeter project will lead to the extension of SensLog by the Demeter API, which will allow individual instances of SensLog to be registered as Demeter Enhanced Entity (DEE) and Deploy SensLog instances discoverable and accessible through Demeter BSE. SensLog will play the role of the main tool for acquiring, storing and publishing sensor data within pilot 2.3 Data Brokerage Service and Decision Support System for Farm Management but if needed, it will also be usable in other pilots. Another possible role that SensLog can play in the DEMETER project is the use within the data preparation and integration pipelines described in Section 9 as SensLog has been tested in the past in pipelines, which are expected to be used as a base of Demeter Data preparation and Integration components.

SensLog is an open sensor data management solution for receive, store, management, analysis and publication of sensor data. SensLog is suitable for web as well as cloud environments. This solution is suitable for static in-situ sensors, sensors on mobile carriers and Volunteered geographic information (VGI) gathered by smart devices. SensLog stores data in relational data model and the used RDBMS is PostgreSQL[91] with PostGIS[92] spatial extension. SensLog data model is inspired by the ISO Observations Measurements standard, but contains additional extensions mainly for user management and hierarchy of sensor network. SensLog is a modular solution where every module is designed as microservice, thus scalability of the solution is ensured. Structure of the SensLog modules is shown in the Figure below.

90 http://www.senslog.org/

91 https://www.postgresql.org/

92 https://postgis.net/

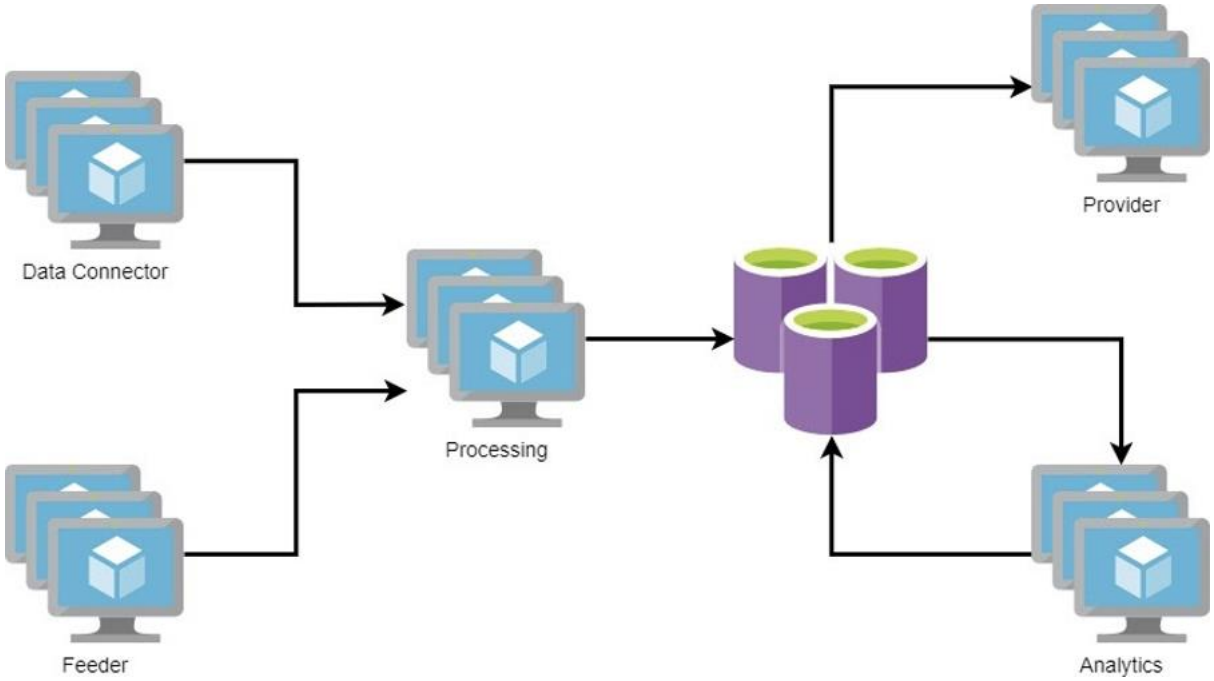


Figure 18: Basic structure of SensLog modules

SensLog provides REST API with JSON data encoding in proprietary format. Interoperability of the solution is improved by the system of Connectors. Connector is a component that translates REST API from different data publishers to SensLog REST API. SensLog is available under the 3-Clause BSD License. SensLog provides REST API interface for the complete data flow receiving - publishing. It uses HTTP protocol and JSON data encoding for most of the services.

- SensLog receiver module - REST API for data receiving, HTTP GET or POST methods, JSON encoding
- SensLog publisher module - REST API for data publishing, HTTP GET method, JSON encoding
- SensLog SOS module - core methods of OGC SOS 1.0.0 specification[93] for data publication, describing sensors, HTTP GET method, XML data encoding
- SensLog VGI module - REST API for receiving and publishing VGI data, HTTP GET or POST methods, mainly JSON data encoding, RDF encoding only for data publication

SensLog can be deployed in a web application container (Apache Tomcat[94]) or using Docker[95] containers.

Sensor data is an important data source for many domains and gathering data from different types of sensors is important especially in agri-food projects. Different observations and measurements can be produced by static in situ sensors (e.g. meteorological station), by sensors on mobile carriers (e.g. telemetry unit) or by smart devices (e.g. VGI - Volunteered Geographical Information). Adding locations and positioning to the sensor data provides additional dimension and valuable information, but it demands additional requirements on the sensor data management. As many sensor data producers provide different types of sensor data in many different formats and protocols, integration and harmonization of heterogeneous sensor data is a very challenging task. Reducing the spent effort during integration of data can be supported from the beginning by data producers by

93 https://www.ogc.org/standards/sos

94 http://tomcat.apache.org/

95 https://www.docker.com/

utilization of common standards and specifications for sensor data publication - like OGC Sensor Observation Service, OGC SensorThingsAPI[96] , OMA NGSI-9/-10[97], OPC UA[98] etc. With the growth of the IoT sector the availability of sensor data and importance of gathering and processing such types of data is growing equally.

The SensLog solution will be used for both tasks of the pipeline as the integration component for data collection and publication by defined endpoint and REST API. As the SensLog data model is based on ISO Observations & Measurements specification, it is suitable for both types of sensor data - static and mobile. It uses spatial extension of the RDBMS to store locations of observations and positions of sensors.

For the first task, collecting and storing observations produced by different static sensors, the design of the pipeline is shown in the following figure. The pipeline is designed to receive data from different sources - static sensor nodes, smart devices, sensor data providers. Data can be received directly by services of the REST API. Data can be received by modern low-powered transfer networks (LoRaWAN[99], Sigfox[100] , NB IoT[101]) using the SensLog Feeder component. Or data can be pulled from different data storages of various data providers by the SensLog Connector by implementing specific provider's API. Different SensLog components are receiving data and pushing them to the main storage - SensLog model. On the opposite side of the pipeline, stored data are provided to different components for further processing by the SensLog Provider component by different services and formats over REST API.

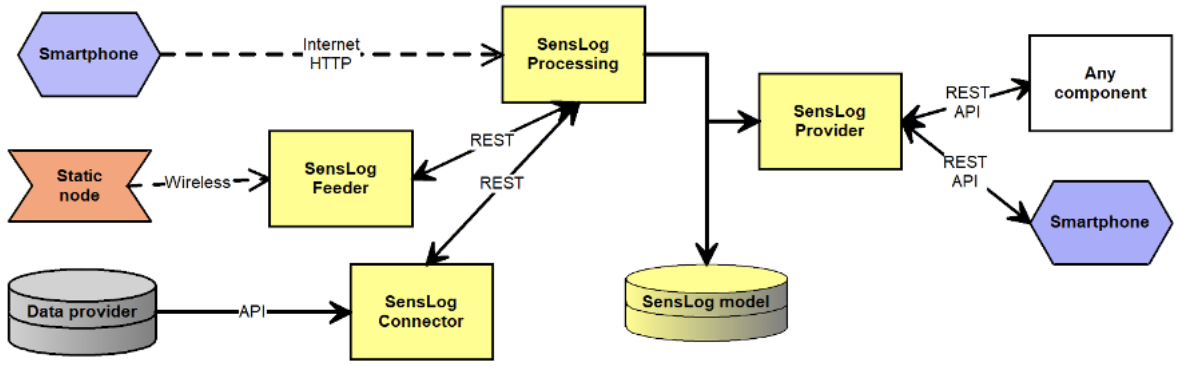


Figure 19: Sensor data pipeline for static sensor data

The goal of this first task of the pipeline is to integrate different static data to common data storage and provide such data over the common REST API as one endpoint.

The second task of the pipeline is collecting and integrating telemetry data from agricultural machinery. Telemetry data can be collected directly from active machinery over some transfer technology directly to the SensLog processing component or over the SensLog Feeder. A smartphone can be used as a telemetry unit on some level of simplification for some basic operations. On the other hand, a very frequent case is accessibility of data not directly on a telemetry unit but only over cloud services of machinery manufacturers. In these cases, utilization of the SensLog Connector

96 https://www.ogc.org/standards/sensorthings

97 http://www.openmobilealliance.org/release/NGSI/V1_0-20120529-A/OMA-TS-NGSI_Context_Management-V1_0-20120529-A.pdf

98 https://opcfoundation.org/about/opc-technologies/opc-ua/

99 https://lora-alliance.org/

100 https://www.sigfox.com/en

101 https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/

component will be necessary to pull data from these manufacturers' services to the SensLog model. Complexity and variability of telemetry data from different manufacturers brings many challenges for integration tasks. A selection of used formats and data specifications from cooperating manufacturers should be selected in next steps of the development.

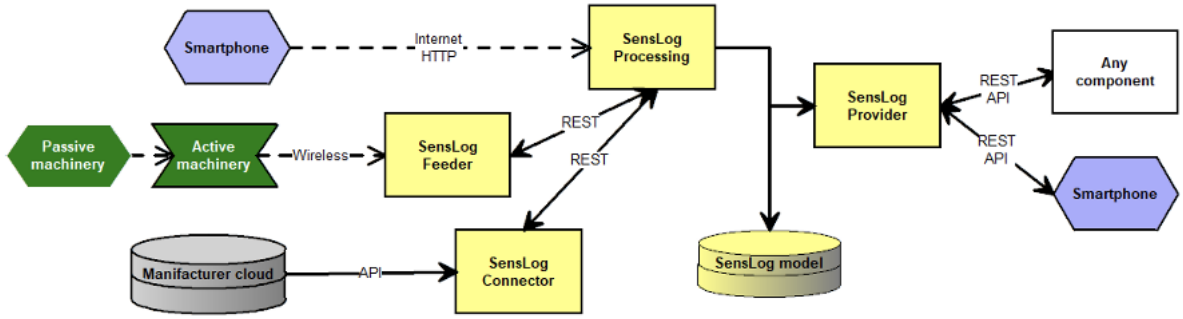


Figure 20: Sensor data pipeline for telemetry sensor data

The goal of this second task of the pipeline is to integrate different telemetry data from different machinery manufacturers to common data storage, run basic analyses and provide such data and analyses results over common REST API as one endpoint.

9 Data preparation & integration components

9.1 Overview

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It often involves reformatting data and standardizing data formats, making corrections to data, removing outliers and combining data sets to enrich data.

Although this is often a lengthy process, it is critical as it allows fixing errors, the data used will be of high quality and will allow making better decisions.

The steps to be followed in data preparation are:

1. Data collection: find and retrieve the data
2. Assess the data: understand the data to know what needs to be done in the following steps
3. Cleanse and validate the data: remove duplicate or irrelevant data, fix structural errors, filter unwanted outliers, handle missing data and finally validate the data by testing for errors.
4. Transform and enrich data: reformat the data to standardized data formats and enrich it, i.e. enhance existing information by supplementing missing or incomplete data with relevant context usually using external data sources.
5. Store data: once prepared, the data can be stored or forwarded to another application or service.

Once the data has been prepared, data integration is the process used to combine data from disparate sources into meaningful and valuable information. It allows achieving a unified view of the data. The most commonly used data integration approaches are[102][103]: data consolidation, data propagation, data virtualization, data federation and data warehousing.

[102] https://www.kdnuggets.com/2020/03/beginner-guide-data-integration-approaches-business-intelligence.html

[103] https://www.educba.com/what-is-data-integration/

- *Data Consolidation* gets data from several individual systems, creating a unified version of the combined data in one repository. The main objective of consolidation is to reduce the number of data storage systems. The ETL (Extract, Transforms and Load) process fetches data from source systems, transforms it into a comprehensible format, and then sends it to the destination system, which could be a database or a data warehouse.
- *Data Propagation* uses the application to duplicate the data from one location to another. It can be made possible in a dual way between source and client.
- *Data Virtualization* creates an abstracted layer to present an almost real-time, integrated view of data from diverse source systems. This approach enables you to view data in a unified place; however, the data is not stored at that site.
- *Data Federation* is a type of data virtualization as it uses a virtual data store and constructs a shared data model for various data from multiple sources. It brings data together from various sources and makes it available from a single point of access, as a single view. Unlike data virtualization, data federation enforces strict data models.
- *Data Warehousing*. The data consolidated via ETL or ELT approach is loaded into a centralized repository called a data warehouse. Data warehousing offers a comprehensive, integrated view of all data assets, with relevant data bundled together. As data warehousing integrates data in a single place, it becomes easier to identify data patterns and make plans accordingly. Warehousing is included as the last step because of its large repositories of data.

9.2 DEMETER approach for data preparation and integration

DEMETER will base its data integration approach on Linked Data. Linked Data is increasingly becoming one of the most popular methods for publishing data on the Web due to the benefits it can provide (e.g., improved data accessibility, support for data integration and interoperability, knowledge discovery and linking). In particular in the agri-food sector, many different projects (e.g., FOODIE, DATABIO, CYBELE, SIEUSOIL) have used or are using extensively Linked Data as a federated layer to support large scale harmonization and integration of a large variety of data collected from various heterogeneous sources in order to provide an integrated view on them. The triple store populated with Linked Data during the course of these projects and few other related projects resulted in the creation of a repository of over 1 billion triples, being one of the largest semantic repositories related to agriculture, as recognized by the EC innovation radar naming it the “Arable Farming Data Integrator for Smart Farming”. Additionally, these have also deployed different endpoints providing access to the dynamic data sources in their native format as Linked Data by providing a virtual semantic layer on top of them.

This action has been realized through the implementation of instantiations of a ‘Generic Pipeline for the Publication and Integration of Linked Data”, which have been applied in different use cases related to the bioeconomy sectors. The main goal of these pipeline instances is to define and deploy (semi-) automatic processes to carry out the necessary steps to transform and publish different input datasets for various heterogeneous sources as Linked Data, i.e. automatic processes for data preparation and integration using Linked Data. Hence, they connect different data processing components to carry out the transformation of data into RDF [1] format or the translation of queries to/from SPARQL [2] and the native data access interface, plus their linking, and including also the mapping specifications to process the input datasets. Each pipeline instance used are configured to support specific input dataset types (e.g. same format, model and delivery form etc.), and they are created with the following common goals:

- Capability of a pipeline to be directly re-executable and re-applicable (e.g. extended/updated datasets)
- Easy reusability of a pipeline
- Easy adaptation of a pipeline for new input datasets
- Automatic execution of a pipeline as far as possible, though the final target is to create fully automated processes
- Pipelines should support both (mostly) static data and dynamic data (e.g. sensor data)

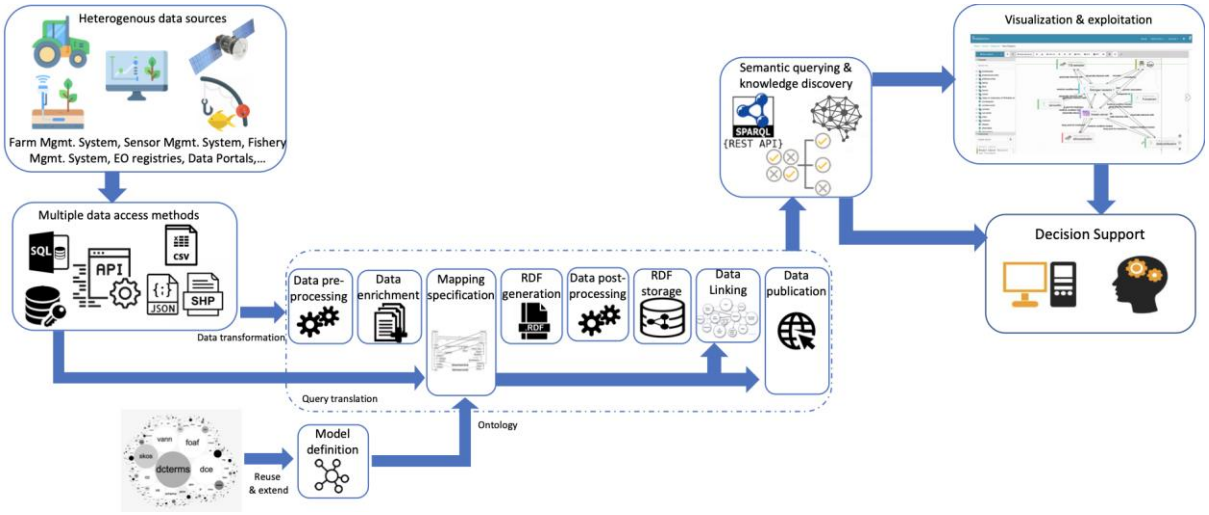


Figure 21: Generic flow for Linked Data Integration and Publication pipeline

A high-level view of the end-to-end flow of the generic pipeline for data preparation and integration using Linked Data, is depicted in Figure 21. Following the best practices for linked data publication [3] [4], these pipelines i) take as input selected datasets that are collected from heterogeneous sources (shapefiles, GeoJSON, CSV, relational databases, RESTful APIs), ii) curate and/or pre-process the datasets when needed, iii) select and/or create/extend the vocabularies (e.g., ontologies) for the representation of data in semantic format, iv) process and transform the datasets into RDF triples according to underlying ontologies, v) perform any necessary post-processing operations on the RDF data, vi) identify links with other datasets, and vii) publish the generated datasets as Linked Data and applying required access control mechanisms. The transformation process depends on different aspects of the data like format of the available input data, the purpose (target use case) of the transformation and the volatility of the data (how dynamic is the data). For the purpose of the specific pipeline tasks various components were used to reach the final goal of transformation of Linked Data. The list of relevant components identified and used in each pipeline instance will be discussed in the later subsections of this deliverable. Another aspect of choosing the most suitable tools for transformation of the source data depends on the targeted usage of the transformed Linked Data and the goal for accessing the data integrated with other datasets also influences the preferred tools to be used. Finally, based on how often the data is changing (i.e. rate of change) the transformation methods and the related tools are to be further determined. Based on the above-mentioned characteristics i.e. mode/format of input data sources there are broadly two main approaches for making the transformation for a dataset:

- Data upgrade or semantic lifting, which consists of generating RDF data from the source dataset according to mapping descriptions and then storing it in semantic triple store (e.g., Virtuoso).

- On-the-fly query transformation, which allows evaluating SPARQL [2] queries over a virtual RDF dataset, by re-writing those queries into source query language according to the mapping descriptions. In this scenario, data physically stays at their source and a new layer is provided to enable access to it over the virtual RDF dataset. This applies mainly to highly dynamic relational datasets (e.g. sensor data) or RESTful APIs.

The implementation of the data preparation & integration components can be found at the WP2 Data Preparation and Data Integration folders, respectively in DEMETER GitLab: https://gitlab.com/demeterproject/wp2/datapreparation & https://gitlab.com/demeterproject/wp2/dataintegration

9.3 Design of data preparation & integration pipelines

This section will describe the top-level generic pipelines and the components used for this purpose. Figure 22 provides i) a simplified top-level representation of the Linked Data Integration and Publication pipeline aligned with the top-level generic pipeline, ii) the pipeline view with specific components of the generic pipeline, respectively.

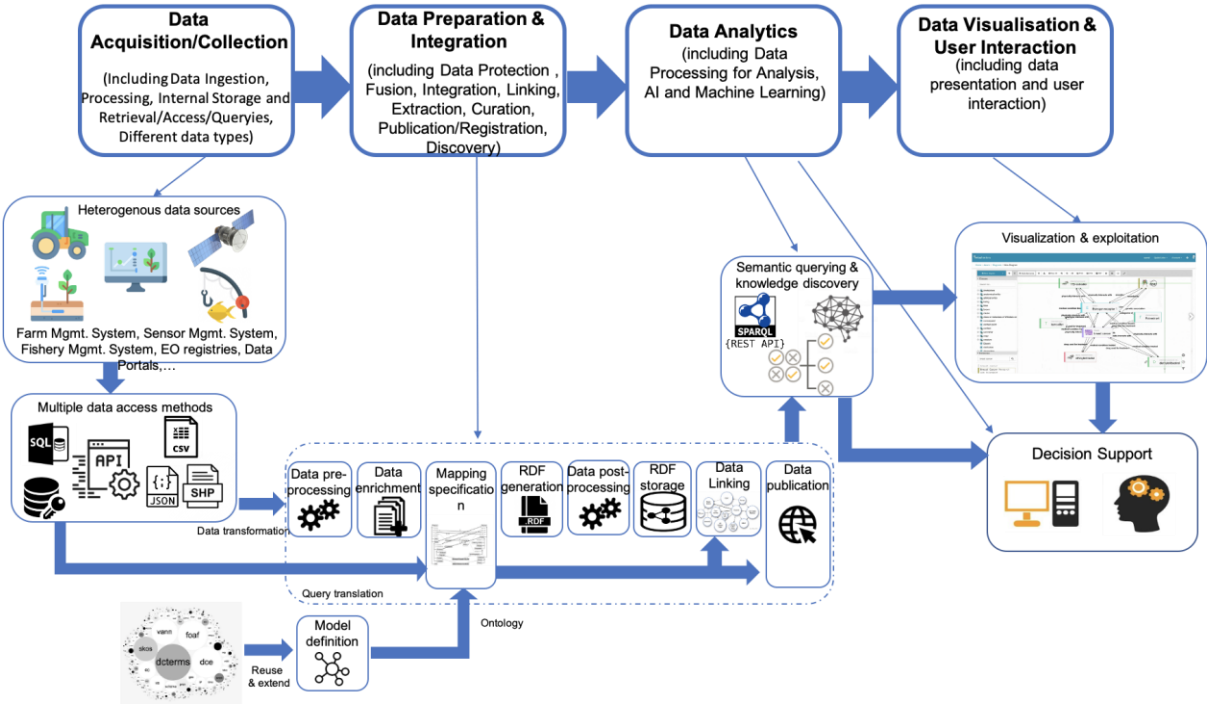


Figure 22: Generic flow for Linked Data Integration and Publication pipeline aligned with top-level generic pipeline

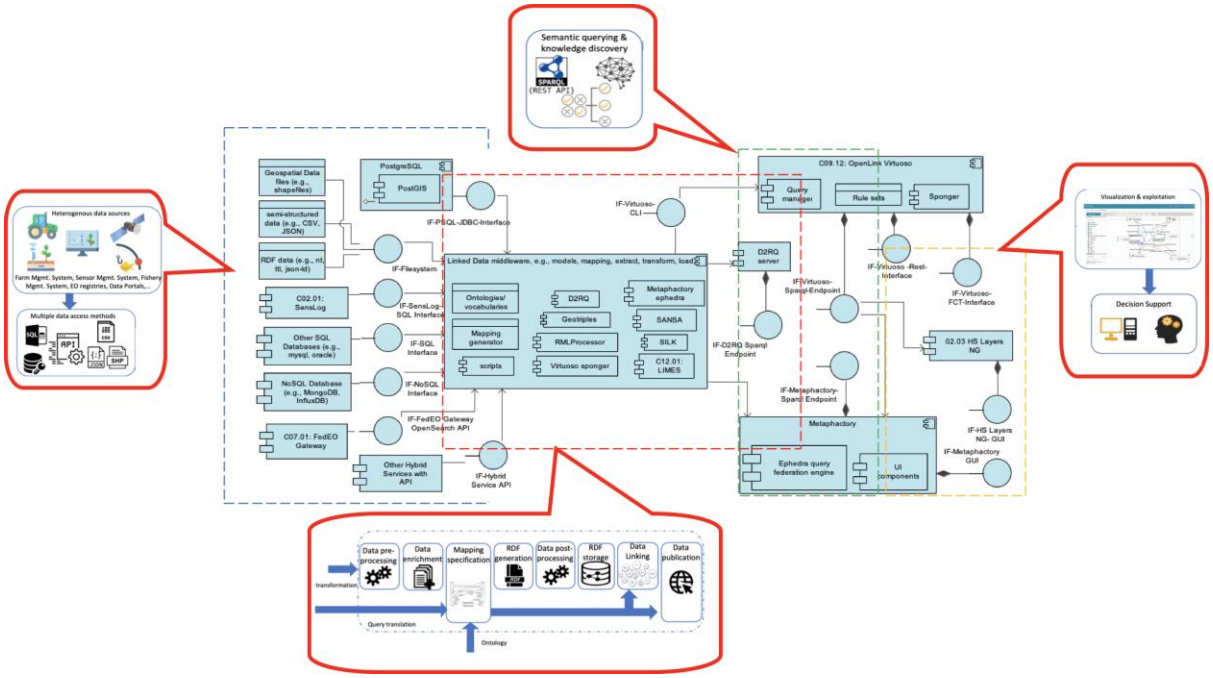


Figure 23: Generic Linked Data publication pipeline component diagram

9.3.1 General workflow and Pipeline instantiations

This section describes the various steps involved in the process of data preparation related to the transformation of data from heterogeneous sources into RDF format and publication of those datasets as Linked Data. Hence, the initial setup includes many different tools and data processing components, including models, mappings, and a large number of linked datasets that can be reused, and extended over a period of few related projects. The approach for data preparation is as follows:

- **Data collection/acquisition:** This is the primary step including data access, ingestion and collection of data from various heterogeneous sources (e.g. Farm Management Systems, Sensor management Systems, Fishery Management Systems, Government Data portals, Earth Observation registries etc.) and hence in various formats (shapefiles, relational databases, CSV, JSON etc.). The collection of data can be carried out using multiple data access methods like retrieval or access or querying from relational databases.
- **Data Preparation:** This is the most important part of the whole process and involves the following steps for various data sources:
    - *Data pre-processing*: This is the step in which the raw data from heterogenous steps are mainly cleaned and prepared accordingly for the next steps in the process, with the help of various customized shell scripts and/or scripts written out of various programming languages like python.
    - *Mapping Specification:* In every transformation process a mapping specification has to be defined to specify the rules to map the source elements (e.g., tables columns, JSON elements, CSV columns, etc.) into target elements (e.g., ontology terms). Generally, this specification is an RDF document itself written in RML[104]/R2RML[105] (and extensions) languages and/or non-standard extensions of SPARQL, e.g., in the case of the Tarql CSV to RDF transformation tool. In case of translation of hybrid

[104] http://rml.io/spec.html
[105] https://www.w3.org/TR/r2rml/

services semantic models are used to represent data resources from the API and thereafter implementing a wrapper (e.g. mapping or describing the REST Service Signature on Ephedra platform) around API to transform SPARQL request to API call and generate RDF data. In case of dynamic data translation mapping specification is done by using various tools using RML/R2RML mapping and well-known ontologies. Mapping specification is carried out by various applications like Geotriples (for shapefiles, CSV, JSON), D2RQ (for translation of dynamic data and/or data from relational databases). More details about these components are provided in the section below.

- *RDF generation* is also a part of this step which involves various techniques for various data formats. In this step RDF triples are generated either by generation of RDF dumps, through on the fly transformation of data or by translating RDF triples by using API wrappers exposed with SPARQL queries. Applications like Geotriples (for transformation of spatial data in the form of shapefiles), RML Processor (for data in the form of CSV/ JSON), D2RQ (on the fly translation of dynamic data and/or t of data from relational databases) are used in the transformation/translation process.
- *Post Processing* of the generated RDF triples which sometimes need to be post processed to normalize the data by e.g. removing some duplicated triples or modifying some triples etc. This is performed again by the use of some customized shell scripts.
- *Data storage* is the step where the RDF triples are stored in some triple store like Virtuoso and exposing SPARQL access to them.
- *Data linking* is performed to discover links between the generated RDF triples with other existing Linked Datasets. This process is performed by using various applications (e.g. LIMES, SILK) available or through SPARQL queries.
- *Data Publication* is the step where the integrated RDF dataset is published in the triple store and exposed through some SPARQL or Faceted search endpoint/interfaces for access and exploitation for various analytical or visualization purposes over various application platforms.

- **Data Analytics** includes data processing for analysis and knowledge extraction from the published Linked Data by mainly semantic querying (e.g. SPARQL queries)
- **Data Visualization & Exploitation** is the step where the Linked Data is exploited on some application platform for the graph or map visualization of the data for the purpose of data presentation and user interaction. Web applications like Metaphactory serve the purpose of data visualization and presentation of the datasets for user interaction.

9.3.1.1 Pipeline instantiations

The main instantiations of the generic pipeline, classified according to the format of the input data source are:

- **Pipeline for geospatial data (shapefiles) transformation:** This pipeline is focused towards transformation of geospatial data in the form of shapefiles into Linked Data, using some underlying model/ontology (e.g. FOODIE ontology[106] was used in many use-cases in various projects). The pipeline uses an RDF mapping definition that specifies how to map the contents of a dataset into RDF triples using a specific ontology/vocabulary or any of its

[106] https://github.com/FOODIE-cloud/ontology/blob/master/foodie_v4.3.2.ttl

extensions. In this process first a generic R2RML definition of the mapping file is generated from the input shapefiles by using applications like GeoTriples[107] and thereafter manually edited as per the used data model vocabulary to generate the final mapping definition. Applications like GeoTriples can also be used to generate the RDF dumps from the source data contents. Once the primary dump is created some post processing might be involved as per requirements where mainly bash scripts are used as part of the pipeline activity. The RDF datasets generated are loaded into Virtuoso triple store. A SPARQL endpoint[108] and a faceted search endpoint[109] are available for querying and exploiting the Linked Data in the Virtuoso instance presently available within the PSNC infrastructure. The final task involves providing an integrated view over the original dataset. In most cases where the source datasets are particularly large (especially when considering connections with open datasets), and the connections are not of equivalence (i.e., resources are related via some geospatial relationship instead of direct equivalence, such as within, intersection, contained, etc.) it is decided to use queries to access the integrated data as per need rather than using link discovery tools like SILK or LIMES. Hence cross querying within the datasets is done in Virtuoso SPARQL endpoint for some use cases to establish possible links between agricultural and other related open datasets. Figure 24 shows the pipeline component view for the geospatial datasets.

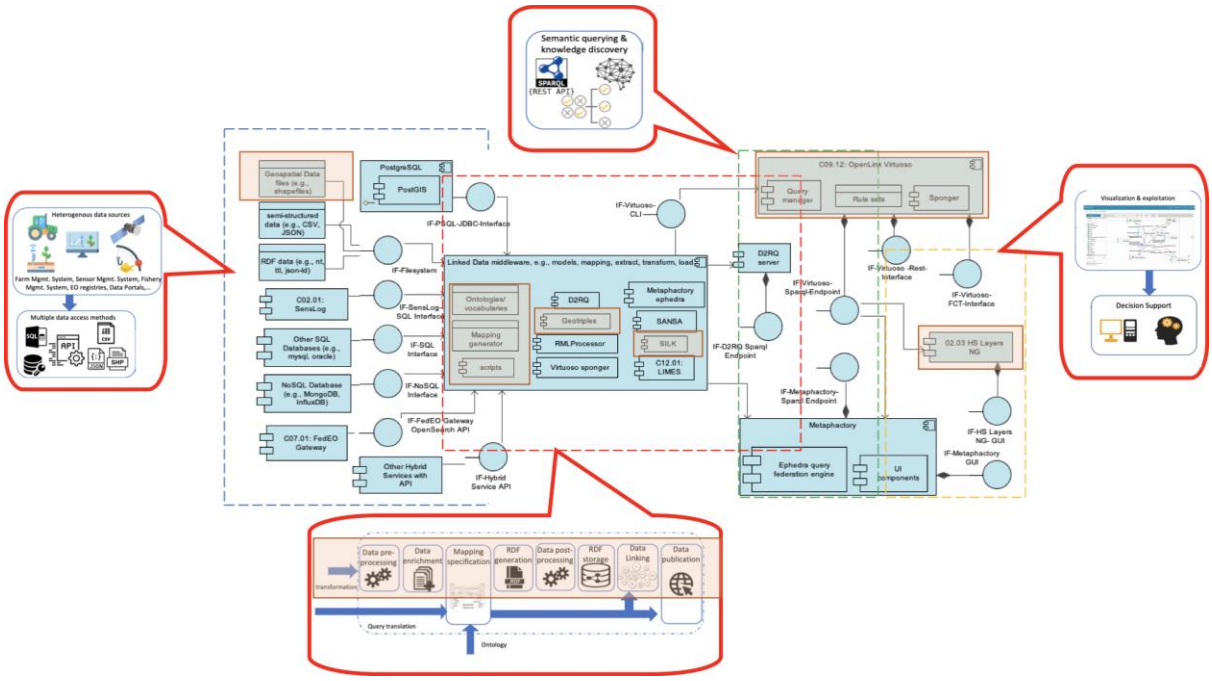


Figure 24: Pipeline components for geospatial data (shapefiles) transformation

- **Pipeline for (semi-)structured data (csv, json) transformation:** This pipeline is designed to support datasets in different semi-structure formats, including at the moment CSV and JSON files, which need to be transformed and published as Linked Data. From practical experience, initially most of these (semi-)structured data mainly the CSV files need to be pre-processed to align as per the mapping requirement. For this purpose and from the automation point of view python scripts or bash scripts are used in the pipeline. However,

[107] http://geotriples.di.uoa.gr/
[108] https://www.foodie-cloud.org/sparql
[109] https://www.foodie-cloud.org/fct/

the main goal is to have the whole process of transformation to RDF data automated which is still undergoing. In this pipeline also the main aspect is to have the RML/R2RML mapping specification that specifies how to map the contents of the dataset into RDF triples using any specific ontology/vocabulary (e.g. Data cube ontology, review, FOAF, schema.org, POI etc.). In most cases applications like Geotriples are used to generate the initial mapping using the input data in CSV or JSON format and then the mapping structures and are further modified and refined as per the modelling vocabularies. Once the mapping definition is done applications like RML processor[110] are used to generate the RDF dumps from the mapping definitions. Later these dumps might need some post processing actions mainly to remove duplicates triples or unwanted data triples which were done using mainly bash scripts. The resulting RDF datasets are then loaded into Virtuoso triple store providing SPARQL and faceted search endpoints for further exploitation. Finally, for the provision of an integrated view over the original datasets in case of agricultural and open data, SPARQL queries are generated. The visualization of the integrated data is discussed in the further sections. Figure 25 below shows the pipeline component view for the (semi-)structured datasets.

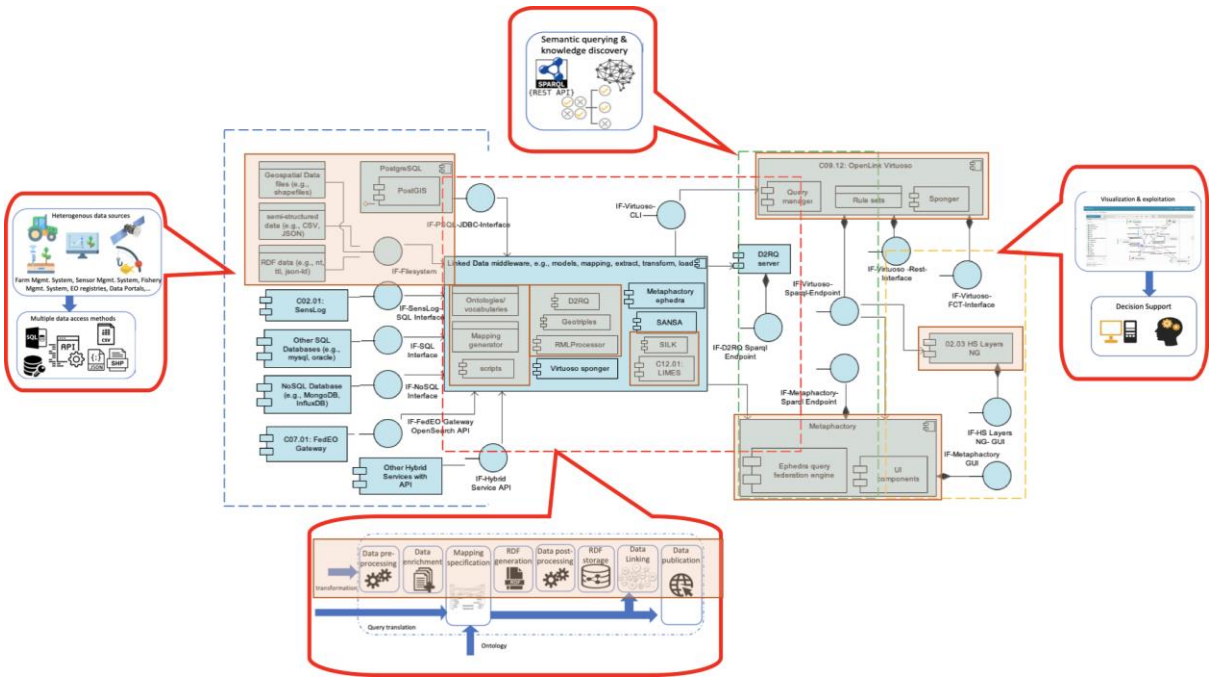


Figure 25: Pipeline components for (semi-)structure data transformation

- **Pipeline for relational databases translation:** This pipeline involves the transformation of dynamic data from relational databases into Linked Data on the fly, i.e. data stays at the source and only a virtual semantic layer is created on top of it to access it as Linked Data. This is mostly done for dynamic data like sensor data and the pipeline is effective in that mode of translation. The data from relational datasets first needs to be modelled and mapped in RDF definitions, for modelling the data well known vocabularies are used and in this case sensor data related vocabularies e.g. SSN ontology[111], Data cube ontology[112] etc. The input data comes from a relational database (e.g. PostgreSQL) that stores the dynamic data. Hence, in the mapping stage, the creation of R2RML/RML definitions requires different

[110] https://github.com/RMLio/RML-Processor
[111] https://www.w3.org/TR/vocab-ssn/
[112] https://www.w3.org/TR/vocab-data-cube/

pre-processing tasks and some on-the-fly assumptions to engineer the alignment between the sensor database and the ontologies/vocabularies. Once the mapping file is generated (manually), the RDF Data of the dataset can be published using D2RQ[113] server that enables accessing relational database sources as virtual RDF graphs. This on-the-fly approach allows publishing of RDF data from large and/or live databases and thus the need for replicating the data into a dedicated RDF triple store is not required. For example, the Linked Data from the sensor data from SensLog[114] (version 1) is already published in the PSNC infrastructure in a D2RQ server[115]. The associated SPARQL endpoint to query the data is available at: http://senslogrdf.foodie-cloud.org/sparql. Figure 26 below shows the components involved in the pipeline for transformation of data from relational databases.

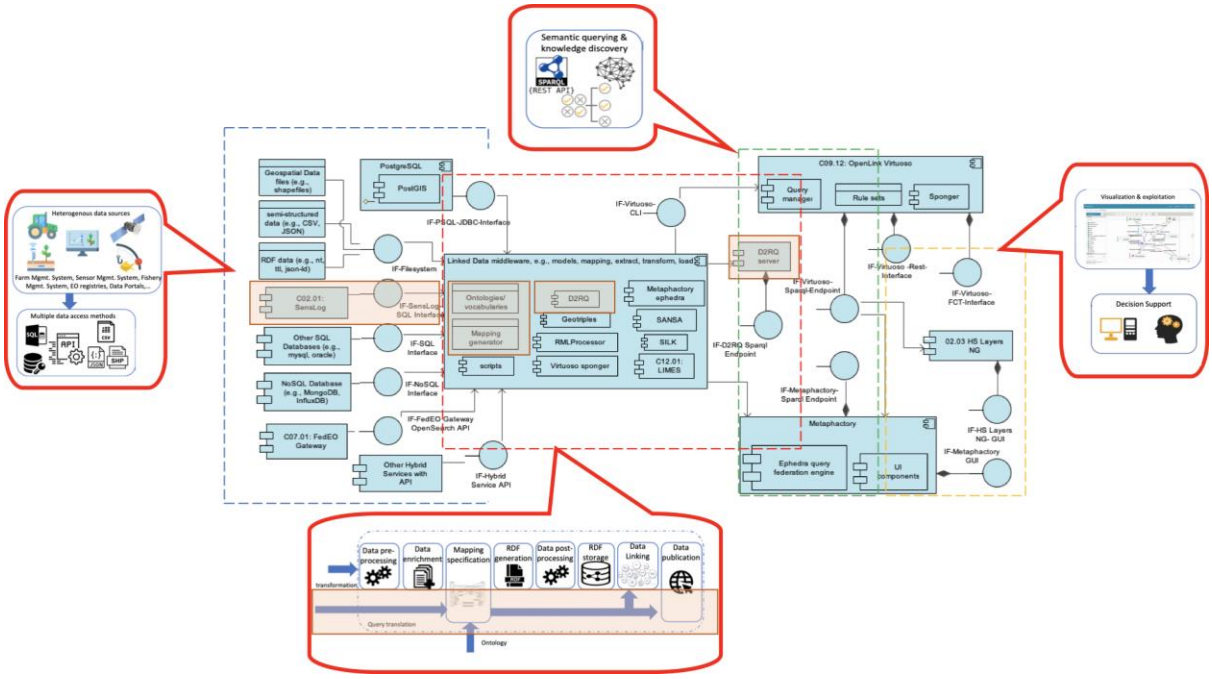


Figure 26: Pipeline components for relational databases transformation

- **Pipeline for hybrid services translation**: This pipeline focuses mainly on the publication of Linked Data from hybrid data origins; additionally, on making mainly structured metadata available as Linked Data via a SPARQL compliant endpoint which makes requests to non-sparql backends on-the-fly. Hence, it is targeted to enable querying via SPARQL without harvesting all the metadata and storing the data in a triple store but to access them dynamically via the existing on-line interfaces. In broader terms the input data is accessed via some API gateway (e.g. FedEO gateway[116]). Generally, in the pipeline regarding the modelling ontologies for hybrid data the main idea is to reuse the standard and/or widely used ontologies/vocabularies whenever it's possible and if possible, to extend them as needed. The use cases where this pipeline is already being used has the input data mainly in structured format e.g. in (Geo)JSON-LD [117] representation, thus it is required only to expose the results as Linked Data. In this case a component of API wrapper (e.g. Ephedra from

[113] http://d2rq.org/
[114] http://www.senslog.org/
[115] http://senslogrdf.foodie-cloud.org/
[116] http://ceos.org/ourwork/workinggroups/wgiss/access/fedeo/
[117] https://geojson.org/geojson-ld/

Metaphactory platform[118]) is used to access the data by using SPARQL description of a REST Service Signature defining the input and output terms and thereafter configuring a REST Service Repository. Once the RDF data is generated the data can be exposed via a SPARQL endpoint provided in some online platform like we have used Metaphactory platform[119]. A demo interface has also been implemented to visualize the linked data in Metaphactory that will be presented in the later sections. Figure 27 shows the components used in the pipeline.

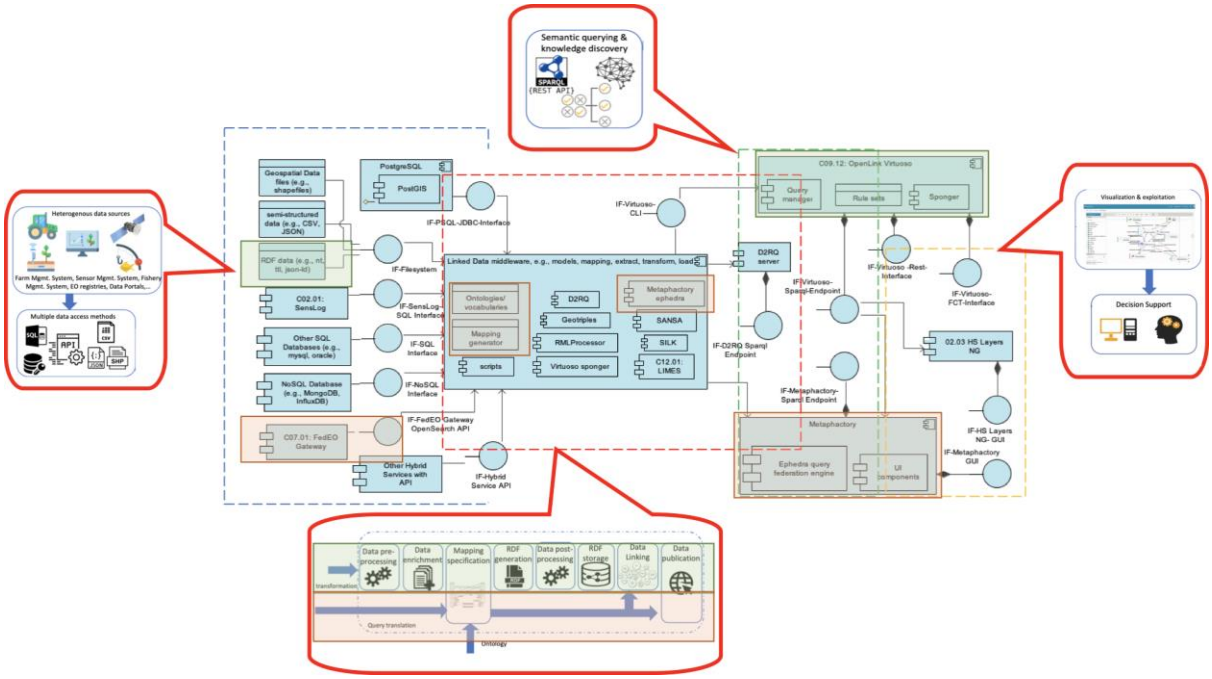


Figure 27: Pipeline components used for hybrid services transformation

9.3.1.2 Pipeline adaptations & extensions in DEMETER

The pipelines already available, which are discussed in the previous section, will provide the basis for DEMETER, instead of creating everything from scratch. However, as part of DEMETER, these pipelines will have to be adapted/extended, new pipelines will need to be created and a DEMETER API will have to be implemented. In particular, the following main tasks will have to be carried out in DEMETER:

- Implement and use DEMETER AIM as the target model to represent data in the transformations/translations
- Existing pipelines will require adjustments and extensions, including updates in the mapping specifications and wrapper implementations. Additionally, existing pipelines will have to be extended, and new pipelines will have to be created in order to support the processing and integration of the different DEMETER datasets. For instance, new data collection methods and pre/post processing services will have to be implemented, new wrappers and mappings will have to be generated, and new tools will have to be integrated.
- The pipelines and their underlying facilities will comprise a data preparation & integration enabler in DEMETER, which will expose a DEMETER-enabled API allowing to reuse and automatize the provided functionalities. So, while the pipelines use many different tools/services via different interfaces and protocols (API, CLI) to prepare and integrate data,

[118] https://help.metaphacts.com/resource/Help:Ephedra

[119] http://metaphactory.foodie-cloud.org/sparql?repository=ephedra

the DEMETER API will abstract these differences and provide a single and common access point to these facilities. The API will also provide access to data based on AIM model, abstracting complexities in using SPARQL queries, at least for some basic/common use case scenarios.

9.3.2 UML sequence diagram

The UML diagram below depicts the sequence of steps for all the main data stages involved in the preparation and integration of data using Linked Data as a federated layer. Figure 28 depicts the two main data integration sequences: in green are depicted the main steps related to the transformation of existing datasets, for example local or isolated files with semi-structured data that are not accessible directly by services/apps; in orange are depicted the main steps related to the translation of existing datasets available though some interface (API), dynamically and directly from their source, such as relational data and hybrid services (e.g., data source with Rest API).

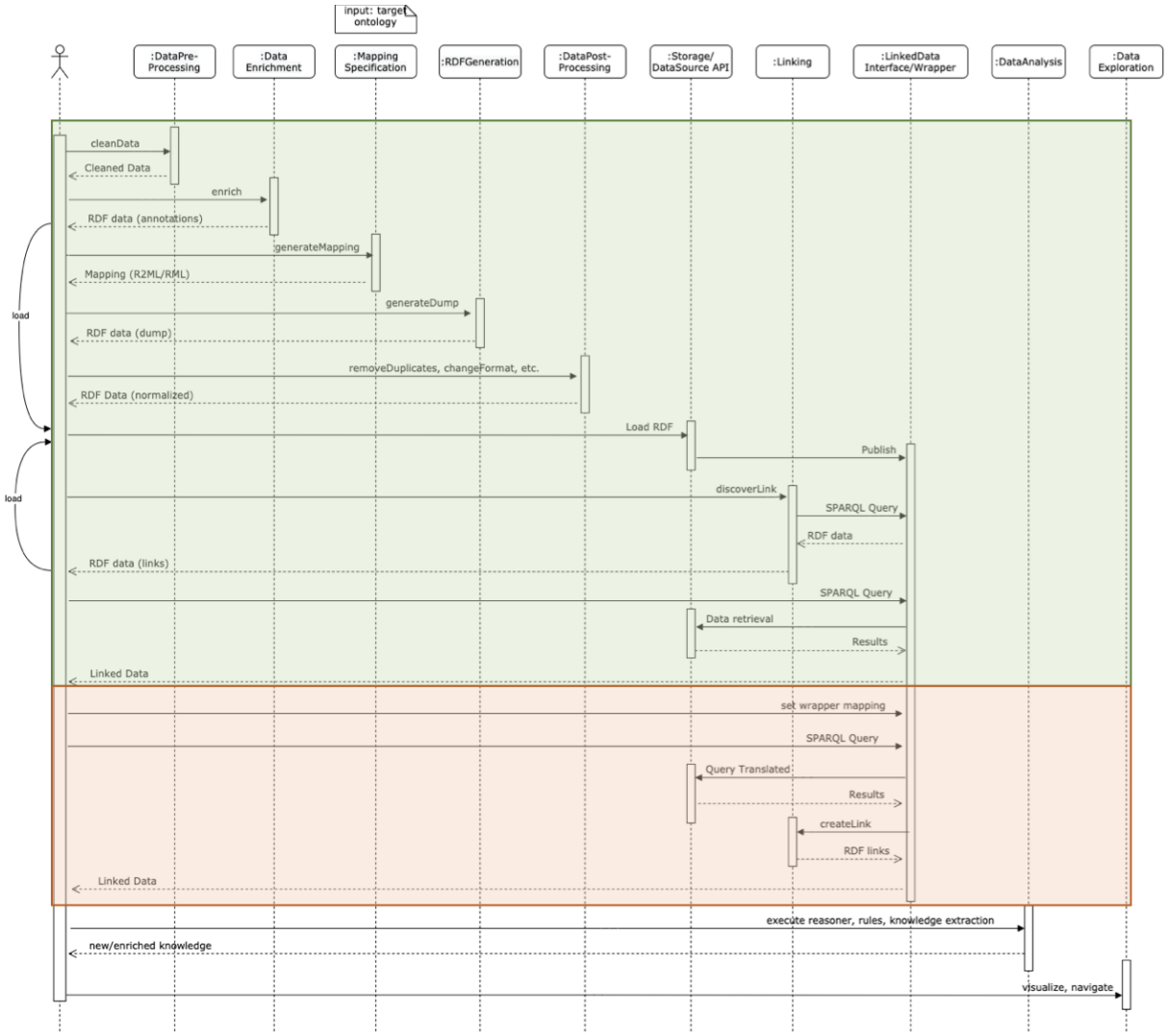


Figure 28: Linked Data preparation and Integration UML sequence diagram

9.3.3 Data Preparation & Integration Enabler in DEMETER

The facilities for data preparation and integration constitute one of the enablers of the higher-level Data & Knowledge DEMETER Enablers, as depicted in Figure 6 in Section 7. This enabler has strong ties and interactions with the other enablers in this module, including the data management, data

fusion, data analytics & knowledge extraction. Figure 29 depicts the facilities in this enabler, and the relations with other enablers. Additionally, this enabler may be used, not only by other DEMETER enablers, but also directly by DEMETER enhanced entities (e.g., services or apps in different pilots).

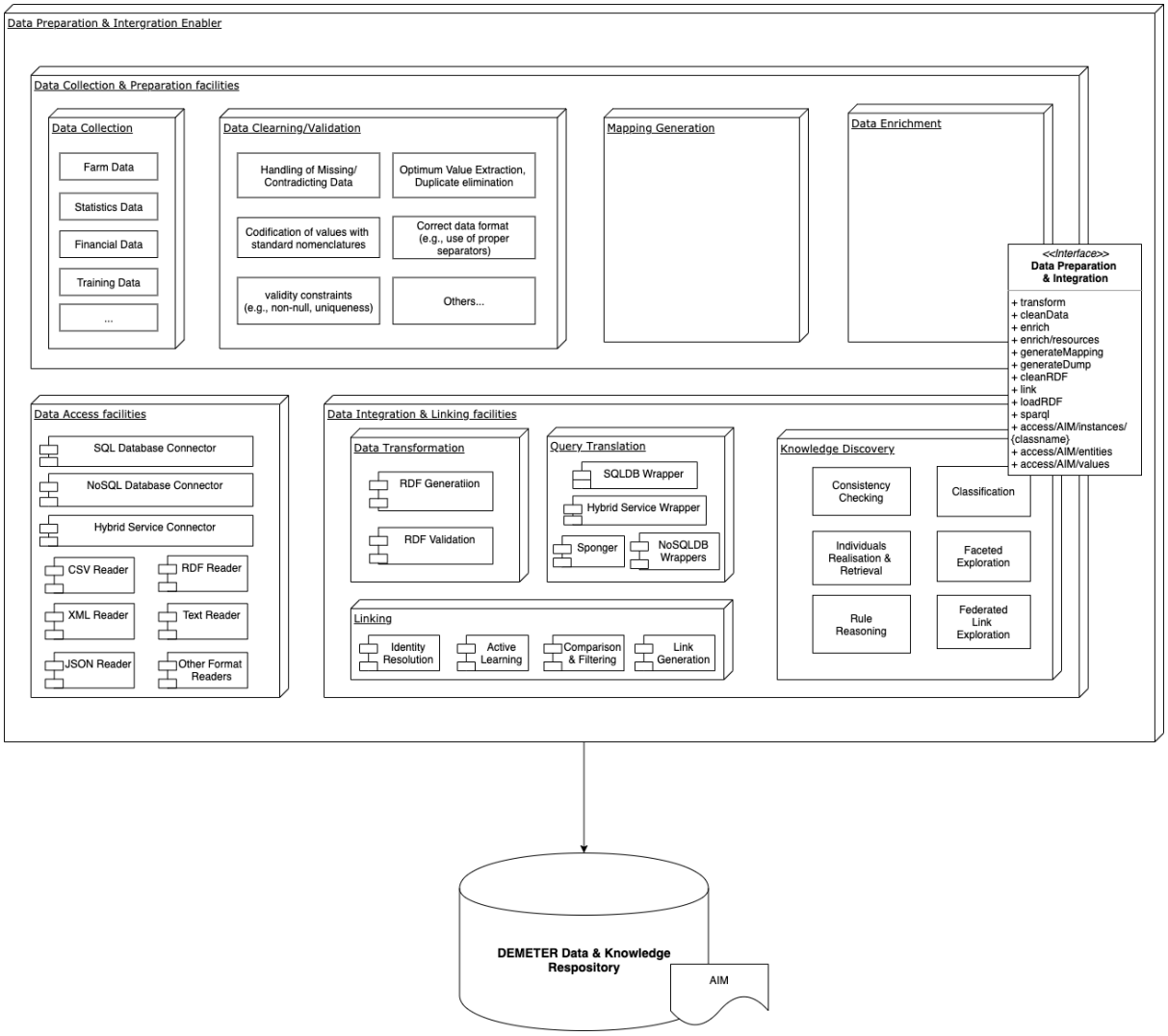


Figure 29: Data Preparation & Integration Enabler facilities and relationships

9.4 Implementation

This section mainly describes the components and tools used in the above-mentioned pipelines in the process of transformation of the Linked Data. These pipelines implemented and deployed in various projects include many different tools and data processing components, as well as models, mappings, and a large number of linked datasets generated in the course of those projects. The main tools and applications used during the various stages of the process are summarized below. Note that most of the underlying components are open source, and models and mappings[120] are also openly available, as will be described below.

120 https://github.com/FOODIE-cloud/ontology/tree/master/mappings/new_mappings

9.4.1 Components

9.4.1.1 Virtuoso (incl. sponger, faceted browser)

OpenLink Virtuoso[121] (open source edition of Virtuoso Universal Server)[122] is a middleware and database engine hybrid that combines the functionality of a traditional RDBMS (Relational Database Management System), ORDBMS (Object-Relational Database Management System.), virtual database, RDF, XML, free-text content management & full-text indexing, linked data server, web application server and file server functionality in a single system. Primarily the RDF store is the most important feature of Virtuoso for the provisioning of a semantic database (triplestore) as a service to store the RDF data, as well as for their publication as Linked Data. Along with the RDF triplestore Virtuoso also provides a SPARQL query language support, SPARQL protocol supports inline SPARQL integration within SQL, use of bitmap indices for optimizing storage and management of RDF triples, implementation of the HTTP-based Semantic Bank API that enables client applications to post to its RDF Triple Store, and several RDF insert methods, including http PUT and POST. Virtuoso SPARQL can be used as an inference context for inferring triples (not physically stored) by supporting some RDFS and OWL constraints (e.g., owl:sameAs, rdfs:subClassOf etc.). Virtuoso also includes a faceted search interface which allows simple text search and navigation of RDF data through their links. Virtuoso also includes Sponger which is a Linked Data middleware component. It generates Linked Data from a variety of data sources, and supports a wide variety of data representation and serialization formats (e.g., standard non-rdf like csv, atom, rss, etc., and vendor-specific like Facebook, google+, geonames, OpenStreetMap, etc.). The Sponger is also a full-fledged HTTP proxy service, directly accessible via SOAP or REST interfaces. Similarly, Virtuoso includes RDBMS-to-RDF mapping functionality (also known as Linked Data Views of SQL data). Recently as of Virtuoso Enterprise Edition Release 8.2 and Virtuoso Open Source Edition Release 7.2.6, a number of major enhancements have been made to Geospatial support, improving the Geometry data types and functions, and adding support for the OGC GeoSPARQL 11 standard. The OGC GeoSPARQL 11 standard addresses the need for standardized representation and interaction with geospatial data via SPARQL Query Language extensions. This manifests as new data types, magic predicates, and a vocabulary of terms for describing geospatial data using RDF statements that are inserted to a compliant database (or store) using INSERT statements and/or bulk loading. Some of the common WKT (Well Known Text) representations for several types of geometric objects used in RDF are: Multipoint, LineString, MultiLineString, Polygon, MultiPolygon etc. The endpoints available for the Virtuoso instance, with a description of the exposed APIs are:

- SPARQL endpoint is current available at https://www.foodie-cloud.org/sparql[123]. Virtuoso SPARQL endpoint also provides a REST API having the following functionalities[124]:
    - POST: This method can be used to perform SPARQL 1.1 SELECT queries and also SPARQL 1.1 update operations: INSERT/UPDATE/DELETE.
    - GET: This method can be used to perform SPARQL 1.1 SELECT queries. With GET methods you can get the triples which are saved in the whole triplestore or in any particular named graph identified by IRI.
    - PUT: Using this method we can load data to a named graph identified by the provided IRI. The data is provided as RDF content constructed using any RDF

[121] https://virtuoso.openlinksw.com/

[122] http://vos.openlinksw.com/owiki/wiki/VOS

[123] Note that endpoint exposed by DEMETER for this and other exposed services will be a DEMETER one, as described in the API Section 8.5

[124] See examples at: http://vos.openlinksw.com/owiki/wiki/VOS/VirtGraphProtocolCURLExamples#HTTP%20PUT%20Example

concrete syntax (or notation e.g., N-Triples, Turtle, JSON-LD, RDF/XML). This method can also be used to upload RDF resources into Virtuoso's WebDAV repository and then automatically upload the triples within the resource into the Virtuoso Quad store[125]

- DELETE: This method can be used to delete triples from the named graph identified by the provided IRI.

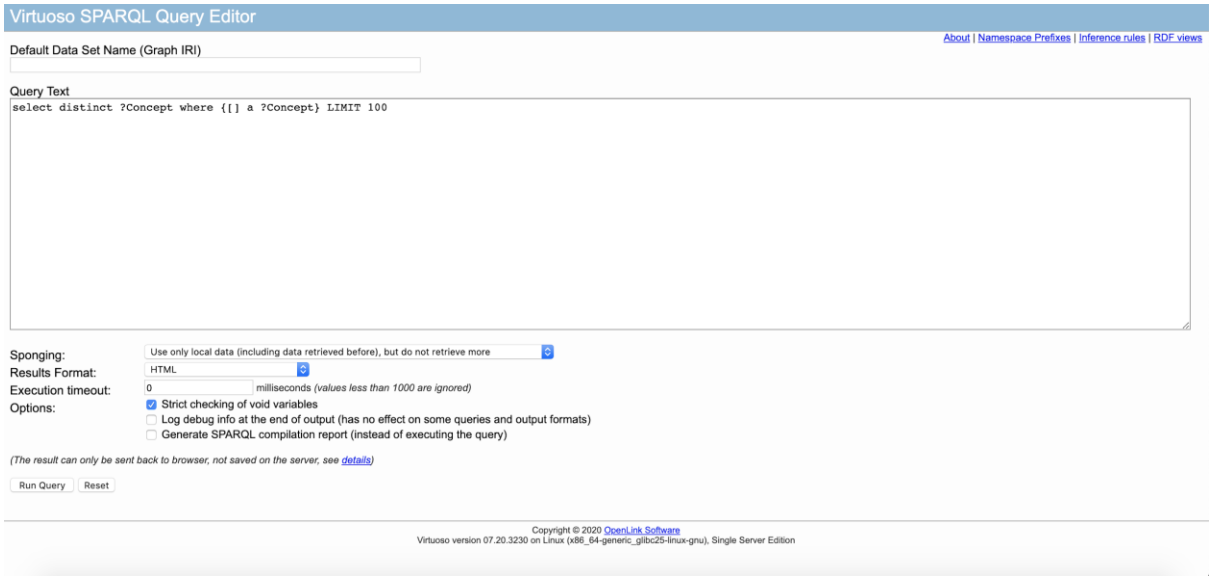


Figure 30: SPARQL GUI of Virtuoso

- Faceted search endpoint available at https://www.foodie-cloud.org/fct. Virtuoso also have APIs for FCT REST services that enable the use of Virtuoso's VSP/VSPX technology to produce (X)HTML-based Linked Data explorer pages that are endowed with high-performance (in-process) faceted browsing capability. This service enables faceted browsing over Linked Data hosted in the RDF Quad Store. This also includes Linked Data that is progressively added to the Quad Store via URI de-referencing. The Virtuoso Facets web service provide the following REST interface[126]:
  - Service Description:
    - The service supports only POST methods.
    - The content type must be in the form of txt/xml format.
    - The entity body must be an XML document with the top element as described in the query.
    - The request response namespace MUST be http://openlinksw.com/services/facets/1.0
  - Error Conditions: All the error conditions are reported via '<error>Error explanation</error>'
  - The facet_svc.sql contains web service code and virtual directory mapping, and it uses fct_req.xsl and fct_resp.xsl as request and response filters.

[125] See http://vos.openlinksw.com/owiki/wiki/VOS/VirtRDFInsert#HTTP%20PUT

[126] http://vos.openlinksw.com/owiki/wiki/VOS/VirtuosoFacetsWebService#REST%20Interface

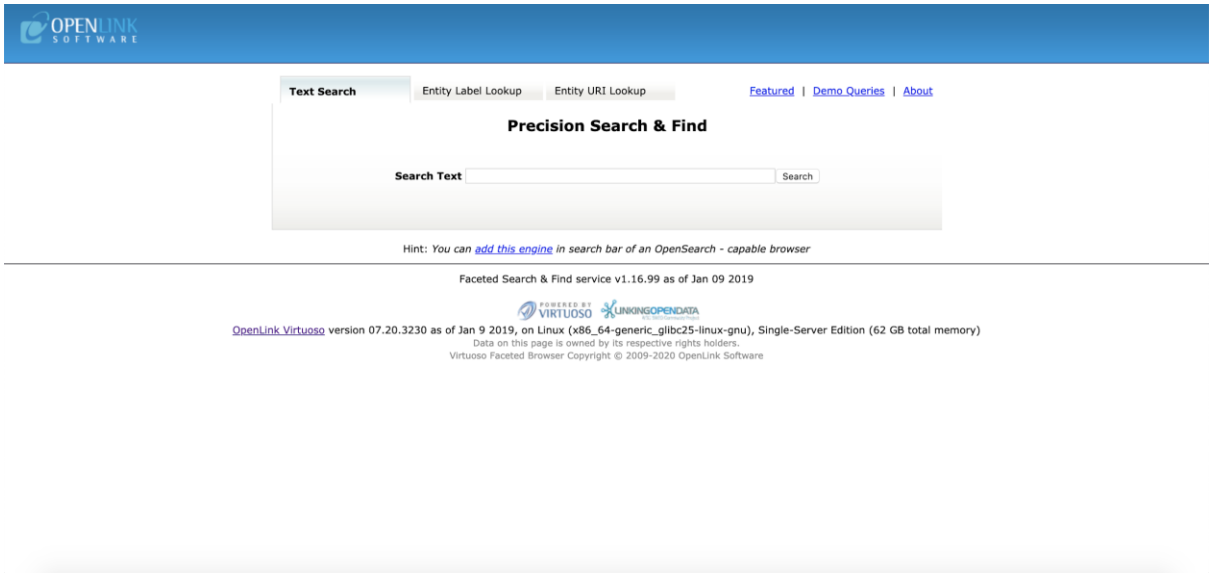


Figure 31: Faceted Search GUI of Virtuoso

- Sponger endpoint: Virtuoso includes Sponger that is a Linked Data middleware component that generates Linked Data from a variety of data sources, and supports a wide variety of data representation and serialization formats (e.g., standard non-rdf like csv, atom, rss, etc., and vendor-specific like Facebook, google+, geonames, OpenStreetMap, etc.). The Sponger is also a full-fledged HTTP proxy service directly accessible via SOAP or REST interfaces. The RESTful applications use Sponger via proxy. The following presents list of the supported parameters:
    - *refresh*: This parameter is used for overwriting that explicitly clears the graph i.e. it will cause the Sponger to drop cache even if it is marked to be on the fly.
    - *sponger:get :* This parameter progressively adds new triples to named graphs. This is the default value for the parameter sponger:get. It can also be used together with *refresh=<seconds>* to overwrite the expiration in the cache.
    - *sponger:get :* This parameter can be used to cache invalidation mode and associated rules of instance e.g. Network Resource Fetch data with option soft and refresh. May be used together with *refresh=<seconds>* to overwrite the expiration in the cache.
    - *sponger:get* : This parameter having value *replace* is capable of replacing from non-fetched triples and can also be used with the refresh option.

  The sponger endpoint can be accessed along with the URL parameters at https://www.foodie-cloud.org/about/

9.4.1.2 D2RQ

D2RQ[127] is a system for accessing relational databases as virtual, read-only RDF graphs. It offers RDF-based access to the content of a relational database without having to replicate it into an RDF store. Using D2RQ we can:

- query a non-RDF database using SPARQL

[127] http://d2rq.org/

- access the content of the database on the fly as Linked Data over the Web.
- create custom dumps of the database in RDF formats for loading into an RDF store
- access information in a non-RDF database using the Apache Jena API

The D2RQ Platform consists of:

- the D2RQ Mapping Language, a declarative mapping language for describing the relation between an ontology and a relational data model.
- the D2RQ Engine, a plug-in for the Jena Semantic Web toolkit, which uses the mappings to rewrite Jena API calls to SQL queries against the database and passes query results up to the higher layers of the frameworks.
- D2R Server, an HTTP server that provides a Linked Data view, a HTML view for debugging and a SPARQL Protocol endpoint over the database

D2R Server can be started from the command line, or as a J2EE web application inside an existing servlet container, such as Apache Tomcat or Jetty. The D2R Server uses a customizable D2RQ mapping to map database content into this format, and allows the RDF data to be browsed and searched which are the two main access paradigms to the Semantic Web. D2RQ has a command line interface and the mapping as well as translation are done through commands. The on-the-fly translation allows publishing of dynamic RDF data from large live databases and eliminates the need for replicating the data into a dedicated RDF triple store (e.g. dynamic sensor data). In the pipeline this application is used extensively in the translation of dynamic data from the relational database. As for example, the Linked Data from the sensor data from SensLog (version 1) is already published in the PSNC infrastructure in an open source version of D2RQ server available at http://senslogrdf.foodie-cloud.org/.

The associated SPARQL endpoint to query the data is available at: http://senslogrdf.foodie-cloud.org/sparql. This D2RQ endpoint also provides a REST API having supporting the operations:

- GET & POST to perform SPARQL SELECT queries

D2RQ also includes different tools exposing a command line interface, including:

- generate-mapping: creates a D2RQ default mapping file by analyzing the schema of an existing database. This mapping maps each table to a new RDFS class based on the table's name, and maps each column to a property based on the column's name. This mapping file can be used as-is or can be customized.
- d2r-query: allows executing SPARQL queries against a D2RQ-mapped relational database from the command line, with or without a D2RQ mapping file. If a mapping file is specified, the tool queries the virtual RDF graph defined by the mapping. If no mapping file is specified, the tool uses the default mapping of generate-mapping for the translation
- dump-rdf: dump the contents of the whole database into a single RDF file, with or without a D2RQ mapping file. If a mapping file is specified, the tool uses it to translate the database contents to RDF. If no mapping file is specified, the tool uses the default mapping of generate-mapping for the translation.

Below are a few screenshots of the GUI provided by the senslog D2RQ server instance:

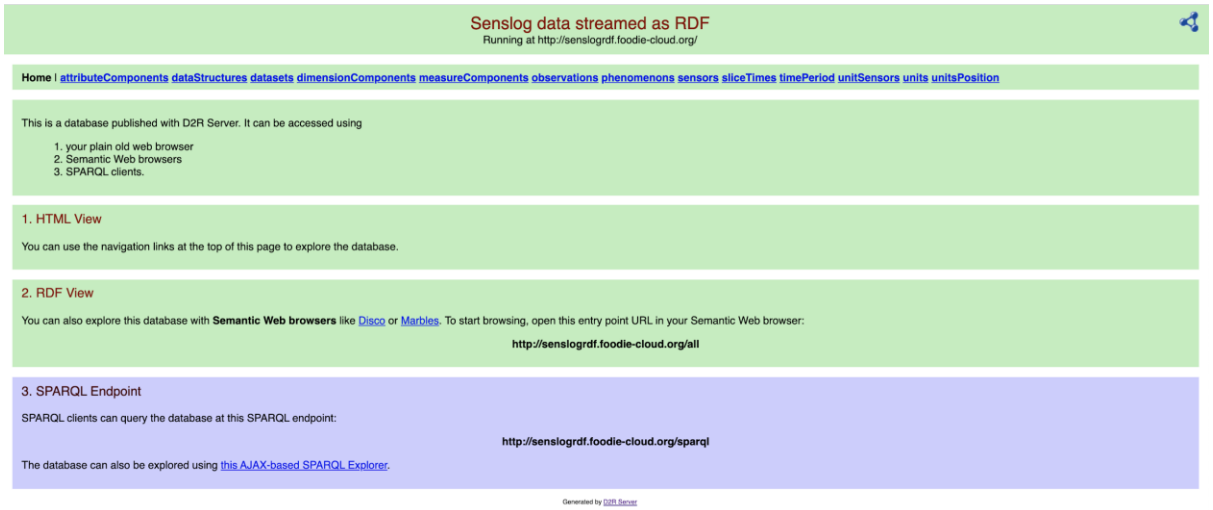


Figure 32: D2RQ server

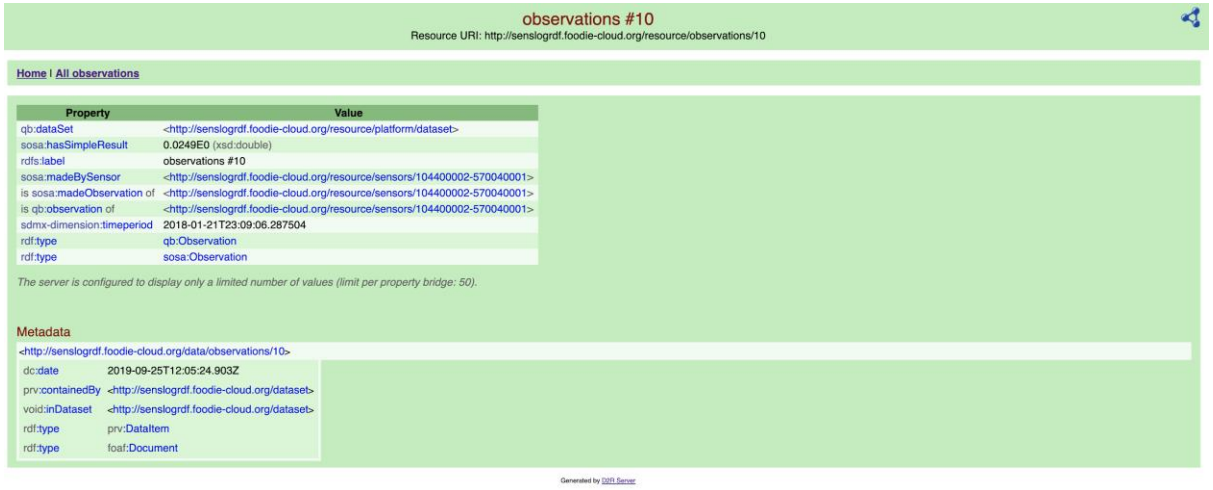


Figure 33: Visualization of an observation details in RDF generated on-the-fly

9.4.1.3 Geotriples

This is an open source tool used for transforming geospatial data from their original formats (e.g., shapefiles or spatially-enabled relational databases) into RDF. The following input formats are supported: spatially-enabled relational databases (PostGIS and MonetDB), ESRI shapefiles and XML, GML, KML, JSON, GeoJSON and CSV documents.

GeoTriples[128] comprises two main components: the mapping generator and the R2RML/RML mapping processor. The mapping generator takes as input a geospatial data source (e.g., a shapefile) and creates automatically an R2RML or RML mapping that can transform the input into an RDF graph which uses the GeoSPARQL vocabulary. The mapping statements can be customized based on any other data models or ontologies in course of its transformation into RDF triples. In the pipeline processes this tool has proved to be a very effective one as it has a versatile usage in many of the pipeline tasks.

[128] http://geotriples.di.uoa.gr/

This tool has a command line interface that supports the following main commands:

- The command *geotriples-cmd generate_mapping* is used to get the RML/R2RML mapping statements for shapefiles, CSV or JSON files. In this command the parameters supported are "-o" for the path to the output file, "-b" for the namespace of the named graph along with the type of input file provided. An example format is given below:

  ./geotriples-cmd generate_mapping -o "path to output file (ttl format)" -b "namespace used" "path to the input shapefile, CSV or JSON file"

- The command *./geotriples-cmd dump_rdf* is used to generate RDF dumps from the mapping specification by the parameters, "-o" and "-sh" as this function is limited to only shapefiles input format, for example as below:

  ./geotriples-cmd dump_rdf -o "path of the output dump file" -b "namespace used" -sh "path to the used input shape file" "path of the mapping specification used"

9.4.1.4 RML Processor

This open source tool RML Processor[129] can be used to generate RDF data using any type of semi-structured data. There is support for different formats, such as CSV, JSON, and XML, together with support for different data sources, such as files, databases, Web APIs, and streams. A mapping process can be executed with the following command using the command line interface.

java -jar target/RML-Procssor-0.2.jar -m <mapping_file> -o <output_file> -f <output_format> [-g <graph> -tm <triples_map>]

where:

- <mapping_file> is the RML mapping file conforming with the RML specification http://rml.io/spec.html)
- <output_file> is the file where the output RDF triples are stored
- <output_format> is the preferred output format supporting the following: turtle, ntriples, nquads, rdfxml, rdfjson, jsonld.
- <graph> (optional) is the named graph in which the output RDF triples are stored.
- <triples_map> (optional) is the specific Triples Map of the mapping document to be executed.

9.4.1.5 FOODIE Semantic Annotation Service:

FOODIE semantic annotation service provides a simple open source REST API designed for creating, updating and retrieving semantic annotations. The service orchestrates other components (existing annotation tools described below) in order to control and fully perform data analysis process, creates semantic form of the generated data and persists semantic data using semantic store.

Annotations created by the service are modelled using the Modular Unified Tagging Ontology (MUTO)[130] which is designed specifically for tagging and folksonomies. MUTO allows representing public and private tagging, simple and auto generated tags and others. It is also easily extensible since all concepts defined in MUTO ontology inherits from other more general ontologies like

[129] https://github.com/RMLio/RML-Processor

[130] https://dl.acm.org/doi/pdf/10.1145/2063518.2063531

SKOS[131], SIOC[132] or vocabularies as RDFS[133]. The service includes external tools used as libraries or services that allows keywords extraction, sentence disambiguation and text meaning identification. The results generated by these tools are then used to create semantic representation of the annotations.

The current implementation of the semantic annotation service uses the following annotation tools:

- **AgroTagger** is a keyword extractor that uses the AGROVOC thesaurus[134] as its set of allowable keywords. It also identifies concepts from AGROVOC vocabulary in terms of Linked Open Data. AGROVOC is a controlled vocabulary covering all areas of interest of the Food and Agriculture Organization (FAO) of the United Nations, including food, nutrition, agriculture, fisheries, forestry, environment etc. It is published by FAO and edited by a community of experts.
- **Babelfy** is a unified, multilingual, graph-based approach to Entity Linking and Word Sense Disambiguation based on a loose identification of candidate meanings coupled with a densest subgraph heuristic, which selects high-coherence semantic interpretations.

The REST API[135] associated with the Semantic Annotation Service have the following functionalities:

- *POST /tagging*: Creates new tagging for requested list of files and text description. Tagging will consist of a set of tags (annotations) identified by an-notation tools. It also creates tagged resource instances. If resource instance URI is provided in the request, the method updates existing tagging.
- *GET /tagging:* Retrieves tag labels for the resource and language specified in the request.
- *POST /tag*: Adds a new tag to the existing tagging.
- *GET /statistics/meaning:* Retrieves statistics showing which concepts are most commonly identified in the annotated resources.
- *GET /resources:* Gets resources tagged by tags with specified meaning.

The tagging creation request may contain plain text description or set of files. The current implementation of the semantic annotation service supports processing of the following file types: PDF, DOC, XLS, HTML, TXT. A simple web client application[136] , and API documentation (swagger)[137] , are available for testing.

9.4.1.6 Silk

Silk[138] Workbench is a web application which guides the user through the process of interlinking between different data sources. Silk Workbench offers the following features:

- Enabling users to manage different sets of data sources, linking tasks and transformation tasks.
- Offers a graphical editor which enables the user to easily create and edit linking tasks and transformation tasks.

131 https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html
132 https://www.w3.org/Submission/sioc-spec/
133 https://www.w3.org/TR/rdf-schema/
134 http://aims.fao.org/vest-registry/vocabularies/agrovoc
135 http://www.foodie-cloud.org/swagger_ui/?url=http://www.foodie-cloud.org/swagger_api/Semantic_Annotation_API.json
136 https://www.foodie-cloud.org/semanticAnnotation/
137 http://www.foodie-cloud.org/swagger_ui/?url=http://www.foodie-cloud.org/swagger_api/Semantic_Annotation_API.json
138 http://silkframework.org/

- Silk Workbench makes it easy for the user to quickly evaluate the links which are generated by the current link specification.
- It allows the user to create and edit a set of reference links used to evaluate the current link specification.

The open source version of the SILK instance deployed in PSNC is available at http://silk.foodie-cloud.org/ .

In addition to the Workbench, Silk provides three different command line applications for executing link specifications:

- Silk Single Machine: allows to generate RDF links on a single machine. The datasets that should be interlinked can either reside on the same machine or on remote machines which are accessed via the SPARQL protocol.
- Silk MapReduce: allows to generate RDF links between data sets using a cluster of multiple machines, based on Hadoop, and can for instance be run on Amazon Elastic MapReduce.
- Silk Server: can be used as an identity resolution component within applications that consume Linked Data from the Web. Silk Server provides a REST API for matching entities from an incoming stream of RDF data while keeping track of known entities.

The SILK REST API implements the following methods:

- Manage Projects
  - *GET project*: Retrieves a JSON listing all projects in the workspace and their tasks by type
  - *PUT projects/<project>*: Adds a new empty project
  - *DELETE projects/<project>*: Deletes an existing project
- Resources (e.g. files in a project)
  - *GET projects/<project>/resource*: Retrieves a JSON listing of all resources in a project.
  - *GET projects/<project>/resources/<name>* :Retrieves a specific resource from a project
  - *PUT projects/<project>/resources/<name>:* Uploads a specific resource to a project.
  - *DELETE projects/<project>/resources/<name>:* Deletes a specific resource from a project
- Datasets (have a description that holds all properties needed to read entities from a dataset. The dataset may either be local e.g., a resource or remote e.g., accessed through queries.)
  - *GET projects/<project>/datasets/<name>*: Retrieves the properties of a specific dataset from a project.
  - *PUT projects/<project>/datasets/<name>*: Creates or updates a dataset in a project.
  - *DELETE projects/<project>/datasets/<name>*: Deletes a dataset.
- Start/Stop Activities (An activity is a unit of work that can be executed in the background in a project)
  - *POST activities/start*: Starts an activity.
  - *POST activities/cancel:* Cancels an activity.
  - *GET activities/config*: Retrieves the configuration of an activity as key-value pairs.
  - *POST activities/config*: Updates the configuration of an activity.
  - *GET activities/status*: Retrieves the status of an activity.
  - *GET activities/updates*: Retrieves a Comet stream of JavaScript calls to updateStatus whenever the status changes.

All resources support three parameters: i) project i.e. the project name; ii) task i.e. the task name and iii) activity i.e. the name of the activity. For example, for starting the *Generate Links* activity for the any task the expression is:

POST activities/start?project="name of project"&task="name of task"&activity=Generate%20Links

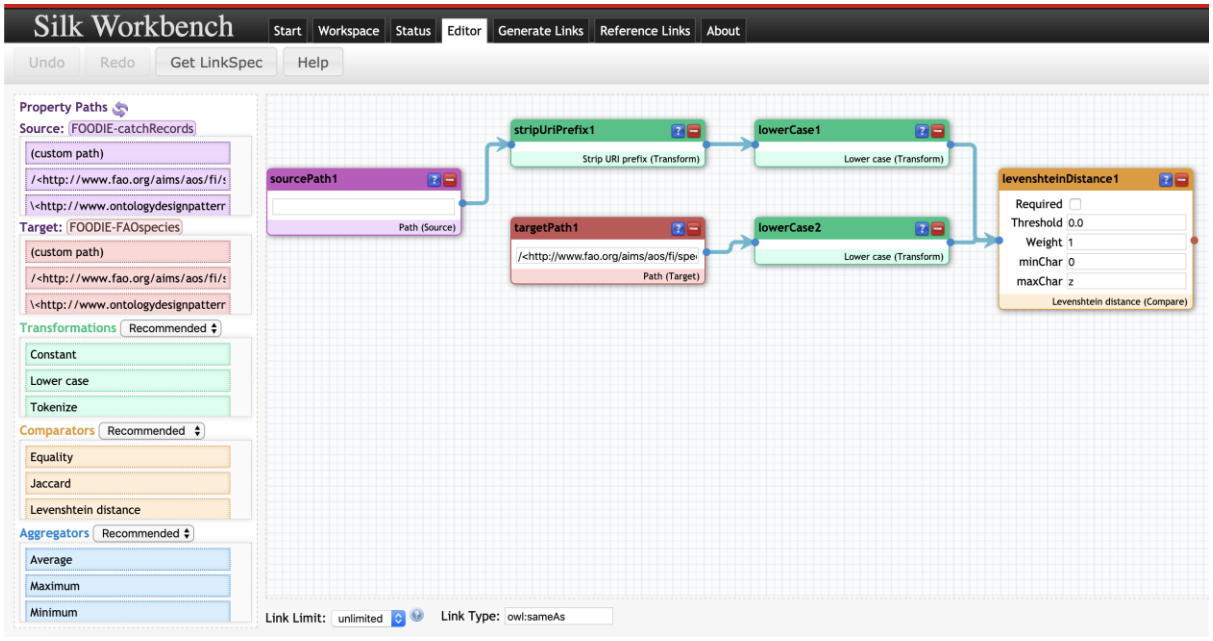


Figure 34: Workflow of the SILK workspace for new link generations

9.4.1.7 LIMES

LIMES[139] (Link Discovery Framework for metric spaces) is an open source, easy and efficient approach for the discovery of the links between various Linked Data sources. It addresses the scalability problem of link discovery by utilizing the triangle inequality in metric spaces to compute estimates of instance similarities. Based on these approximations, LIMES can filter out a large number of instance pairs that cannot suffice the matching condition set by the user. The real similarities of the remaining instances are then computed and the matching instances are returned. Large-scale link discovery in Linked Data sources based on the characteristics of metric spaces computes pessimistic approximations of the similarity between instances of different data sources and filters out the instances that do not suffice the mapping conditions (non-related data instances).

LIMES can be executed via the graphical interface after downloading the package and run it locally or in a server, or it can be executed via the command line as a Java executable package.

9.4.1.8 Geo-L

Geo-L[140] is an open source tool for discovery of geo-spatial links that retrieves specific properties of spatial objects from source and target datasets, through their respective SPARQL endpoints, and finds topological relations between objects in source and target objects according to topological predicates. The specifications of the relevant properties are provided in a configuration file, which allows constraining the number of objects by specifying offset and limit . A dataset can be created

[139] http://aksw.org/Projects/LIMES.html
[140] https://github.com/DServSys/geo-L

through properties which already exist in the graph, and, in addition, Geo-L allows direct construction of ad-hoc values through a SPARQL SELECT statement for a given resource.

Geo-L can be run from the command line or as a server with a REST API.

- Command line : To run geo-L from the command line, a config file and a database config file are needed[141] with the command:

  python main.py -c config.json -d postgresql_config.json

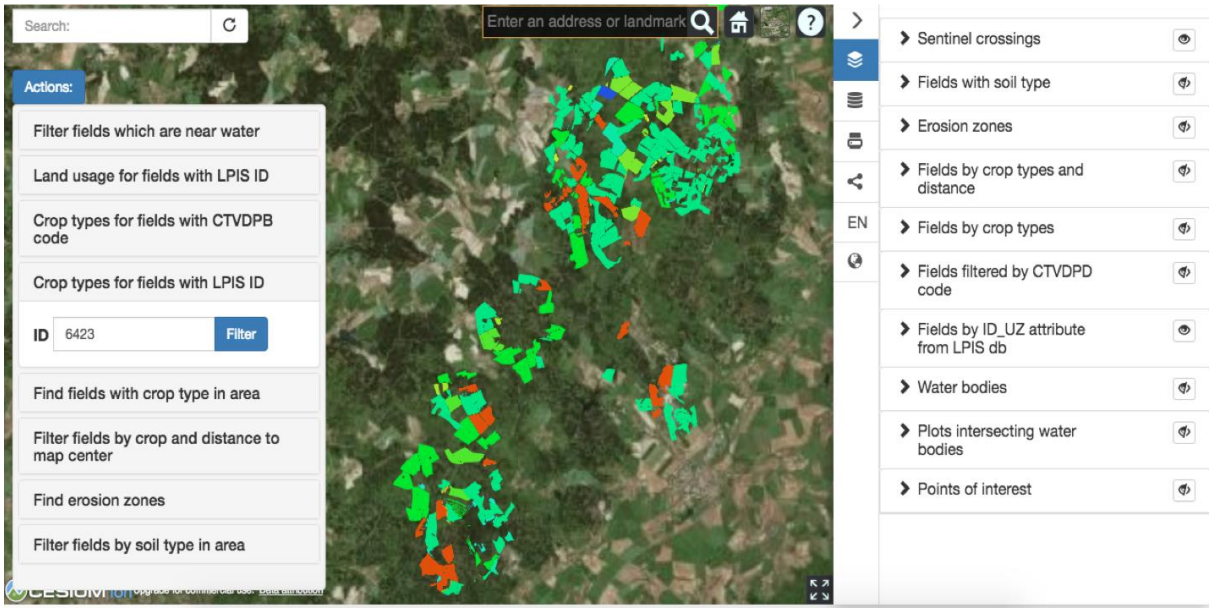
- Server with Rest API: To run the server the database config file is needed as mentioned above, with the command:

  python server.py -d postgresql_config.json

  Then POST requests can be sent to *serverurl:8888/limes* with a Json body that contains the geo-L config.

9.4.1.9 HslayersNG

HSlayersNG[142] is a web mapping library written in JavaScript. It extends OpenLayers 4 functionality and takes basic ideas from the previous HSlayers library, but uses modern JS frameworks instead of ExtSJS 3 at the frontend and provides better adaptability. That's why the NG (Next Generation) is added to its name. HSLayers is open sourced and is built in a modular way which enables the modules to be freely attached and removed as far as the dependencies for each of them are satisfied. The dependency checking is done automatically. In case of the pipelines the tool is mostly used as a visualization tool for exploiting and showcasing the various integration possibilities of the Linked Data from various sources which gave rise to various use cases in many of the above-mentioned projects. To visualize the graphical user interface and explore the Linked Data in a map different application/system prototypes were created for use cases in a few projects using this component as mentioned earlier. One such example can be accessed in the open source version of the application available at https://app.hslayers.org/project-databio/land/.



141 https://github.com/DServSys/geo-L/tree/master/configs

142 https://ng.hslayers.org/

Figure 35: Map visualization prototype (HSLayer application)

9.4.1.10 Metaphactory

This platform is one of the most widely used commercial tools in the pipeline mainly for visualization purposes of the geospatial data. Metaphactory[143] supports knowledge graph management, rapid application development, and end-user oriented interaction. Metaphactory runs on top of your on-premise, cloud, or managed graph database and offers capabilities and features to support the entire lifecycle of dealing with knowledge graphs. Metaphactory's generic approach based on open standards offers great flexibility in different usage scenarios and across various industries and application areas.

Metaphactory also allows unified access over heterogeneous data sources with Query as a Service (QaaS) and the federation engine Ephedra, e.g., define a query spanning several data sources and expose the result through one REST API.

An exemplary Metaphactory instance deployed at PSNC is available at http://metaphactory.foodie-cloud.org/resource/Start.

Metaphactory exposes the following services with associated REST API:

- SPARQL endpoint /sparql (e.g., http://metaphactory.foodie-cloud.org/sparql) that can be used as a Web service, using GET/POST operations to execute SPARQL GET queries
- Query as a Service: physically constructing a SPARQL query string and sending it to the SPARQL endpoint as text can be inconvenient and require much effort from the developers of the client applications. To ease the development of client applications, the platform provides a possibility to define custom REST APIs supported by parameterized SPARQL query templates. The REST API will be exposed under the URL {PLATFORM_URL}/rest/qaas/{id}.

A screenshot of the landing page of the example instance is presented in the Figure 36.

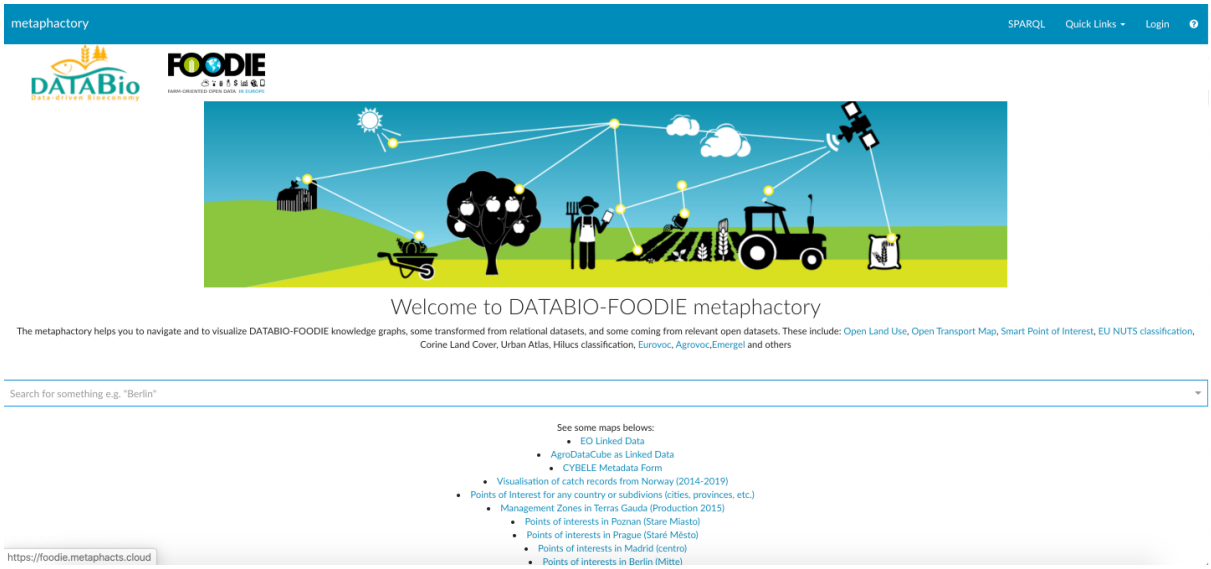


Figure 36: DataBio Metaphactory (entry page)

143 https://www.metaphacts.com/product

9.4.1.11 Ephedra

Ephedra[144] is a component of Metaphactory and an end-to-end Knowledge Graph Platform for knowledge graph management which facilitates rapid application development and end-user oriented interaction. Ephedra allows for processing hybrid queries by providing a flexible mechanism for including hybrid services into a SPARQL federation, and addresses hybrid information needs across multiple dimensions. Ephedra allows accessing hybrid service, like REST API in SPARQL federated queries via API wrappers. Hybrid information needs are captured interactively by the Ephedra engine allowing users to define search clauses, explore partial results, and incrementally add new clauses, while the system provides relevant suggestions. These interactions generate information requests that are expressed as SPARQL queries by the UI components and given to Ephedra to process them. A demo interface has also been implemented to visualize the linked data in Metaphactory (commercial version) having an GUI entry point: http://metaphactory.foodie-cloud.org/resource/:ESA-datasets. The screenshots in Figure 37, Figure 38 and Figure 39 below are some of the examples from this application.

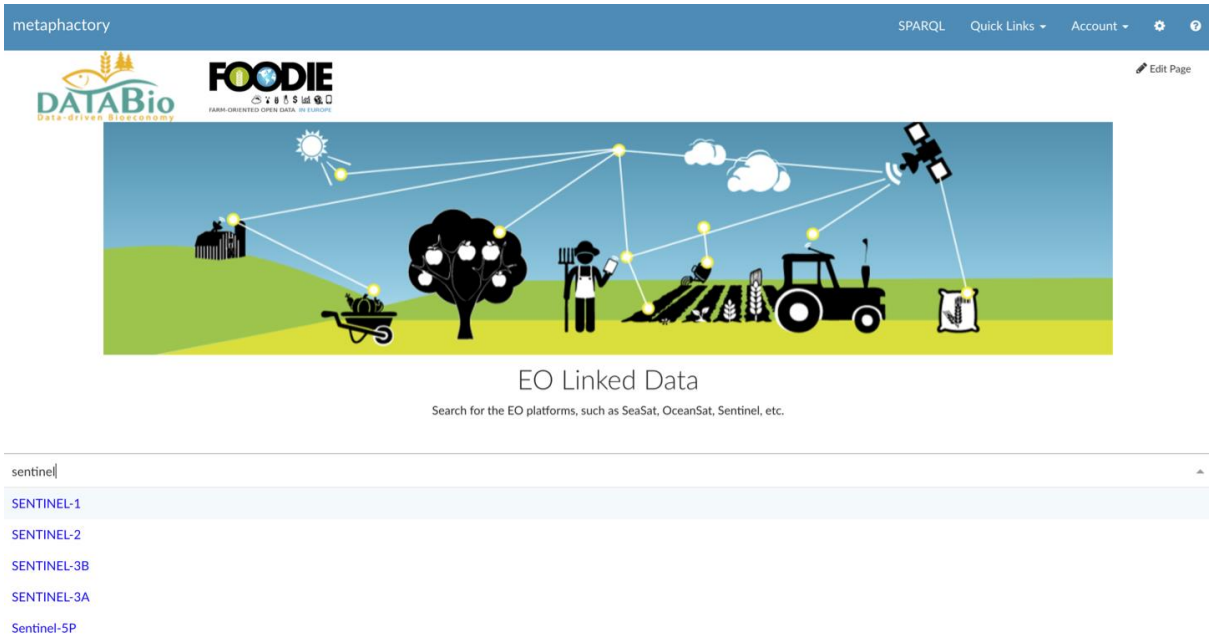


Figure 37: Metaphactory demo application to access FedEO REST API as Linked Data

144 https://help.metaphacts.com/resource/Help:Ephedra

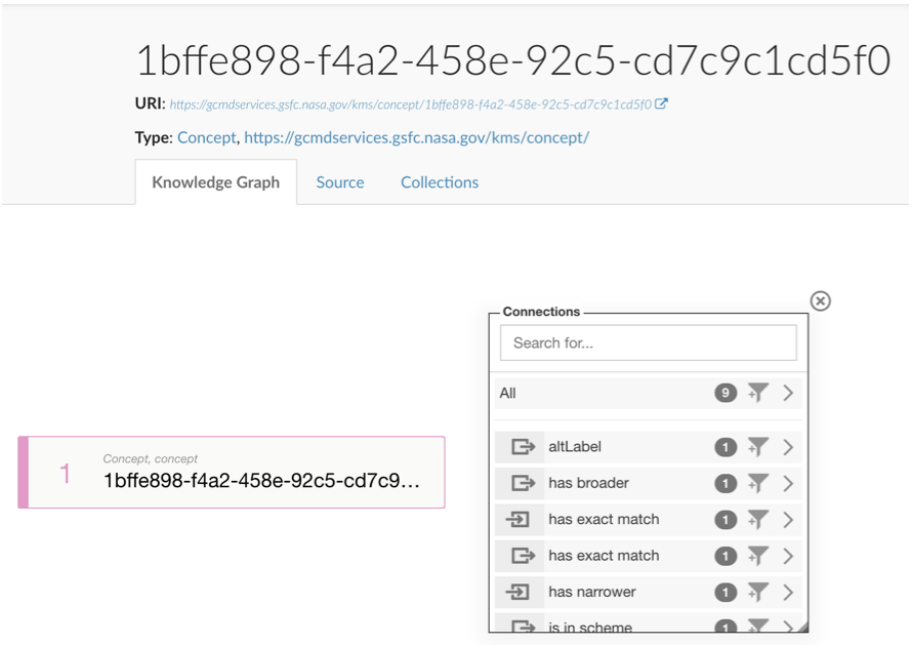


Figure 38: Example of Knowledge Graph visualization of EO platform in Metaphactory

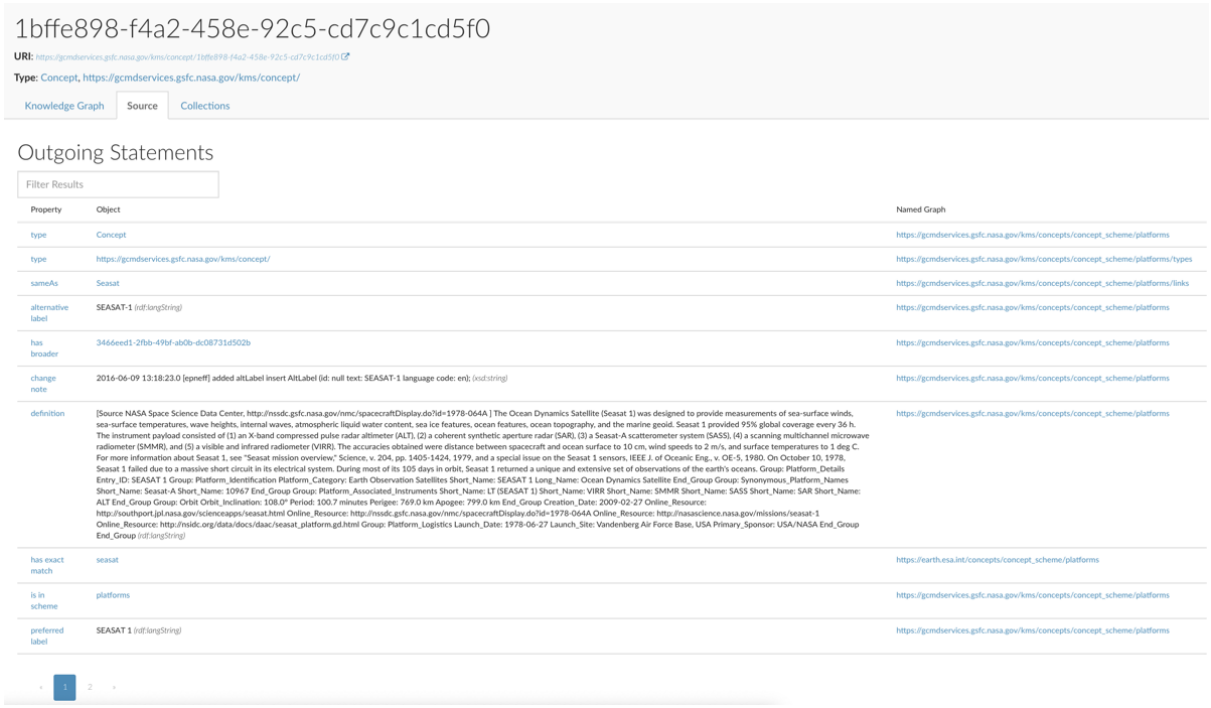


Figure 39: Source visualization of EO platform in Metaphactory

9.4.1.12 Scripts

Additionally, as in many cases it is necessary to carry out some pre and/or post data processing tasks, e.g., to correct data format or to clean data, a set of shell scripts have been created. The scripts are available to reuse as needed in the pipelines. Some of the most generic ones are reused to remove empty/incomplete triples with no values generated due to the absence of values while transformation; post processing scripts for removing duplicate triples and selective removal of the

transformed data etc. Some script is also in Python for the purpose of automation in the pre-processing tasks involved in CSV files transformation. Some exemplary scripts are described below:

- Pre-processing Python scripts to add columns with predefined formulas taking into account the case parameters and in order to make the process automatized. For instance, some of the tasks carried out by the scripts are codifications, adding columns with standard codes associated to string value columns, which will later facilitate integration with existing datasets and vocabularies. The scripts also clean the CSV files with disambiguation, and homogenize the separator character, e.g. removing ";" with "," wherever needed.

```
import pandas as pd
from sys import argv


script, filename1, filename2 =argv

with open (filename1, 'r') as file1, open (filename2, 'w') as file2:
    content1= file1.read()
    if ',' in content1:
        content2=content1.replace(',','')
        content3=content2.replace(';',',')
        file2.write(content3)
    else:
        content2=content1.replace(';',',')
        file2.write(content2)

df = pd.read_csv(filename2)

if 'REGION' in df:
    df.drop('COUNTRY', axis=1, inplace=True)
else:
    pass
```

```
def anc3():
    anc3_code = pd.Series([])
    for i in range(len(df)):
        if df['ANC3'] [i] == 'not in ANC or LFA areas':
            anc3_code[i] = 'ANC3_v3'
        elif df['ANC3'] [i] == 'in mountain ANC or LFA areas':
            anc3_code[i] = 'ANC3_v2'
        elif df['ANC3'] [i] == 'in ANC or LFA areas other than mountain':
            anc3_code[i] = 'ANC3_v1'
        else:
            anc3_code[i] = 'x'
    df.insert(5, 'ANC3CODE', anc3_code)
```

- Post processing script to remove empty/incomplete triples with no values generated due to the absence of values while transformation.

```
#remove empty crops (i.e., no crop in that year)
sed -i '' '/<http:\/\/foodie-cloud.com\/model\/foodie#cropSpecies> <http:\/\/w3id.org\/foodie\/core\/CZpilot_fields\/CropType\/> .$/d' $1
#remove empty crop definition
sed -i '' '/^<http:\/\/w3id.org\/foodie\/core\/CZpilot_fields\/CropType\/> </d' $1
```

- Post processing scripts for removing duplicate triples and selective removal of the transformed data.

```
#remove duplicate lines
sort -u $1 -o $1

#remove 4326 value from aswkt
sed -i '' 's/<http:\/\/www.opengis.net\/def\/crs\/EPSG\/0\/4326> //g' $1


#tranform to latin1 representation of utf-8
native2ascii -encoding UTF-8 -reverse $1 temp.nt

#transform to nice rep
iconv -f UTF8 -t LATIN1 temp.nt > $1

#rem temp
rm temp.nt
```

9.4.1.13 Ontologies and vocabularies

The following ontologies and vocabularies have been used in the past in the existing pipelines; however, as part of DEMETER, few other vocabularies will be used, in particular the DEMETER AIM, which integrates many different major cross-domain and domain specific ontologies and models, like SAREF4Agri[145], FIWARE[146], NGSI-LD[147] etc.

Few of the previously used ontologies and vocabularies have been reused for the implementation of the pipelines, including:

- **FOODIE ontology**[148]: FOODIE ontology is based on INSPIRE schema and the ISO 19100 series standards is mostly suitable in order to represent and model all aspects of the farm and open data from any related input datasets. Its extension includes data elements and relations from the input datasets that were not covered by the main FOODIE ontology [5] but that were critical for the transformation needs. To ensure the maximum degree of data interoperability, the FOODIE data model [6] is based on INSPIRE based generic data models, specially the data models for Agricultural and Aquaculture Facilities (AF), which is being extended and specialized in various projects.
- **SOSA/SSN**[149] : An ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties. A lightweight but self-contained core ontology called (Sensor, Observation, Sample, and Actuator) or SOSA.
- **RDF Data Cube Ontology**[150] : Data Cube Vocabulary and its SDMX ISO standard extensions were effective in aligning multidimensional survey data like in SensLog. The Data Cube includes well known RDF vocabularies (SKOS, SCOVO, VoiD, FOAF, Dublin Core).
- **CatchRecord.owl Ontology**[151]: A design pattern for populating an ontology of aquatic species catching records. With this vocabulary a pattern can be modelled for the kind of species and what amount of organisms have been caught from which areas/countries and at what date and fishing year.

145 https://mariapoveda.github.io/saref-ext/OnToology/SAREF4AGRI/ontology/saref4agri.ttl/documentation/index-en.html
146 https://fiware-datamodels.readthedocs.io/en/latest/Device/Device/doc/spec/index.html
147 https://fiware-datamodels.readthedocs.io/en/latest/ngsi-ld_howto/index.html
148 http://agroportal.lirmm.fr/ontologies/FOODIE
149 https://www.w3.org/TR/vocab-ssn/
150 https://www.w3.org/TR/vocab-data-cube/
151 http://www.ontologydesignpatterns.org/cp/owl/fsdas/catchrecord.owl

9.4.1.14 SANSA Middleware

SANSA[152] is an open source big data engine for scalable processing of large-scale RDF data. SANSA uses Spark and Flink which offer fault-tolerant, highly available and scalable approaches to efficiently process massive sized datasets. SANSA provides the facilities for Semantic data representation, Querying, Inference, and Analytics. The core technology in SANSA-Stack is a processing data flow engine that provides data distribution and fault tolerance for distributed computations over RDF large-scale datasets. SANSA includes several libraries for creating applications:

1. Read / Write RDF / OWL library for RDF/OWL operations,
2. Querying library supports a query language on top of distributed RDF/OWL library, as well as querying heterogeneous non-RDF data.
3. Inference library implements rule-based reasoning on RDF/OWL data,
4. ML- Machine Learning core library

Squerall an integration and extensible framework inside SNSA can be used for querying Data Lakes. It allows ad hoc querying of large and heterogeneous data sources virtually without any data transformation or materialization. It also allows the distributed query execution, in particular the joining of various heterogeneous sources. Squerall also enables users to declare query-time transformations for altering join keys and thus making data joinable and integrates the state-of-the-art Big Data engines Apache Spark and Presto with the semantic technologies RML and FnO. There is a graphical user Interface called Squerall-GUI[153], a solution for querying Data Lakes in a unified manner. Squerall-GUI produces three input files used by Squerall to execute queries:

1. *Config*: It stores information needed to connect to a data source, e.g., host, port, user, password, cluster name, replicas number, etc.
2. *Mappings*: A dedicated database (embedded) storing mappings between data and ontology terms.
3. *Query*: SPARQL query to pass to Squerall for execution.

9.4.2 Data Preparation & Integration Pipeline API:

This API provides a simple yet powerful interface to the functionalities offered by the data preparation & integration facilities. The API facilitates the use and automatization of the underlying components via a homogenous layer, enabling other DEMETER enablers or enhanced entities to launch the whole pipeline of Linked Data Publication and integration, or individual steps. The goal of the API is to abstract the different types of interfaces and details through a simple to use interface. Additionally, the API would facilitate access to integrated data in the data & knowledge repository, represented according to DEMETER AIM. This data will be accessible via SPARQL queries; however, creating a SPARQL query and sending it to the SPARQL endpoint as text may be inconvenient and require much effort from the developers of the client applications. Hence the API will also pre-define access methods leveraging the AIM. An initial set of operations identified for this API, along with a short description, are the following:

- */transform*: In this operation the whole pipeline of transformation into Linked Data can be executed where the user can:
    - provide input datasets and the API will run the pre-processing scripts for the input datasets (if specified)

[152] http://sansa-stack.net/

[153] https://github.com/EIS-Bonn/squerall-gui

  - The API will generate the mapping specifications and/or use the provided one to transform the data into RDF triples using components like geotriples, D2RQ, RML Processor
  - This operation will also store the RDF dumps in the triplestore and also provide options for any post processing if required.
  - The API will also enable the link discovery and thereafter storing/publishing in the triplestore the discovered links.
- */cleanData*: This is an operation in the API that exposes the pre-processing scripts as API mainly in python used for cleaning or modifying CSV files.
- /enrich: This operation in the API will be used to enrich text by exposing the FOODIE annotation service API mentioned above where:
  - the request POST/tagging can create new tagging for the requested list of files and text description. Tagging will consist of a set of tags (annotations) identified by annotation tools. It also creates tagged resource instances. If resource instance URI is provided in the request, the method updates existing tagging.
  - The GET /tagging can retrieve tag labels for the resource and language specified in the request.
- */enrich/resources:* This operation will also expose the FOODIE Semantic annotation service API where a request *GET /tagging will be used to get* resources tagged by tags with specified meaning.
- */generateMapping*: This operation will expose the components used for generation of RML/R2RML mapping specifications with components like D2RQ and GeoTriples via command line interface command to generate mappings.
- */generateDump:* This operation will expose components like Geotriples and RMLProcessor and D2RQ via command line commands for producing RDF dumps from mapping specifications.
- */cleanRDF*: This operation is aimed to expose the post-processing scripts as APIs.
- */link*: This operation will be used to discover links where the API will fetch input dataset source and dataset target and link-spec as xml from the SILK component or minimally source concept and target concept from the source and that will compare primarily for e.g. rdfs:labels and/or skos:prefLabel). This operation will have the ultimate aim to completely expose the SILK REST API.
- */loadRDF*: API operation to load input RDF file in Virtuoso triplestore using conf file.
- */sparql*: This operation will be used to send SPARQL queries and expose virtuoso SPARQL endpoint in the integrated API.
- */access/AIM/instances/{classname}*: This operation will return all the instances of a particular AIM class in format like RDF or JSON/JSON-LD, optionally with some filter parameter from SPARQL queries and/or converted as API in JSON format.
- */access/AIM/entities*: This operation will return all the entities in the triplestore which are within the AIM graph and that include the provided input value in any of their properties, or in a specific property. The SPARQL queries provided are changed as APIs in JSON format.
- */access/AIM/values*: This operation will provide the values of any properties supplied through a query or API expression from the AIM graph.

9.5 Available Linked Datasets

Through the different examples highlighted in the previous sections, multiple RDF datasets have been deployed in the Virtuoso triple store within PSNC, which can be accessed via SPARQL and

faceted search endpoints. Currently the triplestore has over 1 billion triples, being one of the largest semantic repositories related to agriculture, which has been recognized by the EC innovation radar as an agriculture integrator database. The table below shows some of the respective graphs produced by all the pipelines previously described and the number of triples contained in them. The official SPARQL and the faceted search endpoints of the triple store are also mentioned in this section. Below Table 3 is a list of Graphs and the updated number of triples present in the triple store.

Graph URI (note: URIs are not resolvable; they can be used to refer to the specific dataset in the triplestore)	**Name of dataset**	**Number of RDF triples**
http://w3id.org/foodie/open/pl/LPIS/{voivodeship}# (where voivodeship in poland = mazowieckie, dolnoslaskie, kujawsko-pomorskie, lodzkie, lubelskie, lubuskie, malopolskie, opolskie, podkarpackie, podlaskie, pomorskie, slaskie, warminsko-mazurskie, wielkopolskie, zachodniopomorskie, swietokrzyskie)	LPIS Poland	727517039
http://w3id.org/foodie/olu# agriculture related lands (hilucs_code<200) in CZ, PL, ES & for main cities in Czech Republic (centers of NUTS3 regions), Poland (agglomeration areas from Urban Atlas) and Spain (agglomeration areas from Urban Atlas)	Open Land Use	127926060
http://w3id.org/foodie/otm# CZ, ES, PL; but RoadLinks only for FunctionalRoadClassValue of type: (‘mainRoad’, ’firstClass’, ’secondClass’, ’thirdClass’, ’fourthClass’) (see http://opentransportmap.info/OSMtoOTM.html)	Open Transport Map	154340785
http://micka.lesprojekt.cz/catalog/dataset#	Open Land Use Metadata	10456676
http://www.sdi4apps.eu/poi.rdf	Smart Points of Interest (SPOI)	407628622

Graph URI (note: URIs are not resolvable; they can be used to refer to the specific dataset in the triplestore)	Name of dataset	Number of RDF triples
http://w3id.org/foodie/open/cz/pLPIS_180616_WGS#	LPIS Czech Republic	24491282
http://w3id.org/foodie/open/cz/lpis/code/LandUseClassificationValue	LPIS Czech Republic Land Use Classification	83
http://w3id.org/foodie/atlas# agriculture related lands (hilucs_code<200) & for main cities in Czech Republic (centers of NUTS3 regions), Poland (agglomeration areas from Urban Atlas) and Spain (agglomeration areas from Urban Atlas)	Urban atlas	19606088
http://w3id.org/foodie/corine# agriculture related lands (hilucs_code<200) & for main cities in Czech Republic (centers of NUTS3 regions), Poland (agglomeration areas from Urban Atlas) and Spain (agglomeration areas from Urban Atlas)	Corine Land Use	16777595
http://w3id.org/foodie/open/cz/Soil_maps_BPEJ_WGSc#	Czech Soil Maps	8746240
http://w3id.org/foodie/open/cz/water_buffer25#	Czech Water Buffers	3978517
http://w3id.org/foodie/core/cz/Predni_prostredni_vyfiltrovano_UTM#	Yield mass in field crops (CZ Pilot)	1111852
http://w3id.org/foodie/core/cz/Pivovarka_vyfiltrovano#	Yield mass in field crops (CZ Pilot)	437404
http://w3id.org/foodie/core/cz/CZpilot_fields#	CZ pilot fields & crop data	20183
http://ec.europa.eu/agriculture/FADN/{FADN category}# (Where FADN category = year-country, year-country-anc3,	FADN	25629882

Graph URI (note: URIs are not resolvable; they can be used to refer to the specific dataset in the triplestore)	Name of dataset	Number of RDF triples
year-country-lfa, year-country-organic-tf8, year-country-siz6, year-country-siz6-tf14, year-country-siz6-tf8, year-country-sizc, year-country-tf14, year-country-tf8m, year-country-typology, year-region, year-region-siz6, year-region-siz6-tf8, year-region-sizc, year-region-tf14, year-region-tf8)		
http://w3id.org/foodie/open/africa/GRIP#	African Roads Network	27586675
http://w3id.org/foodie/open/africa/water_body#	African water bodies	11330
http://w3id.org/foodie/open/gadm36/{level}# where {level} = level0, level1, level2, level3, level4, level5	GADM dataset	7188715
http://w3id.org/foodie/open/kenya/ke_crops_size#	Kenya crop Size	85971
http://w3id.org/foodie/open/kenya/soil_maps#	Kenya Soil Maps	10168
http://www.fao.org/aims/aos/fi/taxonomic#	FAO	318359
http://www.fao.org/aims/aos/fi/water_FAO_areas#	FAO	150
http://www.fao.org/aims/aos/fi/water_FAO_areas/inland#	FAO	15779
http://www.fao.org/aims/aos/fi/water_FAO_areas/marine#	FAO	6768
http://w3id.org/foodie/open/catchrecord/norway/	Catch Record Norway	192867166
http://standardgraphs.ices.dk/stocks#	ICES stocks data	1270280
https://www.omg.org/spec/LCC/Countries/ISO3166-1-CountryCodes/	ISO Country Codes	8629
https://www.omg.org/spec/LCC/Countries/Regions/ISO3166-2-SubdivisionCodes-NO/	ISO Country Subdivision Codes	391
https://www.omg.org/spec/LCC/Countries/UN-M49-	ISO Region Codes	569

Graph URI (note: URIs are not resolvable; they can be used to refer to the specific dataset in the triplestore)	Name of dataset	Number of RDF triples
RegionCodes/		

Table 3: The number of triples in each graph from the triplestore

The "Generic pipeline for Linked Data" is an example of a pipeline pattern that fits different needs, which can be used in different scenarios, and applied with different data types and sources. Therefore, technically it can be considered a "pipeline design pattern" that can be easily customized to different needs. Linked data is increasingly becoming one of the most popular methods for publishing data on the Web due to several reasons, e.g., improved accessibility, integration, and knowledge discovery. Hence, this pipeline has been created with the aim to collect, transform, and publish data related to DataBio sectors, collected in the form of heterogeneous sources (shapefiles, (Geo)JSON, CSV, relational database, REST APIs) as Linked Data. After the publication of the linked datasets the pipeline includes different methods for their exploitation and reuse over applications like HSLayersNG, Metaphactory, and other Linked Data consumers tools. Ultimately the Linked Data pipeline aims at providing an integrated view over different source datasets, which can be used by different analytic and decision support services to provide better and more informed advice to decision makers.

10 Data quality components

10.1 Overview

There are many definitions of data quality. Data quality is the "degree to which data meets user requirements" (ISO - ISO/TS 8000-1:2011[154] - Data quality — Part 1: Overview 2020). 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).

Thus, generally, data quality is the ability of a given data set to serve an intended purpose, in other terms it fits for its intended uses/purpose. Without high quality data, one cannot operate, make valid decisions or plans. Data quality can only be assessed meaningfully in connection with the planned use and in its context only -not in isolation as an independent concept.

Data Quality plays an integral role in the data analytics pipeline. Before data is fed into downstream analytics and decision support tasks – whether it is model training, prediction, or descriptive analytics – it has to be ensured that incoming data meets general quality criteria, and furthermore that use case specific requirements such as plausible data ranges and distributions are respected.

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

10.2 Quality Requirements

The starting point for a quality assessment is thus the quality needs, which depend, among other things, on the context of the use case and its requirements. Based on the latter, the concept of data quality can be illustrated by a number of quality characteristics (dimensions) of varying importance.

154 https://www.iso.org/standard/50798.html

The ISO defines these characteristics (or aspects) as a "category of data quality attributes that bears on data quality" (ISO - ISO/IEC 25012:2008[155] - Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model 2020). The ISO Standard 25012:2008, in its data quality model, defines the following fifteen quality aspects: Accuracy, Completeness, Consistency, Credibility, Correctness, Accessibility, Compliance, Confidentiality, Efficiency, Precision, Traceability, Understandability, Availability, Portability, and Recoverability.

Having this concept in mind, a top-down approach is suitable and efficient. First, relevant data quality characteristics will be identified and second they will be further elaborated and break-down into single data quality measurements which have to fit to the use case context and needs. Therefore, a good practice is to make use of the Goal Question Metric (GQM) paradigm developed by Basili et al. (1994) [137]. This approach has proven to be systematically expedient and practical as well [138] as it provides a mechanism for defining and interpreting operational, measurable goals. First, according to the GQM template in the following table, the main goals, i.e., the relevant data quality characteristics, are identified and specified, e.g. in an interactive workshop session. For the data quality assessment purpose, the corresponding object is the specific data (of the use case) that should be assessed.

	Schema
Analyze	**<Object>**
for the purpose of	**<Purpose>**
with respect to	**<Quality Focus>**
from the point of view of	**<Viewpoint>**
in the context of	**<Context>**

Table 4: The Goal Question Metric (GQM) abstraction sheet suitable to further refine the goal for the data quality assessment

Secondly, the GQM abstraction sheet (see following Table 5) is used in order to decompose the goals into precise questions to capture the concrete quality focus, in which the use case is interested in. In addition, baseline assumptions, variation factors and the impacts of these variation factors are collected and defined.

155 https://www.iso.org/standard/35736.html

Goal:	Object	Purpose	Quality Focus	Viewpoint	Context
	...	...	...	...	...

Quality Focus	Variation Factors
...	...
(Baseline) Hypotheses	**Impact of Variation Factors**
...	...

Table 5: The Goal Question Metric (GQM) abstraction sheet suitable to further refine the goal for the data quality assessment

Based on all this information, the data quality requirements are refined in a detailed representation that allows further definition of corresponding metrics, i.e., data quality measures, to assess the relevant data quality. This procedure is visualized in the following Figure 40 and can be used to start the quality assessment for agricultural data.

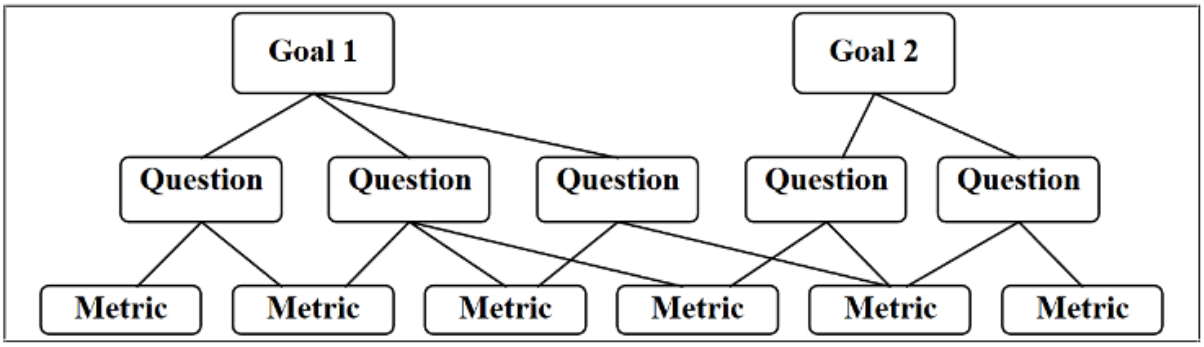


Figure 40: Structure of the Goal Question Metric (GQM) approach by Basili et al. [1994] suitable for the data quality assessment

10.3 Data Quality Assessment

The quantification of a quality characteristics is done by measures that quantify certain properties of a data construct, i.e., data quality measures are defined to measure and assess the specified data quality requirements focusing on data quality characteristics. For example, the data quality characteristic Completeness can be measured with a metric ratio of null values that calculates the ratio of the number of null values within a dataset. Furthermore, those data quality metrics can then be instantiated to perform the real measurement, e.g., by coding a python script for a specific data set.

Having these measured values for different quality characteristics supports the determination of an overall data quality assessment. Therefore, a concrete model should be defined that specify what should be considered concretely and how the single values should be aggregated.

In addition, in order to support a comprehensive quality assessment, rules for combining the partial assessments at the quality aspect level should be available to form a global assessment. For example, it consists of weighting the individual quality aspects according to their importance (depending on the underlying quality needs) and appropriate balancing rules [139]. Indeed, based on the specific information needs of the user, certain data quality characteristics and data quality measures may appear more important than others. The following Figure 41 provides an example.

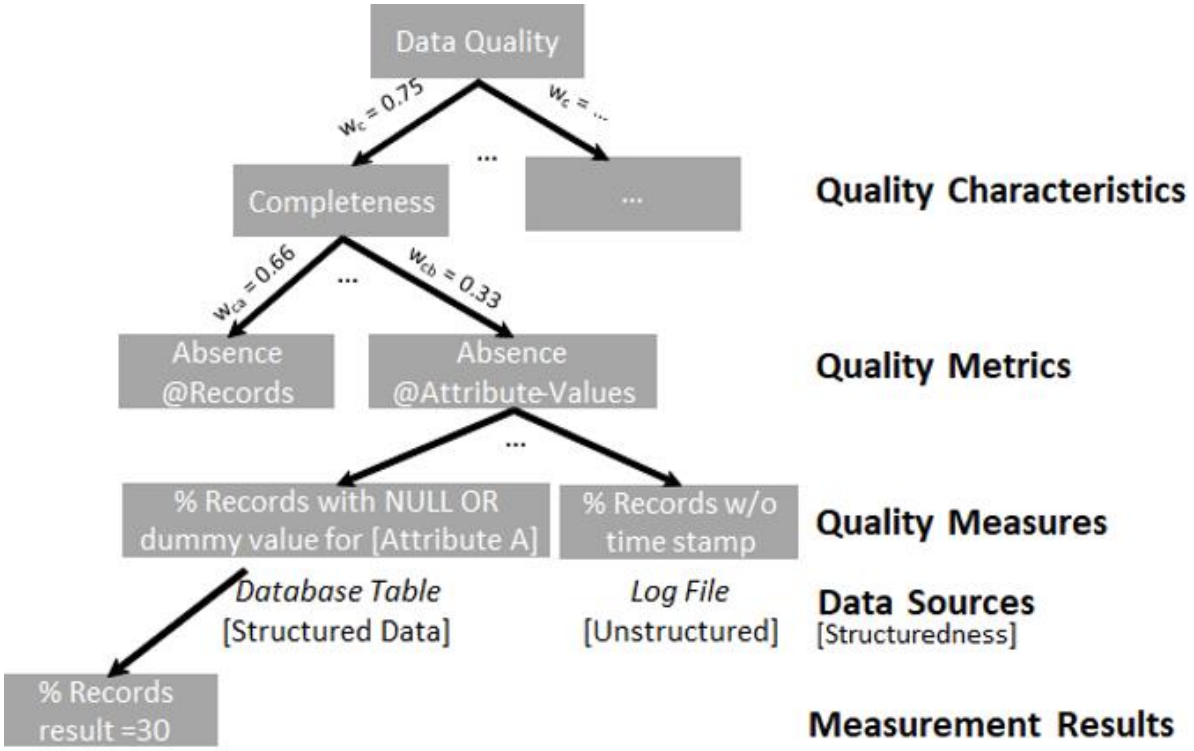


Figure 41: An example of a data quality model including weights

Overall, the approach of data quality assessment in the context of the DEMETER project with agricultural data will combine the different components including data quality needs and requirements, data quality characteristics and measures. The following Figure 42 visualizes this general iterative flow.

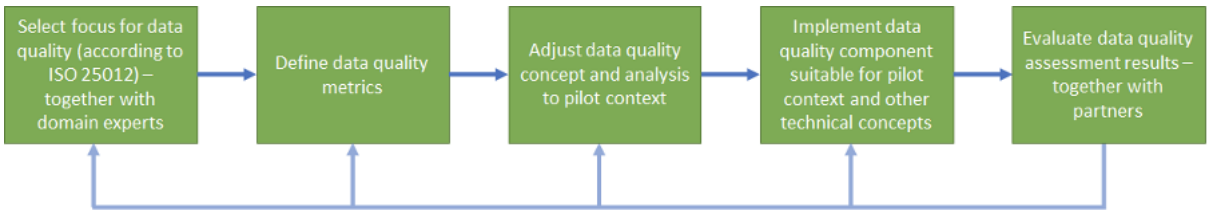


Figure 42: Generic data quality assessment approach

10.3.1 Data Quality Assessment API

From an implementation perspective, the data quality assessment will be realized as depicted in the following Figure 43 . A Data Quality API offers an interface to run a data quality assessment, for a given Dataset and given QualityConstraints. The call then will be redirected to the general Data Quality Assessment module, which will again redirect the call to either SANSA (for linked data) or

deequ (for tabular data). The user can additionally register Specialized Data Quality Assessment Components for particular tabular data sets, containing completely custom quality assessment methods. The assessment result of each component will be returned as DQV-conform (Data Quality Vocabulary) data.

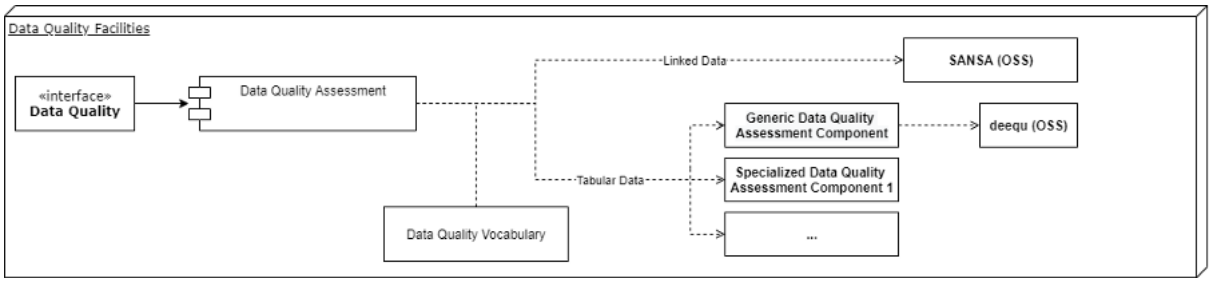


Figure 43: Generic data quality assessment approach

The following Figure 44 illustrates the processing sequence of a call to the Data Quality Assessment API and the related flow of data. A registry for specialized data quality components is in place, but for the sake of simplicity not covered in the diagrams. The registry will consist of a list of mappings from DataResource identifiers (see Metadata Schema in D2.1) to concrete API-Calls of the Specialized Data Quality Assessment components. Each component has the responsibility to handle constraints in form of QualityConstraints objects and return the assessment results as DQV-compliant dataset. It is recommended that specialized components are provided as dockerized solutions to ensure compatibility across different platforms and runtime environments.

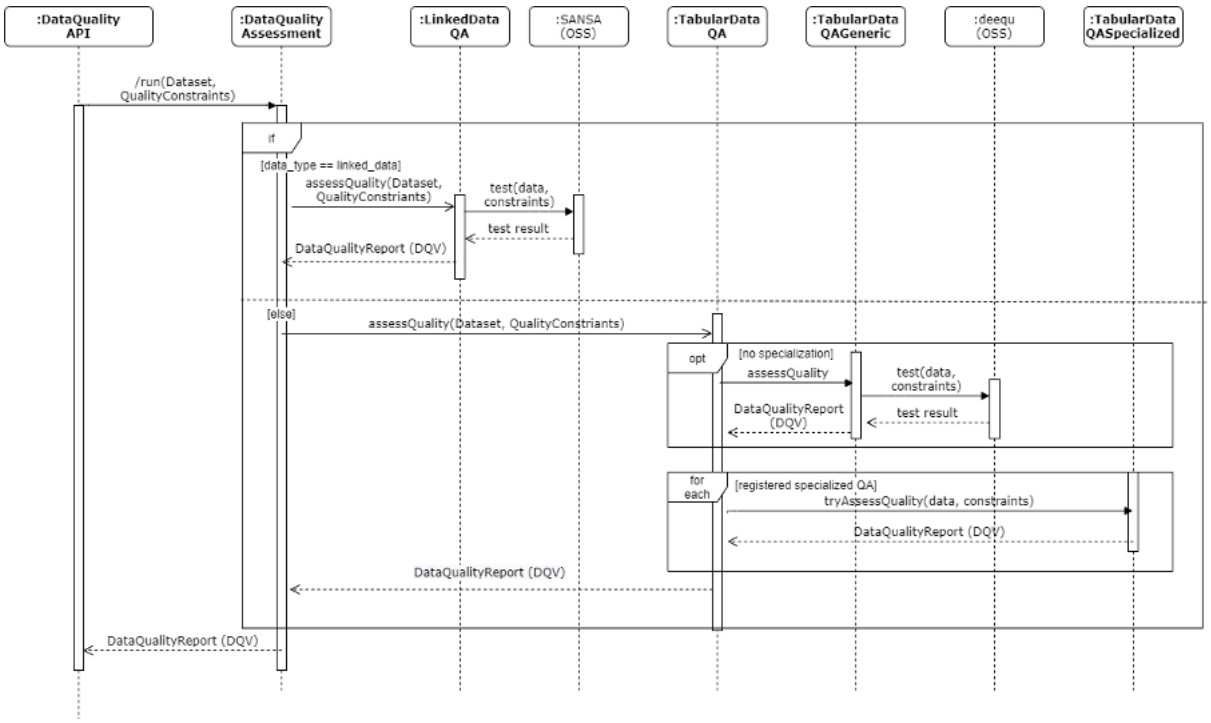


Figure 44: Sequence diagram for Data Quality Assessment API

The Data Quality API offers the following REST-based interface:

Calls

- **/run**: this API call will perform a data quality assessment based on a Dataset object and QualityConstraints. Returns a DataQualityReport.
- **/run/linkedData**: direct call for data quality assessment of linked data (e.g. RDF). Requires a Dataset object and QualityConstraintsLD. Returns a DataQualityReport.
- **/run/tabular**: direct call for data quality assessment of tabular data (e.g. CSV). Requires a Dataset object and QualityConstraintsTabular. Returns a DataQualityReport. A generic QA method will be used if none of the registered targeted modules returns a valid DataQualityReport.
- **/run/tabular/generic**: direct call for data quality assessment of tabular data (e.g. CSV) using a generic assessment method. Requires a Dataset object and a QualityConstraintsTabular. Returns a DataQualityReport.
- **/run/tabular/targeted**: direct call for data quality assessment of tabular data (e.g. CSV) using a custom QA method. Requires the method name, a Dataset object and a QualityConstraintsTabular object. Returns a DataQualityReport.
- **/run/tabular/targeted/register**: register a custom quality assessment module for tabular data based on a TargetedQASpecification.
- **/run/tabular/targeted/list**: list all registered modules by name.
- **/run/tabular/targeted/delete**: delete a registered module based on the module name.

Objects

- **Dataset**: An object, identifying a particular dataset. The concrete attributes are still to be defined in accordance with the final data management implementation and the AIM metadata schema. Generally, it should contain a URI, uniquely identifying a dataset.
- **QualityConstraints**: An object, listing ISO Standard 25012:2008 quality characteristics and corresponding thresholds to check compliance with. A more fine-grained definition of constraints on metric- or attribute-level is also possible. This object is a generalization of QualityConstraintsLD and QualityConstraintsTabular, which only contain quality characteristics relevant for either Linked Data or Tabular Data.
- **QualityConstraintsLD**: Quality constraints as used by SANSA.
- **QualityConstraintsTabular**: Quality constraints as used by deequ.
- **SpecializedQASpecification**: An object containing the quality assessment API URI of a specialized Quality Assessment component and the DataResouce Identifier of data sets it is responsible for.
- **DataQualityReport**: The result of a data quality assessment as JSON-LD, in compliance with the Data Quality Vocabulary DQV.

10.3.2 Quality Assessment of Linked Data

The SANSA-Stack[156] offers a scalable solution for semantic analytics and, amongst others, a solution for scalable data quality assessment of linked data [102]. Other solutions, such as Luzzu[157] and Sieve[158] were considered but found to be less complete or less suitable for ISO 25012-compliant quality assessment, respectively. SANSA covers the following quality criteria within Table 6 :

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			

Table 6: SANSA quality criteria

Most of the used metrics are generic in the sense that they do not require use-case specific parameterization. An exception are literal numeric range checks, which require a user-provided upper and lower bound. Those will be extracted from the QualityConstraints provided by the caller of the Data Quality API or extracted from Metadata information (valid value ranges) provided with the Dataset.

To use the SANSA open-source component, available at: https://github.com/SANSA-Stack/SANSA-RDF, a wrapper module will be implemented. It will wrap the Scala source code of SANSA into a web-service, offering the required functions stated in Sections 9.3.1. SANSA, by default, provides data quality assessment results in a DQV-compliant format, so the main effort for the wrapper will consist of mapping the DEMETER QualityConstrains to meaningful inputs for SANSA. An unparameterized example implementation of a SANSA-based quality assessment reads as follows in Figure 45:

156 http://sansa-stack.net/

157 https://github.com/Luzzu/Framework

158 https://sieve.wbsg.de/

```
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()
val availability_DereferenceableUris = triples.assessDereferenceableUris()
val relevancy_CoverageDetail = triples.assessCoverageDetail()
```

Figure 45: SANSA-based quality assessment

Another important aspect is that SANSA is based on Spark and hence a Spark environment is required for running the analysis. It is still to be defined whether the Spark support will be realized within the DEMETER Data Management component or as part of the SANSA-wrapper. If this remains part of the SANSA-wrapper, a Docker-based Spark Standalone distribution will be used and data will be transferred from the DEMETER Data & Knowledge Repository to the Spark instance on demand.

10.3.3 Quality Assessment of Tabular Data

10.3.3.1 Generic Approach

For data quality assessment of tabular data, deequ[159] [146] offers a scalable open-source solution. It is suitable to determine general quality metrics about missing values but also to determine more specific metrics and meta-metrics aligned with the data quality categories described in Section 10.2.

Completeness	**Description**
Completeness	fraction of non-missing values in a column
Consistency	**Description**
Size	number of records
Compliance	ratio of columns matching predicate
Uniqueness	unique value ratio in a column

159 https://github.com/awslabs/deequ

Distinctness	unique row ratio in a column
ValueRange	value range verification for a column
DataType	data type inference for a column
Predictability	predictability of values in a column
Statistics (Dimension Consistency)	**Description**
Minimum	minimal value in a column
Maximum	maximal value in a column
Mean	mean value in a column
StandardDeviation	standard deviation of the value distribution in a column
CountDistinct	number of distinct values in a column
ApproxCountDistinct	number of distinct values in a column estimated by a hyperloglog sketch
ApproxQuantile	approximate quantile of the value in a column
Correlation	correlation between two columns
Entropy	entropy of the value distribution in a column
Histogram	histogram of an optionally binned column
MutualInformation	mutual information between two columns

The quality assessment is defined in the form of unit tests, as shown in the following example in Figure 46:

```
import com.amazon.deequ.VerificationSuite
import com.amazon.deequ.checks.{Check, CheckLevel, CheckStatus}


val verificationResult = VerificationSuite()
  .onData(data)
  .addCheck(
    Check(CheckLevel.Error, "unit testing my data")
      .hasSize(_ == 5) // we expect 5 rows
      .isComplete("id") // should never be NULL
      .isUnique("id") // should not contain duplicates
      .isComplete("productName") // should never be NULL
      // should only contain the values "high" and "low"
      .isContainedIn("priority", Array("high", "low"))
      .isNonNegative("numViews") // should not contain negative values
      // at least half of the descriptions should contain a url
      .containsURL("description", _ >= 0.5)
      // half of the items should have less than 10 views
      .hasApproxQuantile("numViews", 0.5, _ <= 10))
    .run()
```

Figure 46: Unit test based Quality Assessment in SANSA

Similar to the integration of SANSA, the main challenge for the integration of deequ with DEMETER is the translation of DEMETER QualityRequirements to deequ unit-tests as well as the establishment of an Apache Spark environment. Additionally, test results of deequ, have to be mapped to DQV-conform quality information. The DEMETER Generic Quality Assessment Module for Tabular Data will contain implementations of the required mappings and provide (a connection to) a Spark environment.

10.3.3.2 Specialized Approaches

To enable specialized, use-case specific, quality assessment of data, the Quality Assessment component allows for the registration of specialized quality assessment modules, which have to comply to the API as defined in Section 10.3.1, but have no further constraints apart from that. Modules can be registered, using a *SpecialiedQASpecification* object.

Pilot 2.2

For the technical realization, we will consider the DEMETER WP3 guidelines that are defined in parallel to the activities of WP2. Moreover, our developments are based on existing data quality standards (mainly ISO 25012 and 8000), which we will use to implement the mentioned data quality concepts described above. As a third big driver for our implementations, we will consider the technical situation and context of agricultural stakeholders involved in the DEMETER project and their constraints. This will be used to implement suitable data quality metrics and measures. For enabling a broader use of this solution, the implementation should ideally be compatible with existing technologies in the agricultural domain such as the COGNAC platform (COGNAC – Cognitive Agriculture 2020)[160].

160 https://www.fraunhofer.de/en/research/lighthouse-projects-fraunhofer-initiatives/fraunhofer-lighthouse-projects/cognac.html

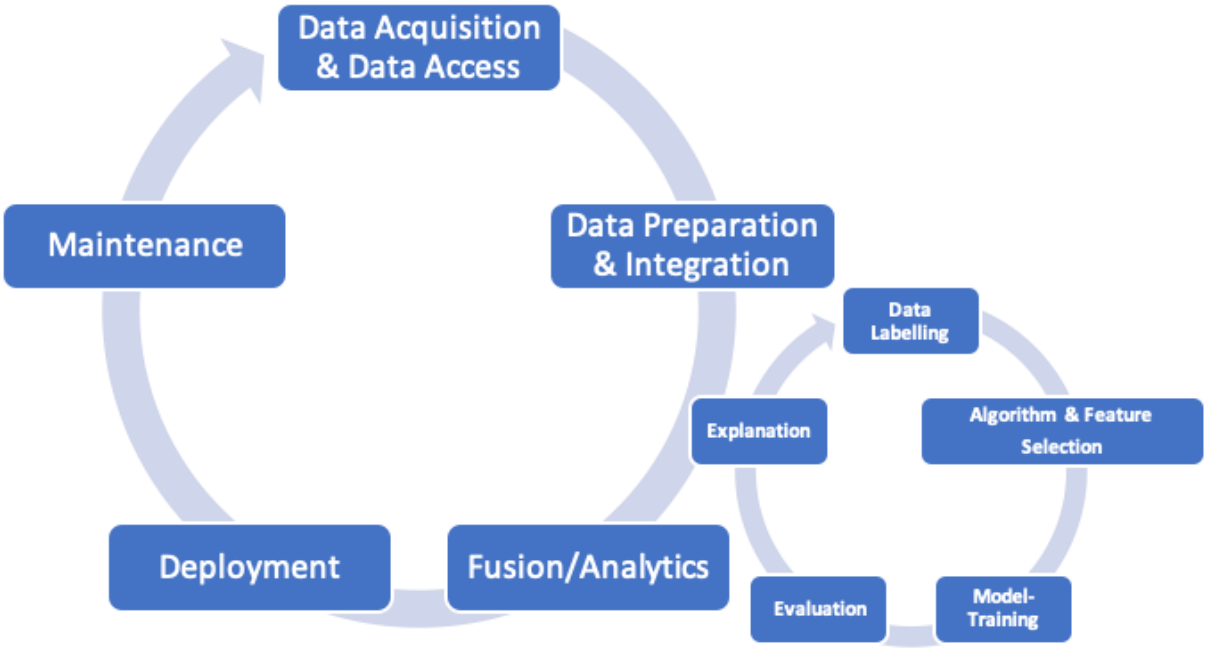
10.4 Data Provenance and Metadata

The Data Quality Vocabulary (DQV) describes data quality characteristics as defined in the ISO 25012 Standard. DQV is part of the AIM Metadata model as described in DEMETER Deliverable D2.1. The output of all quality assessment modules will be in the form of DQV-compatible metadata. Data provenance information is also captured in the Metadata information of AIM following the PROV-O standard[161]. Data Quality information can hence be used along with provenance information to, e.g., identify quality issues at particular data providers and help address them at the source of origin.

11 Data analytics and fusion components

11.1 Introduction

This section presents the data analytics and data fusion components within the DEMETER Data Analytics and Knowledge Extraction Enabler. The differentiation between analytics and fusion tasks is that analytics tasks extract knowledge from data sources, while fusion tasks extract features from data sources. Analytics and fusion tasks within this enabler are targeted and use-case specific. Every module thus addresses one or multiple requirements of DEMETER pilots. Section 11.2 describes modules responsible for data analytics tasks while Section 11.3 describes targeted data fusion tasks. The remaining sections outline the general machine learning (ML) components that can be utilized by all analytics and fusion modules. The ML components aim at centralizing recurring tasks of analytics and fusion modules within the scope of the DEMETER Data Analytics Lifecycle. The lifecycle covers various aspects of machine learning-based analytics solutions and can be regarded as an extension of the CRISP-DM [145] industry-standard as depicted in Figure 47 below. Data Acquisition & Access, as well as Data Preparation and Integration tasks, are supported by the DEMETER Data Management, and Data Preparation & Integration Enabler. The Data Analytics and Knowledge Extraction Enabler takes responsibility for the remaining aspects of the lifecycle and aims at offering those capabilities to support upstream Decision Support and Benchmarking as well.



161 http://www.w3.org/TR/2013/REC-prov-o-20130430/

Figure 47: The DEMETER Analytics Lifecycle

Figure 48 below provides an overview of all DEMETER Data Analytics and Knowledge Extraction facilities. Each fusion and analytics module implements a specific knowledge specific task and adheres to a common interface. A Data Analysis / Data Fusion Template Module (i.e. a prepared docker container) provides the respective interface and serves as a starting point for all implementation. For analytics tasks, an additional template specialized for computer vision tasks is provided. Each of the modules can make use of Machine learning facilities and other Analytics or Fusion modules provided within the enabler.

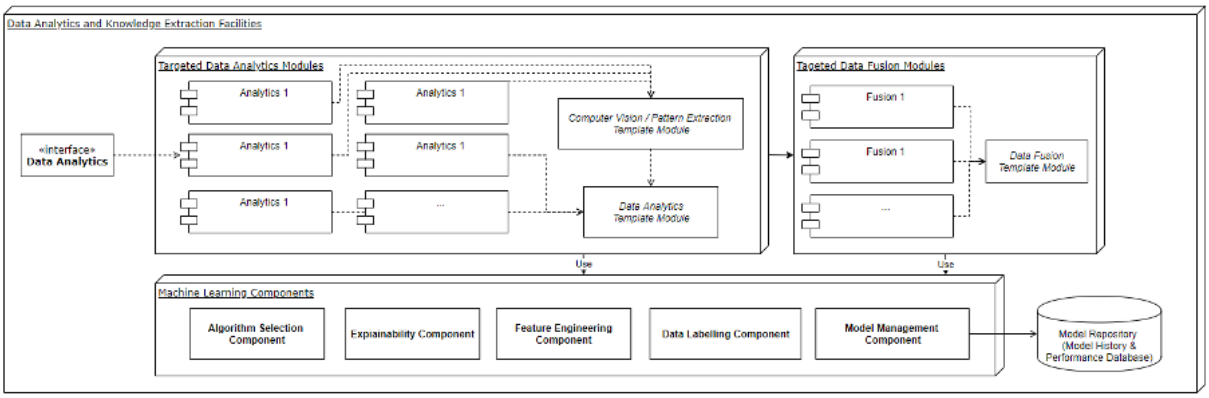


Figure 48: The architecture of Data Analytics and Knowledge Extraction enabler

The implementation of the data analytics & fusion components can be found at the WP2 Data Analytics and Data Fusion folders, respectively in DEMETER GitLab: https://gitlab.com/demeterproject/wp2/dataanalytics & https://gitlab.com/demeterproject/wp2/datafusion

11.2 Data Analytics and Fusion API

The Data Analytics (and Fusion) API will be implemented as REST-based Webservice using FastAPI[162] , which is a high-performance web framework for building API for Python. The solutions will be deployed as a dockerized container.

Calls

- **/runAll**: this API call will start the knowledge extraction process of each registered (analytics and/or fusion) module and returns a ResultSummarySet.
- **/modules/register**: register a module based on a ModuleSpecification
- **/modules/list**: list all registered modules by name
- **/modules/delete**: delete a registered module based on the modules name

Objects

- **ModuleSpecification**: An object, containing the name and API URL of the module being registered
- **ResultSummarySet**: An object, containing a set of ResourceIdentifiers of the extracted knowledge (analytics) or extracted features (fusion) and status information about the performed tasks.

162 https://fastapi.tiangolo.com/

Analytics and Fusion Module API

Calls

- **/run**: this API call will start the knowledge extraction process and return a ResultSummary object
- **/model/retrain**: if the module is machine learning based, the internal models will be retrained, based on a RetrainingReceipe object
- **/model/revert**: if the module is machine learning based, the model can be reverted to a past state, given a ModelIdentifier object

Objects

- **ResultSummary**: An object, containing the ResourceIdentifier of the extracted knowledge (analytics) or extracted features (fusion) and status information about the performed task.
- **RetrainingReceipe**: An object, containing ResouceIdentifier of training data and optional parameters for the training task.
- **ModelIdentifier**: An URI, which uniquely identifies a particular machine learning model.

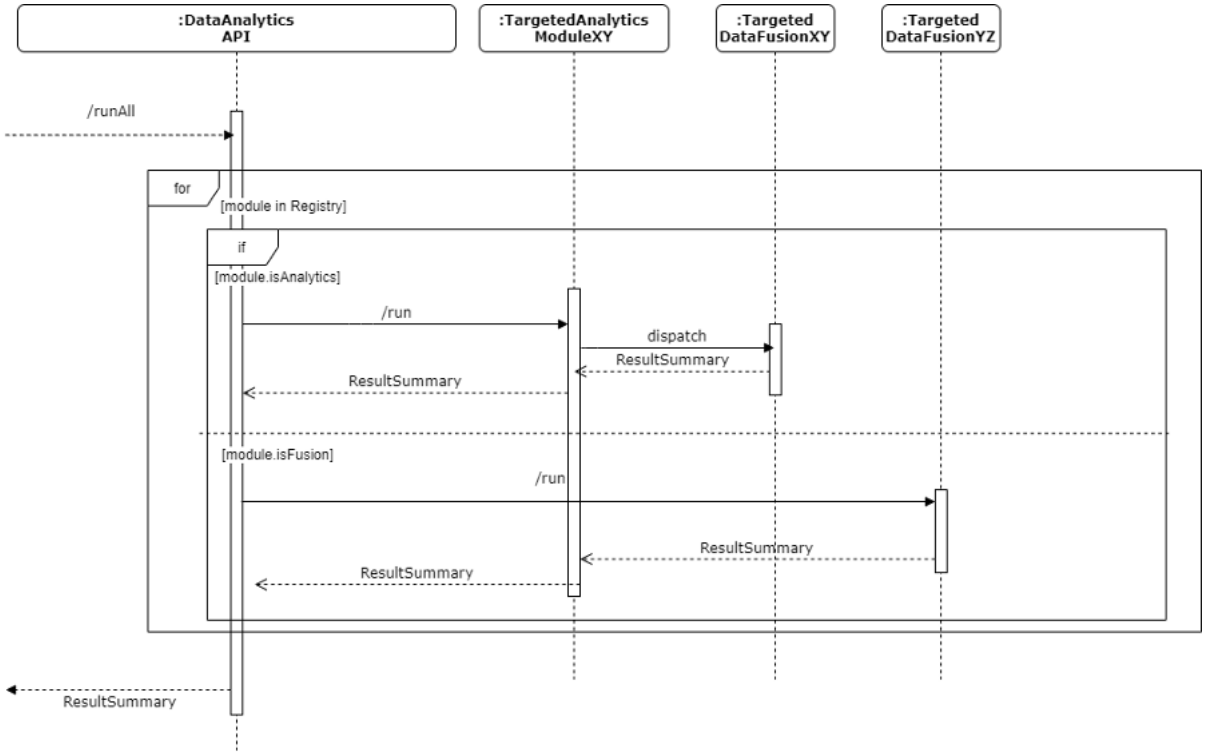


Figure 49: The sequence diagram for Data Analytics service

11.3 Targeted Data Analytics Modules

In the following, a few initial analytics modules are proposed, which shall illustrate the nature of DEMETER analytics tasks that aim at extracting new facts from existing data, by employing primarily machine learning and deep learning methods. In later stages of the project, more analytics/knowledge extraction tasks are aimed to be implemented as modules, to make them more accessible for other users of the DEMETER Enabler Hub.

11.3.1 General Approach for Pattern Extraction with Computer Vision

Many DEMETER pilots make use of imagery data for the identification of visual elements through pattern extraction. This component aims at the identification of patterns in the images to provide some indicators of the appearance of certain elements in the collected images (e.g. counting flies, identifying some kind of parasites like varroa mites, etc.). To do that, the component developed to that end has to be capable of generating prediction models from sets of images properly labelled. It might provide the capability of tuning some model generation parameters to provide a better prediction (e.g. some kernels, evaluation criteria, etc.). Once the model has been generated, test cases (i.e. images taken from the pilot data sources) can be used as input to obtain the proper expected prediction. Additionally, some preprocessing can be carried out before the model generation/usage, to prepare the images for a better prediction, by reducing noise or boosting characteristics of the image that might drive to better results. The Figure 50 below depicts the overall process required for computer vision pattern extraction tasks.

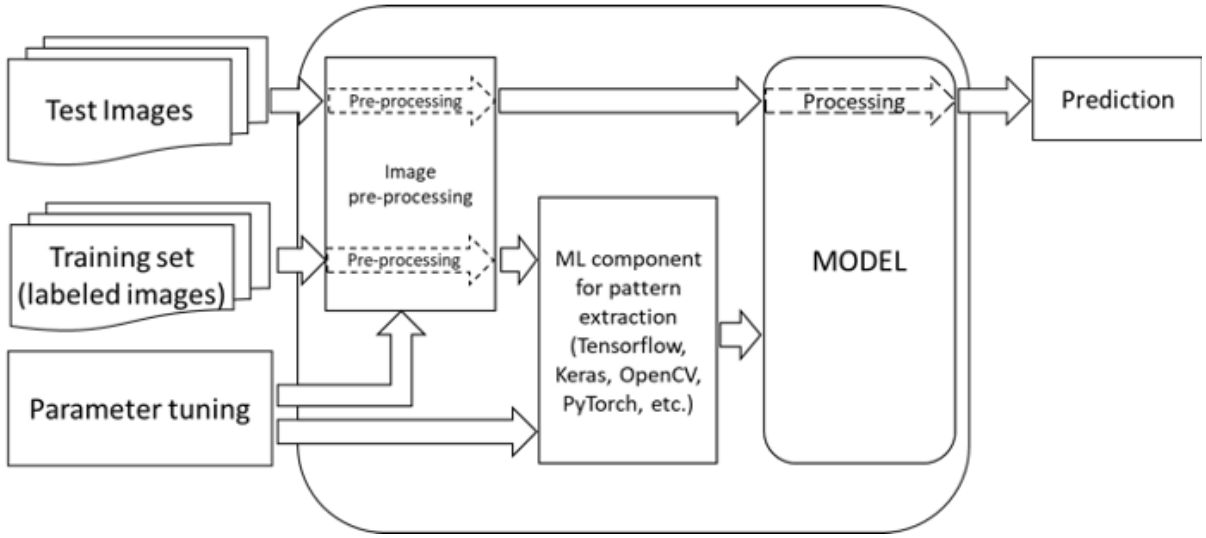


Figure 50: Computer vision knowledge extraction task

The Figure 51 below provides an architectural perspective to the implementation and also indicates which generic ML components are employed, while **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** illustrates the illustrates the sequence diagram for the three main use cases: model setup, model training and model prediction.

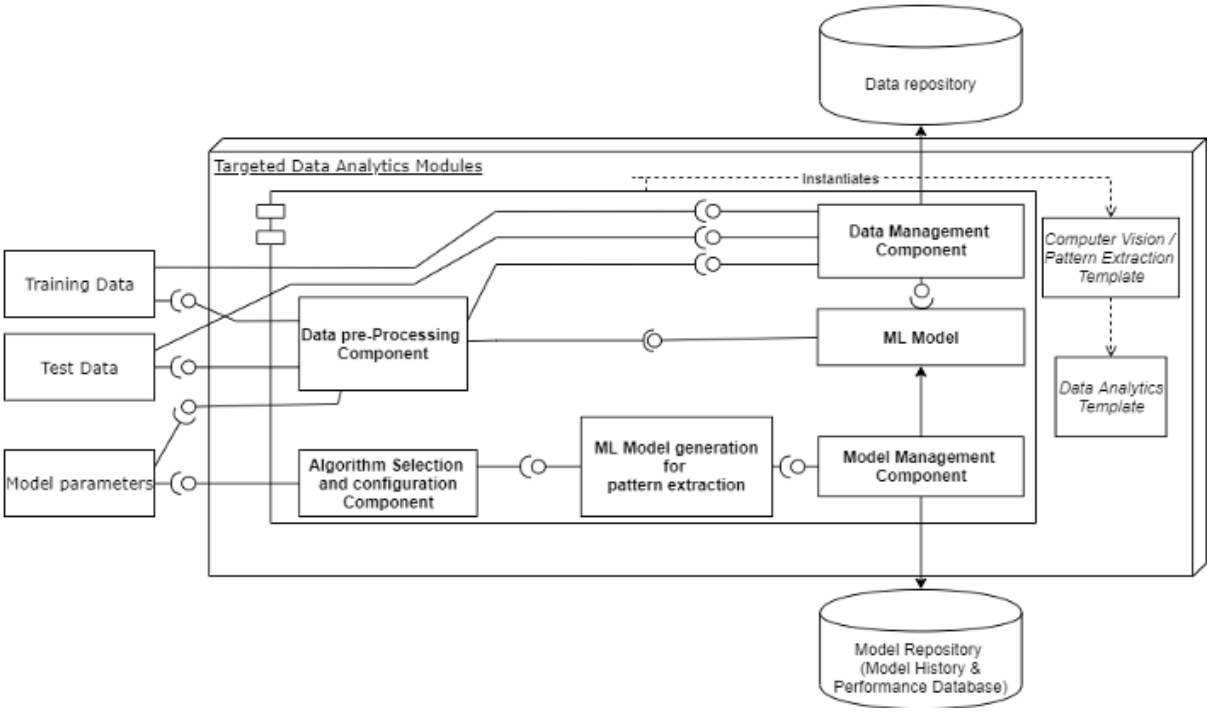


Figure 51: Computer vision knowledge extraction task (component diagram)

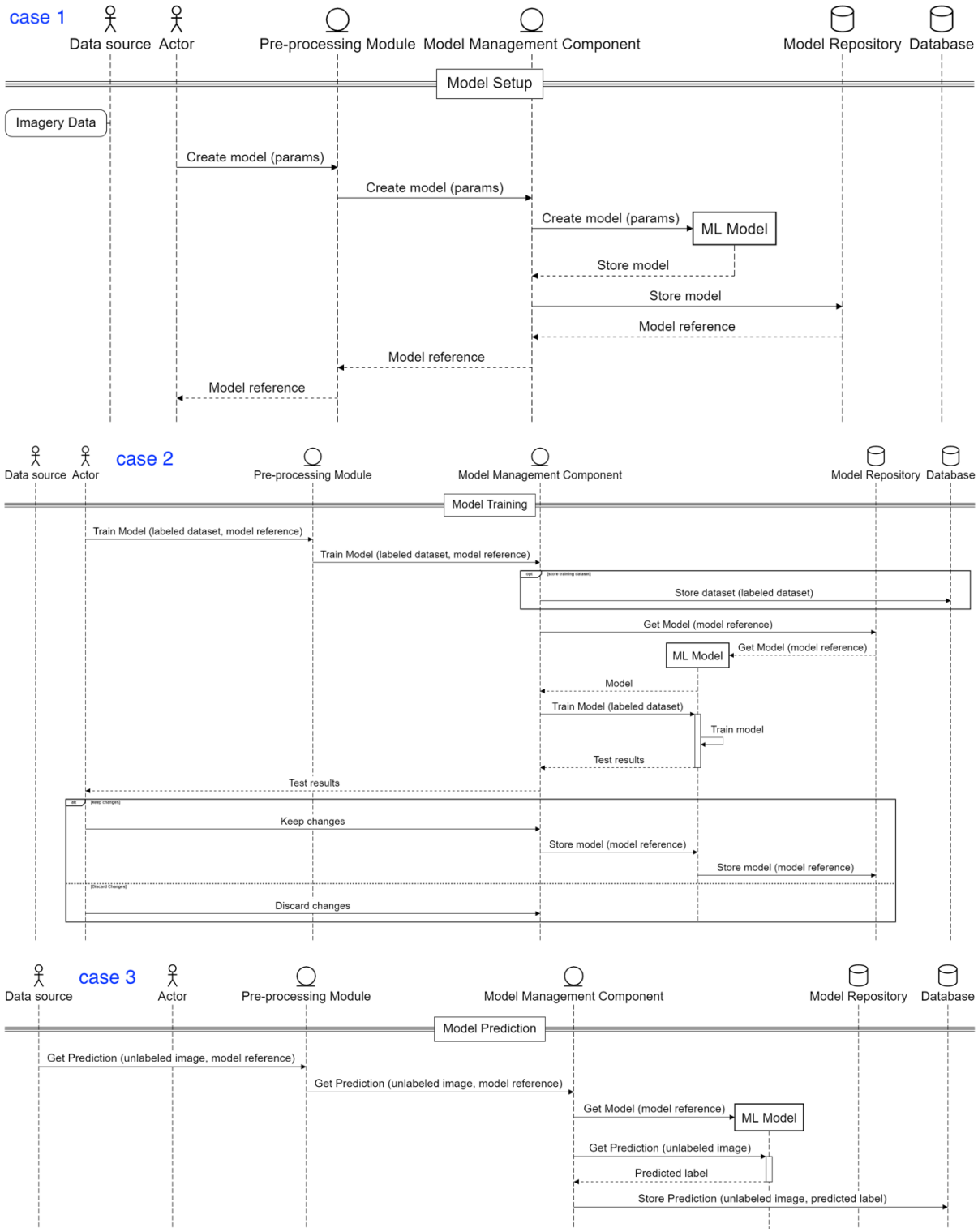


Figure 52 Computer vision knowledge extraction task. Sequence diagram of three main use cases: Model setup, Model training and model prediction

A generic implementation of the describes module will serve as a starting point for all computer vision knowledge extraction tasks within the DEMETER Knowledge Extraction Enabler.

11.3.2 Pattern Extraction for Fruit Fly Counting

The counting of fruit flies based on imagery is a targeted knowledge extraction task, which will instantiate the previously described computer vision template. Although specific details of the task and implementation are still to be defined in correspondence with the respective DEMETER Pilot partners, some aspects regarding this solution can be estimated. It is expected to process images with different lighting conditions, which might drive this use case to generate different models in order to evaluate the images taken with similar lighting conditions (UV lighting and regular lighting). The development of the component in this case is highly dependent on the availability of a proper dataset captured with the traps that will be used in the pilot (in order to train the detection models with images that show the same conditions to the ones of the pilot).

11.3.3 Pattern Extraction for Optimal Fertilizer Usage

Fertilizer use efficiency is one of the biggest challenges that farmers face. Advanced analytics can assist this task by indicating both the right amount and the optimal time to apply fertilization.

This component aims to utilize mostly spectral imagery collected from UAVs or even satellites. Environmental data such as precipitation and nutrient concentration in water could also serve as input for the module.

UAVs have the ability to extract more information than plain image, so it is also possible for us to acquire valuable data, concerning a variety of plant traits like biomass, chlorophyll levels, Nitrogen uptake capability and actual uptake. Specifically, imagery is logged properly and several spectral indices are being computed in the process. These data are 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 the data can also be fed to the model in the form of time series. Furthermore, imagery from additional sources (e.g. satellites) may be fused with the aforementioned data to reduce possible noise. Calculated use efficiency during the past few years shall also be used to empower the model and increase its accuracy, but also as test cases after the model is generated.

The primary output of the module should be the status of Nitrogen in the field and the crop. Nitrogen is the Nitrogen is by far the most used ingredient as fertilizer and it is actually the only one used before sowing. This stage is the one that currently depends solely on human experience and requires decision support by our components. Complementary outputs like Nitrogen concentration and uptake of paddy rice could be fed to empower the prediction and enhance the decision support system for Pilot 1.3 and possibly others too.

Some of the most popular solutions that may be used by DEMETER to perform data analytics and machine learning are the following:

- TensorFlow is an open-source symbolic math library, using data flow graphs to build models. Created by Google, it allows developers and researchers to create large-scale multi-layer neural networks for a variety of machine and deep learning tasks. Keras is the high-level API that runs mainly on top of TensorFlow, but also other backends like Theano or mxnet. Apart from being just an API, the framework provides great and fast support for small-scale experiments on data.
- Apache Spark is an open-source distributed general-purpose cluster-computing framework. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. MLlib is Spark's scalable machine learning library, that works well with a number of languages, having APIs in Java, Scala, Python, and R.

- Scikit-learn or sklearn is a free software machine learning library for the Python programming language. It features various algorithms for all three major machine learning tasks: classification, regression and clustering and is designed to interoperate with the Python libraries NumPy and SciPy, who best perform efficient linear algebra operations on multi-dimensional data and scientific computing, respectively. The model is planned to take advantage of algorithms like multivariate regression or (random) decision forests to enhance decision support on fertilization.

Finally, Recurrent Neural Networks can be a reliable solution for predictions, since they can model sequences of data in a way that takes advantage of the dependence between a sample and the previous ones. Also, convolutional layers may be added to extend the effective pixel neighborhood.

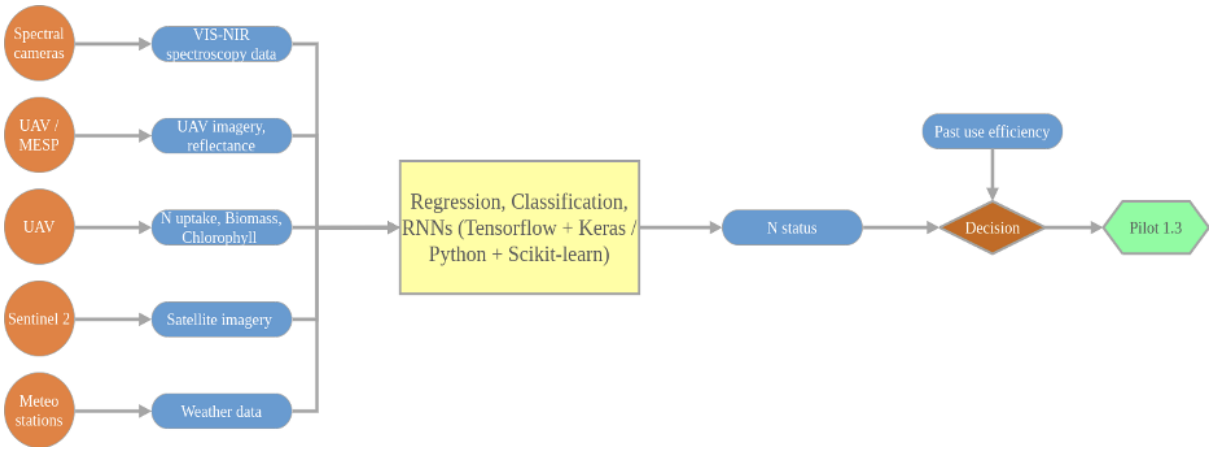


Figure 53:Pilot components in Pattern Extraction for Optimal Fertilizer Usage

11.3.4 Data Analysis for Water Salinity and Plant Toxicity (salt) in Rice Fields

Water is an essential factor for agriculture and for the rice to grow. The water quality, such as the salinity should be controlled with special sensors in order to keep it on certain amounts. The plain of Central Macedonia has one of the most complete irrigation systems in Europe, mainly utilizing the water of the river Axios. In the first stage, the water is channeled to each area of the plain through large canals, as shown in the following images(Figure 54 and Figure 55) of the canal that passes in front of the ELGO station.

Figure 54: Experimental station of Kalochori ELGO DEMETRA

The canal can also be seen in Figure 55, between the station's buildings and the road that runs north of the fields. From the main canals start smaller canals, which pass through all the fields.

Figure 55: (a) N plots, (b) S plots

These are located at a higher level than the fields, so producers put large plastic pipes to flood the rice field. On the other side of the plate there is a corresponding drainage channel, from where the drainage takes place (shown in the following Figure 56 between the two parallel rural roads).

Figure 56: Single image from UAV

Producers dig the embankments at two or three points for drainage and then close them again. The rice paddies are leveled with a small slope (2 cm per 100 m) using a laser for this purpose. The channels are connected to the main channels by doors. Because the water is not enough to irrigate all the rice paddies at the same time, the distribution is gradually starting from the west (Klidi Imathia area) to the east (Kalochori area). During the growing season, the water is distributed in a circular motion in a corresponding way, so each rice field has a water supply every 10-15 days. The distribution of water is done by the Local Land Improvement Organizations (LLIO), which are supervised by the General Organization of Land Improvement (GOEB). LLIO employees periodically go through all the areas and open and close the doors of the canals.

In addition to the drainage and bedding of the rice paddies related to the various cultivation works, there are others throughout the growing season, from sowing to ripening (late August to early September). On the one hand, amounts of water either evaporate or escape through the soil, so it must be supplemented to maintain the desired level. On the other hand, changing water is required to avoid the creation of anaerobic conditions resulting in the growth of stagnant microorganisms. Also, if the salinity of the water increases, all the water must be drained and re-introduced. The main goal is to flush out the saltiness as the salt can move very easily (saltwashing). The ideal water level depends on the growth stage. In the germination stage it ranges from 5 to 10 cm, as the high water height (over 15 cm) during the growth of the seedlings results in the development of slender stems and generally underfed plants and the slow growth of roots. After the end of the development of first leaves, a gradual increase in water height of up to 15 cm is required to ensure the development of panicles (panicle is called the inflorescence in rice and oats, instead of cob used for other grains) and the formation of viable pollen in the event of relatively low temperatures.

In practice, producers change the water every 15 days or so, especially when they see signs of salt toxicity, when the leaves turn white or light yellow color due to lack of photosynthesis or after measurement of electrical conductivity. The irrigation process is manual: they place the pipes in the irrigation canals and at the same time, grooves are opened in the drainage channel. After a few hours, the grooves close again and the tubes are removed. The exact time required is calculated empirically. However, due to the fact that many producers manage a large number of rice fields as

well, but also due to the large fragmentation of the land, is a very common phenomenon of excessive water use (many times the level can reach 20-30 cm), thus wasting large amounts of water.

The solution proposed in the context of DEMETER addresses the above problems. The following Figure 57 shows the standard salt sensor developed in the FP7-SME Smart-Paddy program and that will be used as a base for the Pilot needs. The communication in the original design was done via WiFi, which presents many problems, mainly due to the need for a WiFi solution. In the area of the ELGO station there is usually relatively good coverage of the mobile network (3G/ 4G), which is true for most areas of the plain. So, we suggest the solution of using GSM modems and communication via GPRS or where there is a problem, communicating the prototypes via SMS messages. Thus, it will be possible to apply to all the rice fields of the producers we will work with. In addition, the Smart-Paddy's prototype will be further improved by adjusting the rice water height sensor, allowing the producer to control the amount of irrigation water, while minimizing visits to control the water level. In specific experimental fields of the station, the solution of automatic water management will be tested, through automatic solenoid valves both on the irrigation canal side and on the drainage side of the drainage channel. If the electrical conductivity exceeds a critical limit (2-3 - 3.0 dS/m), they will allow the start of irrigation until the electrical conductivity is reduced below the limit. This conductivity limit is derived from experimentation under the Smart-Paddy program, which determined the electrical conductivity limit above which the output is significantly affected. Also, the height of the water in the rice field - as measured by the height sensor - will control irrigation automatically using the delimited height limits (10 cm). It will also enable the reduction of water in case of need to reduce the amount of water such as the application of herbicides, but also the increase of water when it needs to reach the set limits.

Figure 57: Prototype of the salinity sensor Smart-Paddy, that was placed in a rice field of experimental station of Kalochori of ELGO in 2013.

Monitoring of water quality that exists in a distribution system is too complex and sensitive a process as a result of different factors. For instance, different water qualities coming from different sources and treatment plants, the multiplicity of paths that water follows in the system and the demands that change over the week from the final users make it difficult to predict the water quality at a given point of the system life time.

In general, the water quality (WQ) is measured by the analysis of a few parameters, for example: Starting with input that we can gather, the first one is the water height, another one is the salinity/conductivity. We have to mention here that salinity is measured by the conductivity in the soil. Temperature of water is used in order to determine conductivity. Also, we could use images from UAV that show the salinity level of the plants by analyzing the leaves. UAVs can be used in cooperation with satellite images so that we can identify plant toxicity of salt which can be identified until August. Satellite data can be used on planet scope with payment or Sentinel 2 (without enough accuracy). There is no need for image analysis on rice, as far as thermal images are concerned, because there is no need for more water. The rice is fully flooded in water. Forecasting is useful if there is a provision of rain. One way to conclude in this provision is by barometric sensors. Usually when barometric pressure is low, this is an indication of rain in the near future. NDVI (Normalized difference vegetation index), gathered by UAVs can output information about salinity of the plant. Chlorophyll, which is gathered by the UAV is also used for salinity. We can get input from satellite/UAV/SIS (evolution of Smart-Paddy) /weather forecast in real time about the temperature and output the evaporation in high temperatures. All the above described sensors can be based on solar energy and specific batteries so that they can operate continuously without the need of using network power plugs. Another input that we can gather is relative humidity, air temperature, leaf wetness, solar radiation from data gathered from the experimental station of ELGO. Carbon content can also be used as measurable input if needed. There could be data fusion based on weather services, ELGO station data, MESP (Mobile Environmental Sensing Platform), XMachina stations, national meteorological station at 5 km far.

Pilot 1.3 will provide a service for maximizing water use efficiency in rice, through the deployment of appropriate sensor systems and science-based decision making. Thus, both water quality (e.g., salinity levels) and quantity will be optimized. Since irrigation is tightly linked to fertilization, a nitrogen fertilization advisory service will be set up, leading to optimization of the spatial distribution of nitrogen application based on the real needs of the field.

As far as the historic data that ELGO maintains, those are: mean temperature, max temperature, min temperature, mean RH%, max RH%, min RH%, rainfall in mm, solar radiation (kw/m2), leaf wetness %, absolute average indices reflectance.

We can use the previously referred data with the use of Statistical Models and Machine Learning algorithms, for example: logistic regression, linear discriminant analysis, support vector machines (SVM), artificial neural network (ANN), deep neural network (DNN), recurrent neural network (RNN) and long short-term memory (LSTM) and make prediction about temperature levels, salinity level, future weather condition such as rain. A known problem concerning real-world data is the fact that the data contain noise and are imbalanced to a high degree. In addition, it has been proven that highly imbalanced data sets are difficult to explain and make predictions. Researchers often consider imbalanced classes a minority of 10% -20%, however in reality, can be more imbalanced than this. Recurrent Neural Networks (RNN) is a dynamical system, the next step and output of which depend on the present network state and input. RNNs and LSTMs are good enough at extracting patterns in input feature space, that the input data extends over long sequences. They can almost seamlessly model problems with multiple input variables, this brings great advantage of time series prediction, where classical linear methods can be difficult to adapt to multiple variations or multiple input problems for forecasting. More information about the Machine Learning algorithms is given in the last section. These techniques can be used on pilot round 2.

Input can be taken from the salinity sensors that will be used from the Pilot 1.3 and the UAV images that with machine learning algorithms will output the water stressed fields in order for the farmers to take action. Machine learning algorithms consume time on training level, so specific GPU accelerators could be used in order to get results faster. As far as the image analysis is concerned, when we want to extract information on images, a mainstream machine learning technique is the Convolution Neural Networks. When we want to train on a huge dataset on thousands or even more images, we use GPU accelerators that diminish the time consumed in relation to using multicore CPUs.

Tools/libraries that we could use, for example, for Machine Learning algorithms (not only for CNN) are the following: TensorFlow: a fast, flexible, and scalable open-source machine learning library for research and production, Keras: one of the most popular and open-source neural network libraries for Python, PyTorch: which is a ML library that supports also C++, Scikit-learn: which supports algorithms for classification, regression, clustering, dimensionality reduction, model selection, preprocessing, Pandas: for dataset reshaping and pivoting, merging and joining of datasets, handling of missing data and data alignment, various indexing options such as hierarchical axis indexing, data filtration options, Spark MLlib: developed by Apache is a machine learning library that enables easy scaling of the computations, such as regression, clustering, optimization, dimensional reduction, classification, basic statistics, feature extraction, Theano: for enabling optimizing, easy defining and evaluation of powerful mathematical expressions and supports GPUs when we need to cope with heavy-data computations, MXNet: a library for ML that supports programming languages such as C++, Perl, Julia, R, Scala, Go, Numpy: which is a library for handling multi-dimensional data.

Also, there will be data analytics, decision support mechanisms for optimal resource management/allocation (e.g., water use optimization, optimal scheduling of snapshot propagation). The data that will be used are measurements from the salinity sensors and results from the UAV images, so that knowledge extraction and decision making comes out.

From the above gathered data, an essential output could be, when (decision making) to open/close electro-valves in order to irrigate the field and open/close drainage valves. By the Machine learning algorithms on gathered images we can specify which areas of the field need more water in order to get rid of high salinity levels. When there is forecasting, (based on data from the weather station) about rain in the near future, there is no need for human-activated irrigation.

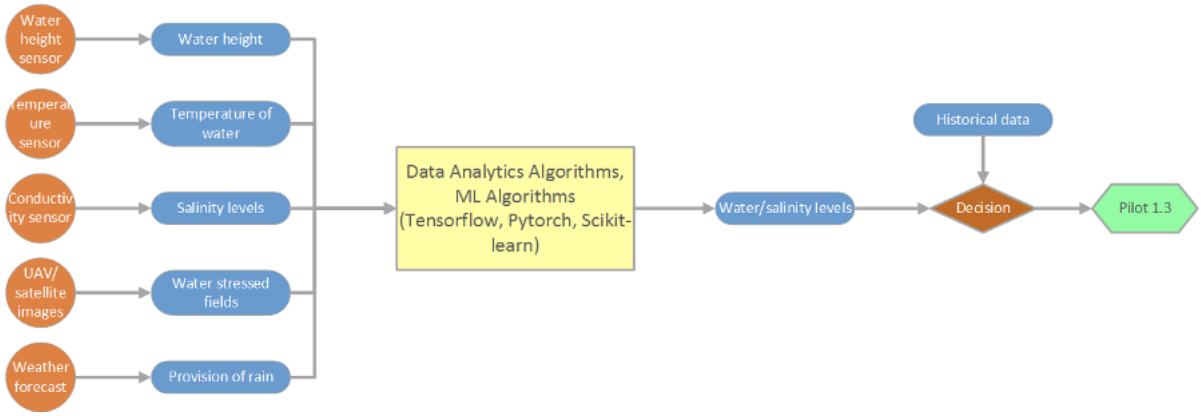


Figure 58: Schematic diagram of the described solution

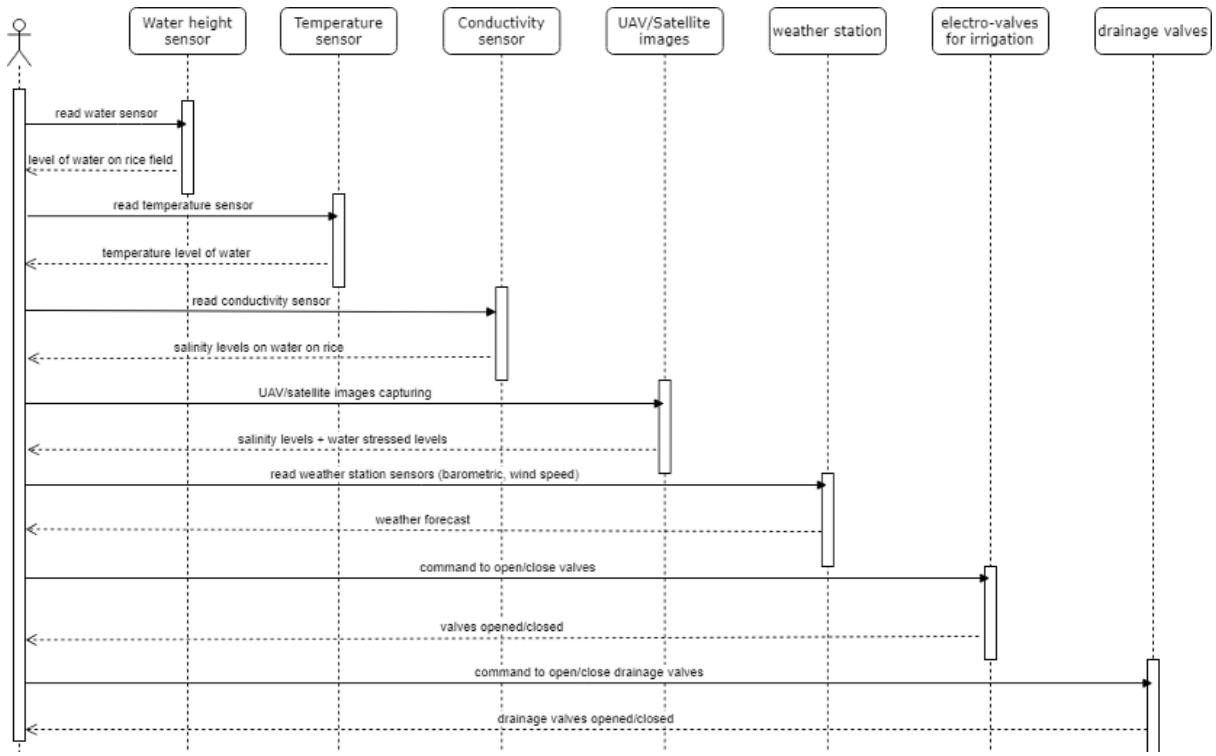


Figure 59: The UML sequence diagram to the solution of the above figure

This is the last section, where we analyze in more detail the Machine Learning algorithms that can be used, for example, for water quality and irrigation.

Logistic Regression is a simple algorithm that can perform well on a wide range of problems. It is used when the dependent variable is binary, TRUE - FALSE, and also to describe data and to explain the relationship when we have one dependent variable and one or more independent variables. Linear Discriminant Analysis can be used in order to find a linear combination of features characterized by two or more events. LDA works well in the case that the measurements of the independent variables are continuous quantities. Classical pattern recognition techniques are an interesting problem to experimental time series data. We have to note at this point, that time series classification problems are not restricted to geophysical applications but take place in varied and many circumstances in other fields. Artificial Neural Network (ANN) is a good solution for classification of complex data sets. The structure of the ANN reminds the structure of the human brain. It involves a network of many interconnected neurons, which are very simple processing elements that each one combines pieces of one big problem. Every neuron computes one output by using an activation function that regards the weighted sum of all inputs. The most known activation is the logistic sigmoid function: $f(x) = \frac{1}{1+ e -x}$ where $f(x)$ is the output of a neuron and stands for the weighted sum of inputs to a neuron. Support Vector Machines (SVM) can be used for classification and regression problems, and it is among the best 'off-the-self' supervised algorithms and it is a linear two-class classifier. The rationale of the linear classification is the finding of a hyperplane that is able to classify data points appropriately. SVM is able to forecast time series data when the underlying system processes are typically and a-priori nonlinear, non-stationary and non-defined. SVMs can excel in performance other non-linear techniques such as MLP (Multi Layer Perceptrons). Deep Neural Network (DNN), is a feedforward neural network with many hidden layers which aims at learning feature hierarchies using features from higher levels of the hierarchy formed by the composition of lower level features. The models are called feedforward because

information is based through the function being evaluated from x. Feedback connections do not exist. Recurrent Neural Network (RNN) uses recurrent loops where the cell's output state is fed back into the input state. Long Short-Term Memory (LSTM) is a form of RNN that has a more complicated cell architecture for more accurately maintaining the memory of important correlations. The purpose of the RNN is to model temporal sequences and their long-range dependencies more accurately than the conventional RNNs.

11.4 Targeted Data Fusion Modules

"Data fusion techniques combine data from multiple sensors and related information from associated databases to achieve improved accuracy and more specific inferences than could be achieved by the use of a single sensor alone."[120]

The integration of data and knowledge from the same or several different sources is called data fusion. There is a little difference in data and information fusion. The term data fusion is used for data coming directly from sensors while information fusion term is used for already processed data.

Data generated by wireless sensor networks often has redundancy, impreciseness, uncertainty, outliers and quality issues. Measurements taken by sensors may not be perfect due to an external noise. Some sensors may stop working for some time or may lose connectivity. Merging of data from many sensors may remove outliers and redundant data which helps in sending less data within network. It increases the efficiency of the network and reduces energy consumption. The batteries may go longer. Moreover, it improves confidence, accuracy, reliability and data quality. The fused data may be more relevant, less expensive, and of higher quality.

DEMETER based solutions are expected to work with very heterogeneous data coming from a wide variety of components and sensors, which might however capture similar aspects of the real world. Some of those sources might require pre-processing or fusion techniques to improve the data quality. The chosen approach for data fusion depends on the intended usage. E.g. images with different characteristics can be fused to provide better images. In other cases, data coming from very different data sources might be fused in order to generate improved multimodal datasets (e.g. data coming from weather stations, fused with crop estate estimation features calculated from pictures taken to the plants).

As aforementioned, a wide variety of data sources are expected to be processed using data fusion, and it has to be known upfront what the purpose of the fusion task is. This fusion module hence has to be context-specific in order to generate useful results. That is why one of the key components to perform this fusion task is the DEMETER AIM, which aims at the interoperability of the different components that will take part in the DEMETER environment. By means of its usage, the context of each piece of data can be properly merged following the conditions to be defined in each of the data fusion components. Initial definitions and designs regarding Semantic Interoperability Support (to be used also by the data fusion components) to be carried out in DEMETER can be seen at D2.1.

In order to facilitate the fusion process, it is expected that all the data (including in particular the image data) should be tagged with information indicating the time and GPS location of where they were obtained, e.g. where and when the image was taken. In case different data are taken in different conditions (or methods), e.g. using different spectra for the imagery data, or using a different type of sensor for e.g. weather data, this information should be saved in metadata together with location and time.

The DEMETER Data Analytics and Knowledge Extraction Enabler aims at harmonizing all data fusion tasks and provide a common implementation template and interface for all fusion tasks. In the following, a few exemplary fusion tasks are presented. In later stages of the DEMETER project, more fusion tasks shall be made available via the same interfaces to enable re-usability of those methods via the DEMETER Enabler Hub.

11.4.1 Fusion of Satellite, Spectral and UAV Imagery for Rice and Maize Fields

One type of data that appears in several DEMETER pilots is imagery data, in particular satellite imagery and other imagery for pattern extraction tasks.

Regarding fusion of satellite imagery, different variables have to be taken into account, ranging from the different kinds of data that will be processed (multispectral data, radar data, etc.) to the different algorithms to be used (Brovey, PCA, etc.) or their combination with different AI based techniques for fusion.

When the images are used for pattern extraction, a different processing might be required, due to the different nature (i.e. different shape, color, photography conditions, etc.) of the images to be processed. Nevertheless, as far as the aim of the pattern extraction scenarios share a main goal (e.g. counting elements), the data fusion approach for those pilots might share a common methodology and tools, while using a different set of training images.

Pilot 1.3 aims to improve the management and automation of rice irrigation, along with nitrogen zonal fertilization. Due to crop rotation, maize is also planted once every three years, thus maize irrigation management is also a goal of this pilot.

Now, imagery is an important data source for this pilot. Multispectral and thermal[163] images will be collected at predefined points within the cultivation period using UAVs. Analysis of these images allows to identify water- or nitrogen-stressed fields through an automated image processing workflow. More specifically, this analysis will help to identify whether plants photosynthesize properly by checking the color of their leaves and stem, give input regarding the height of the plants as well as the height of the water (in rice paddies), which is important for the health of rice plants and the yield of the rice field in general. Some imagery can also be pulled from satellite data and fused with the UAV images.

Other sources of information include sensors that measure a number of water metrics, such as water conductivity (from which water salinity can be derived), water height, water temperature. GPS data from the machinery used when fertilizing the fields which provide the locations and precise quantities of fertilizers used. In addition, another very important source of info will be the weather data (and forecasts) obtained from various different sources and sensors.

Regarding fusion, the analysis from the images of the same field using different spectra of light will be fused together; they will also be combined with the images of the same field taken at previous times in order to facilitate the data analytics process. In addition, the resulting data from the aforementioned image analysis will be combined with input and the resulting analysis from the other sensors. The most important avenues of data fusion that will be performed in this pilot are: the image analysis regarding the color and health of the plants will be fused with information regarding water salinity measurements (which are derived from electrical conductivity sensors in the smart paddy sensor planted in rice paddies); the imagery regarding the fertilization level will be

163 Thermal images will be used for maize irrigation only, as they do not provide useful information for rice which is submerged in water for most of its cultivation.

combined with other sensor data such as GPS data of where fertilization took place in order to reduce the fertilization stress of the fields.

11.4.2 Fusion of Weather information

Another type of fusion pertains to the fusion of data describing the same type of information and this will be primarily performed on weather data. More specifically, weather information will be provided by: a) privately already owned weather sensors of ELGO, b) ExMachina weather sensors, and c) the nation weather service (which has a nearby weather station to the area where the pilot takes place). The information from all three sources will be fused in order to increase the accuracy and quality of obtained data. All of these (weather and imagery data) in turn will be used together with weather forecast data from appropriate services, as well as sensor data regarding water salinity, temperature etc. as input for the analytics and the decision support tools that will control the irrigation of the fields as well as the usage of fertilizer to maximize the yield while reducing the waste of resources (water and fertilizer).

11.4.3 Fusion of Fruit Fly Imagery

Pilot 3.3 aims at using imagery data in order to identify sterile fruit flies. In order to do that, glowing flies have to be counted in comparison to not-glowing ones. Regarding data fusion, features to be extracted from image processing in this pilot should aim at an estimation of the number of glowing and not glowing flies.

Regarding the data fusion, it has to be seen how the data is provided to the component developed in DEMETER. The pictures will be taken by a smart trap that should provide the best conditions to boost the glowing nature of the modified files to be identified (by means of a dark environment as a chamber and UV light), as well as the conditions to capture the not glowing ones.

The expected output of the counting module should be provided properly timestamped and geolocated, in order to allow its fusion with the other data gathered in the pilot. By means of this, it is expected to allow following up the state of the pest as well as the impact of the different actions performed to reduce it.

In order to provide the output labelled as aforementioned, it is expected the input data to be provided with that information (e.g. pictures with meta-data indicating the time and GPS location of the picture taking). In case different successive pictures are taken in different conditions to ease the counting of the different types of flies (e.g. with normal light to count all the flies and another with UV light to count only the glowing ones), the different conditions of the picture should be also saved in the image metadata. Additionally, a possible ID for successive pictures could be created (although the fusion of the data from those successive pictures can be performed by means of the date and time in the image metadata).

Due to the early stage of the pilot and the lack of fly traps and data examples, it is still soon to get a detailed idea of the fusion mechanisms to be used in the data fusion of this scenario. Nevertheless, given the nature of the data to handle in this pilot, it wouldn't be difficult to carry out the approach here introduced if the fly traps provide the capabilities of getting timestamped and geolocated pictures (with UV lighting).

11.5 Training Data and Label Acquisition

Supervised machine learning requires labelled training data. In some cases, labels can be retrieved from the data itself (e.g. for yield forecasting) but in other cases, samples have to be manually

annotated (e.g. for crop type classification). DEMETER will offer a set of tools to facilitate the task of data labelling.

Manual Labelling of Imagery

Computer vision requires a large amount of training data. Transfer learning can reduce the amount of required data, by exploiting visual patterns present in domain-independent imagery. Still, even in a transfer learning setting, for tasks like object detection, each training sample, requires a corresponding file depicting the relevant areas in each image. In general, the following challenges have to be taken into account when designing the image labelling process:

- the type of the label (point or area), the shape of the label (in case it is an area), the level of detail, criteria to label or not in arguable cases, etc. has to be agreed upon by all collaborators in the labelling process
- the format of the labels has to be in line with the used pattern extraction component appropriate choice of a data labelling tool, such as e.g. LabelImg[164], VGG Image Annotator[165], makesense.ai[166] has to be made. Other approaches such as using crowdsourcing approaches such as Amazon Mechanical Turk can be evaluated in case the number of images to be labelled is high and the labelling requires no special skills. This poses the questions of how to account for external factors which are not encoded in the image metadata (distance to the object, external conditions, etc.).

Additionally, the possibility of using external image datasets should be taken into account. In that case, the identification of possible existing labelled image training sets should be carried out in the initial stages of the pilot, before starting the manual data labelling process. Figure 60 below illustrates the labelling process from a high-level perspective

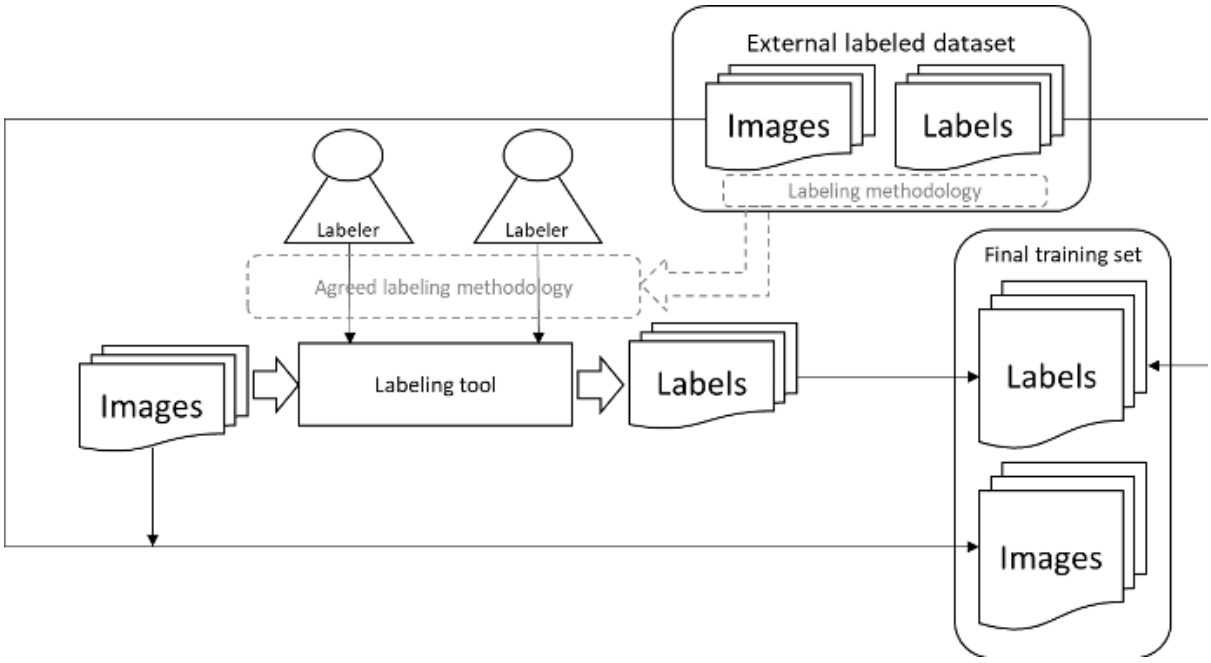


Figure 60: Process for manual labelling of imagery

164 https://github.com/tzutalin/labelImg

165 http://www.robots.ox.ac.uk/~vgg/software/via/

166 https://www.makesense.ai/

In case of implementation, the mentioned challenges have to be addressed in close correspondence with DEMETER pilots and the resulting insights have to be incorporated into an appropriate solution. Generally, the solution will have to blend in with the general Data Analytics and Knowledge Extraction Enabler architecture, by offering a labelling API and User Interface in form of a microservice and by being compatible with the DEMETER AIM data model.

Rule-based Labelling (Weak Supervision)

Weak supervision is a relatively novel concept in machine learning, which advocates the rule-based definition (example in Figure below) of data labels as opposed to a manual case-by-case labelling process. Based on a small set of hand-labelled data, weak supervision exploits programmatic rules or explicit knowledge encoded in knowledge graphs to perform automatic human-like labelling of large data sets. As the semantic AIM-based DEMETER Data & Knowledge repository is a large Knowledge Graphs, it is natural to aim at exploiting semantic relations in the graph to facilitate data labelling tasks. Weak supervision was proven to be particularly useful for labelling large textual corpora and to a limited extent useful for image labelling, however, usage scenarios in the scope of DEMETER are yet to be explored.

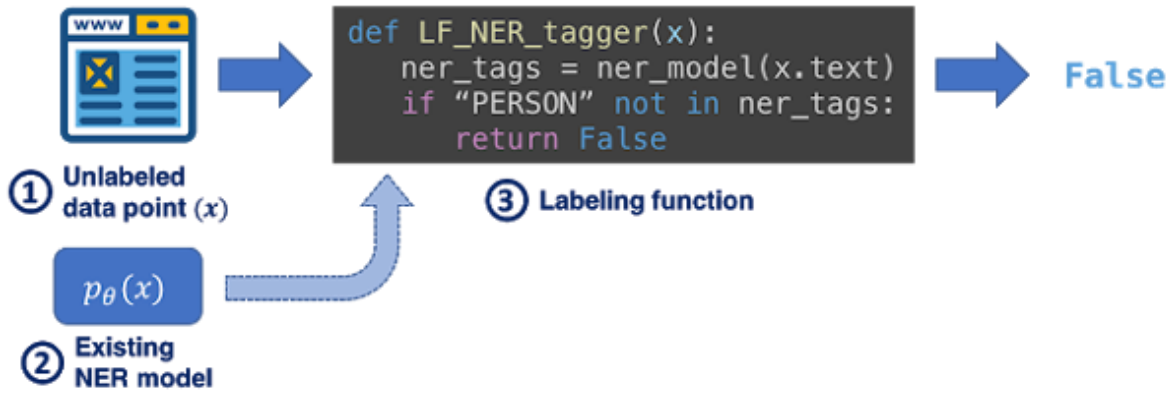


Figure 61: Example labelling function[167]

Snorkel DryBell [144][168] offers an open-source solution, enabling weak supervision in a production setting. DEMETER will offer an instance of Snorkel DryBell and an integration with the DEMETER Data & Knowledge repository as a knowledge source to enable weak supervision for large labelling tasks. The implementation will be performed on-demand and in close alignment with a corresponding pilot use-case which has the requirements of labelling a large set of data for machine learning purposes. Further requirements are expected to arise in this process.

Labelling Services

If neither manual labelling nor weak supervision is a feasible solution, then crowdsourcing platforms such as Amazon Mechanical Turk[169] could be facilitated. DEMETER will offer an integration of a crowd-sourcing solution provider. Crowdsourcing services usually offer convenient API[170], which can be easily integrated into the DEMETER architecture. However, the question remains whether the pay-as-you-go model of crowd-sourcing services is feasible for DEMETER and/or DEMETER pilot partners.

[167] https://ai.googleblog.com/2019/03/harnessing-organizational-knowledge-for.html
[168] https://www.snorkel.org/
[169] https://www.mturk.com/
[170] http://docs.pythonboto.org/en/latest/ref/mturk.html

11.6 Algorithm Selection and Feature Engineering

Algorithm selection, hyperparameter tuning and feature engineering are laborious tasks, usually performed by data scientists and data engineers. Tools like auto-sklearn[171] and featuretools[172] can partly automate respective tasks.

Feature Engineering

The DEMETER Feature Engineering component will offer an automated feature engineering facility, utilizing featuretools. Deep Feature Synthesis [143], as employed by featuretools, uses relations in relational database schemas to automatically generate various synthetic features such as temporal aggregates. An exemplary implementation reads as follows in Figure 62:

```
es = ft.EntitySet('Dataset')
es.entity_from_dataframe(dataframe=data,
                                 entity_id='recordings',
                                 index='index',
                                 time_index='time')

es.normalize_entity(base_entity_id='recordings',
                                      new_entity_id='engines',
                                      index='engine_no')

es.normalize_entity(base_entity_id='recordings',
                                      new_entity_id='cycles',
                                      index='time_in_cycles')
fm, features = ft.dfs(entityset=es,
                      target_entity='engines',
                    agg_primitives=['last', 'max', 'min'],
                    trans_primitives=[],
                    cutoff_time=cutoff_times,
                    max_depth=3,
                    verbose=True)
fm.to_csv('simple_fm.csv')
```

Figure 62: Exemplary implementation of Demeter Feature Engineering component

As opposed to classical RDBMS, the DEMETER Data & Knowledge repository, which will serve as the main source for training data, contains data in graph form. No comparable feature engineering solutions exist, yet, for linked data. Hence, to support automated feature engineering on the DEMETER Data & Knowledge repository, the Deep Feature Synthesis approach as implemented in featuretools has to be extended towards the support of graph relations. The adapted featuretools library will then be wrapped in a component and offers an API that can be utilized by all Analytics and Fusion modules.

Algorithm Selection:

The so-called AutoML approaches facilitate the choice of machine learning algorithms and respective hyperparameters. The DEMETER Data Analytics & Knowledge Extraction Enabler will provide a

[171] https://automl.github.io/auto-sklearn/master/
[172] https://github.com/FeatureLabs/featuretools

wrapper for auto-sklearn[173], an automated machine learning library that automates both, algorithm and hyperparameter selection. The process algorithm selection can be further enhanced by making use of meta-information from the AIM data model, e.g., utilizing data quality information. Taking this information into account. In the case of weak or strongly varying quality of training/prediction data, the candidate set of algorithms can be artificially reduced to more resilient models. An example implementation of a model training using auto-sklearn reads as follows. Note, that the cross-validation results and the final model can be retrieved. The respective DEMETER component will use this information to provide a report containing those statistics to the user:

```
import autosklearn.classification
import sklearn.model_selection
import sklearn.datasets
import sklearn.metrics
X, y = sklearn.datasets.load_digits(return_X_y=True)
X_train, X_test, y_train, y_test = \
       sklearn.model_selection.train_test_split(X, y, random_state=1)
automl = autosklearn.classification.AutoSklearnClassifier()
automl.fit(X_train, y_train)

automl.cv_results_
automl.sprint_statistics()
automl.show_models()

y_hat = automl.predict(X_test)
```

11.7 Auditable, Explainable and Fair Analytics

To counteract the opaqueness of the state-of-the-art approaches, we can apply different explain ability and interpretability logics followed by both ante-hoc and post-hoc approaches to align AI-assisted decision support system of GDPR w.r.t explain ability, algorithmic transparency, and decision fairness. In particular, widely used concepts such as feature importance (some of the features have a higher impact than others), Shapley values, saliency maps, and gradient-based attribution methods along with ontological reasoning can be employed to enable auditable, explainable, and fair precision farming.

Besides, feature importance that can be computed based on permutation importance, perturbation based methods such as Shapley values are also emerging. Shapley value is the average marginal contribution of a feature value over all possible combinations of features. Saliency map and gradient-based attribution methods are used to identify relevant regions and assign importance to each input feature, e.g., pixel for image data. Typically, first-order gradient information of a complex model w.r.t is used to produce maps that indicate the relative importance of the different input features for the classification. Gradient-weighted class activation mapping (Grad-CAM++), sensitivity analysis and layer-wise relevance propagation (LRP) are examples of this category. However, using a set of rules, it is possible to explain a decision directly to humans with the ability to look up the reason for a decision. On the other hand, to explain and interpret the model predictions feature

[173] https://github.com/automl/auto-sklearn

summary statistics (summary of each feature and their impact on the model predictions), feature summary visualization (summary of the methods used to visualize and in order to make the visual communication easier, where outcomes are presented with bars, plots, or tables), summary statistics for the features, and the data point interpretability are widely used options. For the imaging modality, LRP or Grad-CAM++ can be applied indicating the relevance for the classification decision.

Nevertheless, since the explicit representation of domain knowledge and data provenance through the layers of a DNN can pave the path to XAI, where interpretability, decision rules, and decision reasoning based on a knowledge base can be implemented. In particular, we can validate a precision farming decision via a semantic reasoner that characterizes and learns hierarchical relations from the KG to provide reasons and explanations about predictions trustworthy. Eventually, the ability of uncovering most important factors (such as roles of genotype and phenotype with their interactions with environment factors), explaining them to the farmers and the stakeholders towards fair and precision farming will help increase the production levels, bio-products quality, and help providing rich recommendations and insights for farmer decision support and action.

11.8 Model Storage and Management

Model storage and Management is an important component for productive systems with machine learning components. MLFlow is an open-source solution for model management. It offers a python interface, compatible with the most common machine learning frameworks. MLFlow can run either on local hardware or cloud infrastructure. For DEMETER, the MLFlow Model Registry module is most relevant, which acts as a centralized model store including an API and UI. The most straightforward way to store models using MLFlow, is to register a model which was previously created, as in the following example, where the first parameter is the model path and the second one the model name. The Figure 63 below shows the model registration process in the context of DEMETER

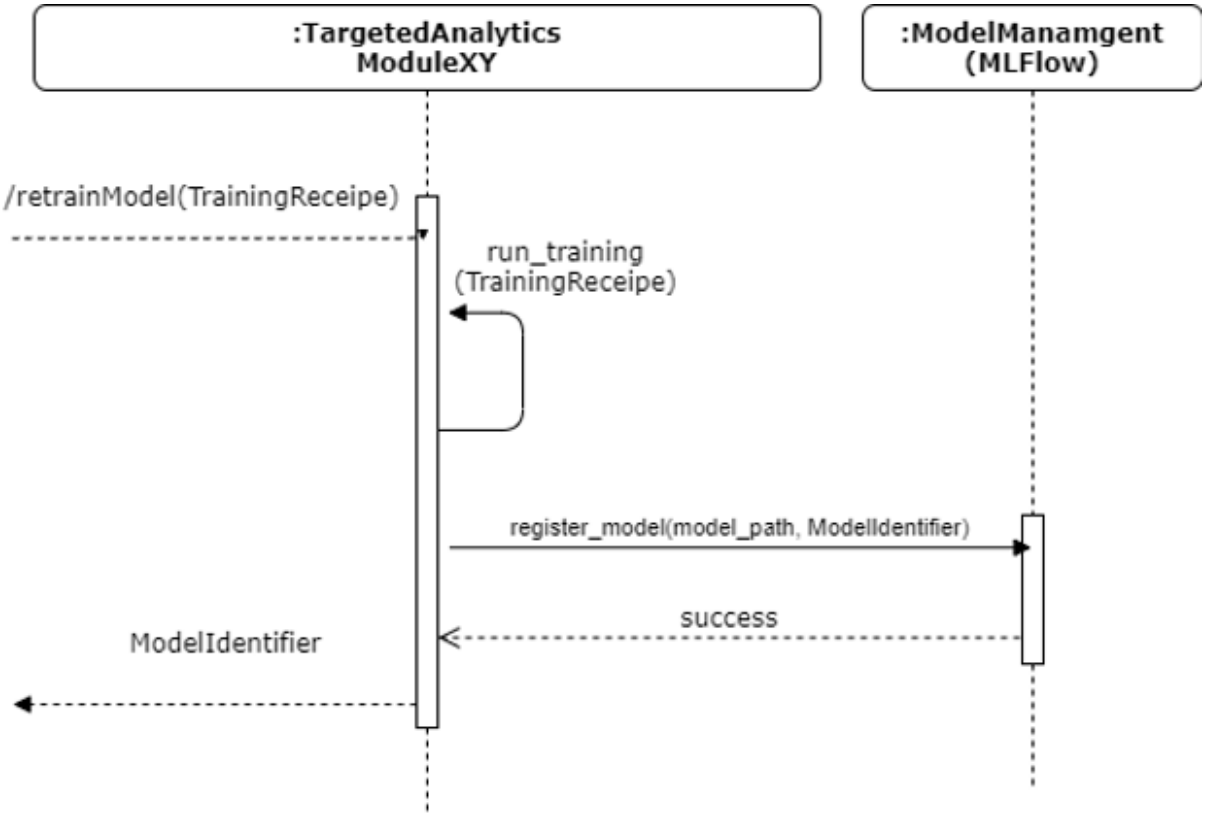


Figure 63: DEMETER model registry, using MLFLow

An exemplary code in Figure 64 for model registration, which has to be called within the Analytics module, reads as follows:

```
result = mlflow.register_model(
      "runs:/d16076a3ec534311817565e6527539c0/artifacts/sklearn-model",
      "sk-learn-random-forest-reg"
)
```

Figure 64: Exemplary code for model registration called within the Analytics module

MLFlow supports various storage systems, such as local hard drive, s3, and hdfs, and interfaces for programming languages Python, Java, R, as well as a REST API. It is hence very flexible and easy to integrate with the existing DEMETER architecture and does not require an additional wrapper.

11.9 DSS Integration

The definition and implementation of analytics and fusion modules will, apart from the few examples presented in this document, be majorly covered within the scope of deliverable D4.2, i.e. the DSS Enabler. This document presented basic components and concept which aim at enabling the development of such modules and provides a starting point by giving applied examples of such. Further development of knowledge extraction tasks and decision support and benchmarking enablers has to happen in close correspondence between partners involved in WP2 and WP4. The Figure 65 below provides a high-level overview of the current Data Analytics and Knowledge Extraction Enabler in the context of other WP2 and WP4 Enablers.

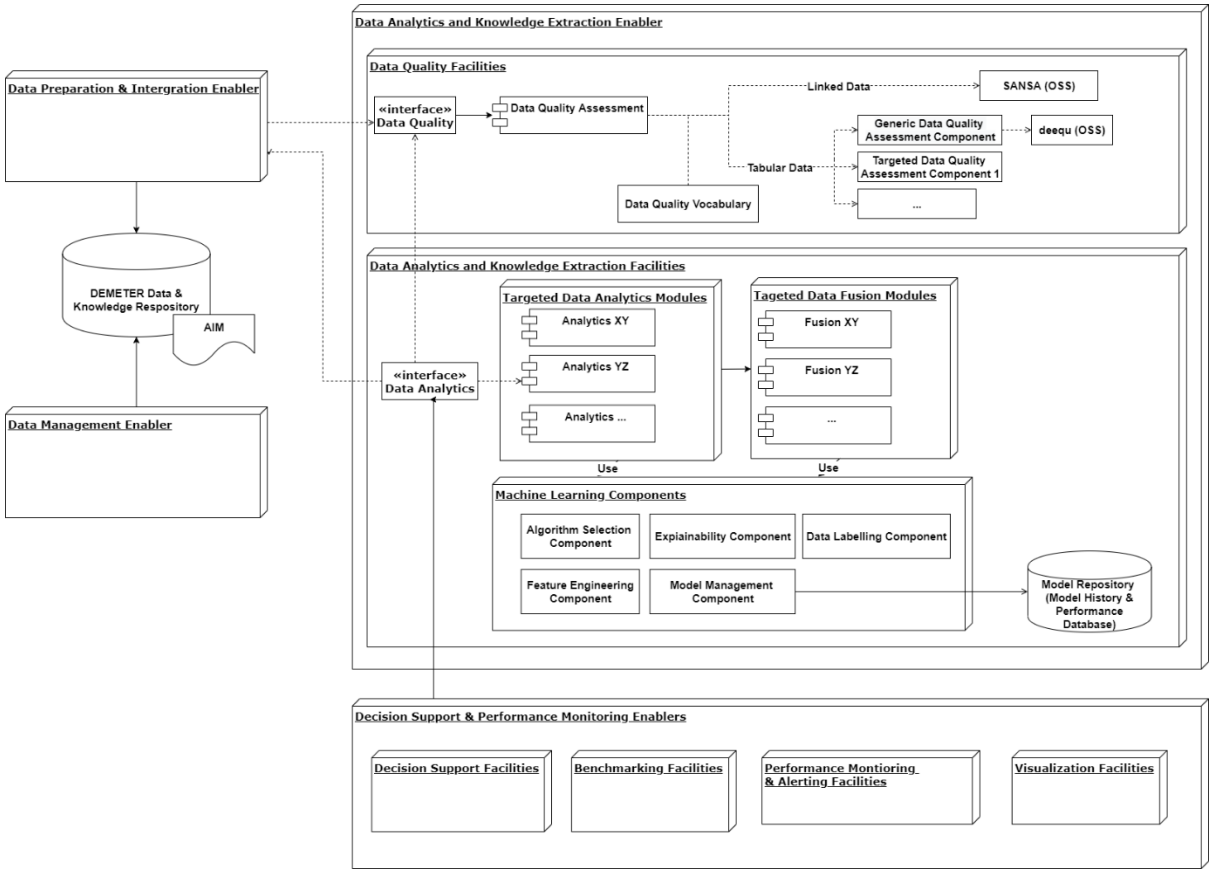


Figure 65: Process for manual labelling of imagery

12 Data security components

12.1 Overview

Security is a fundamental aspect that has been considered in the scope of the DEMETER project. More specifically, we have considered the access to the data at different levels ranging from access to the system, authentication, access to the resources, authorization; and access to the data, privacy or confidentiality. Besides data protection and regulations such as GDPR, and even the traceability of the operations commented above have been taken into account too.

Starting with the authentication, this operation is part of the management of the different entities which must be registered in a system so as to establish the corresponding permissions for accessing the system. There are different solutions in the market, such as Keyrock, a Generic Enabler from FIWARE, Workspace ONE access[174] (from VMWare) or Microsoft Identity Manager[175] , which could make use of different protocols for performing the authentication. There are some authentication frameworks, which are widely adopted and used, that have become standards such as OAuth2 or OpenID.

Regarding authorization, or the access control to the resources of a system, traditional approaches were not designed to deal with heterogeneous scenarios where interoperability is a must. Under the foundations of ZBAC (AuthoriZation Based-Access Control) [16] which roughly proposes a scheme where a central service is asked for permission which can then be used to access to a service, the

174 https://www.vmware.com/es/products/workspace-one/access.html

175 https://docs.microsoft.com/es-es/microsoft-identity-manager/microsoft-identity-manager-2016

proposal of using Distributed Capability-Based Access Control (DCapBAC) [17] was conceived. DCapBAC has been implemented as a DEMETER Security component to manage resources access control.

In addition, all these authentication and authorization components also provide a logging activity so that traceability is implemented and can be easily followed.

Any user/device wishing to access the DEMETER Dashboard or use the DEMETER services will have to login and, once logged and according to their granted permissions, will be able to recover a determined type or amount of information. These operations will be performed thanks to the Identity Management and Authorization components.

In the following sections it will be described the Security Architecture design, providing and overview description and providing more specific details of its four components: authentication, authorization, traceability and confidentiality, both at design level (section 12.2) and at implementation level (12.3).

The implementation of the security components can be found at the WP3 Security folder in DEMETER GitLab: https://gitlab.com/demeterproject/wp3/se

12.2 Design/approach (including UML diagrams)

12.2.1 Security architecture overview

The following Figure 66 depicts the relationships between the different data security elements:

- Identity Manager
- Authorization manager
- Information Audit tool

and their interconnection with other Demeter components such as the Demeter Dashboard (WebApp) and the Brokerage Service Enabler.

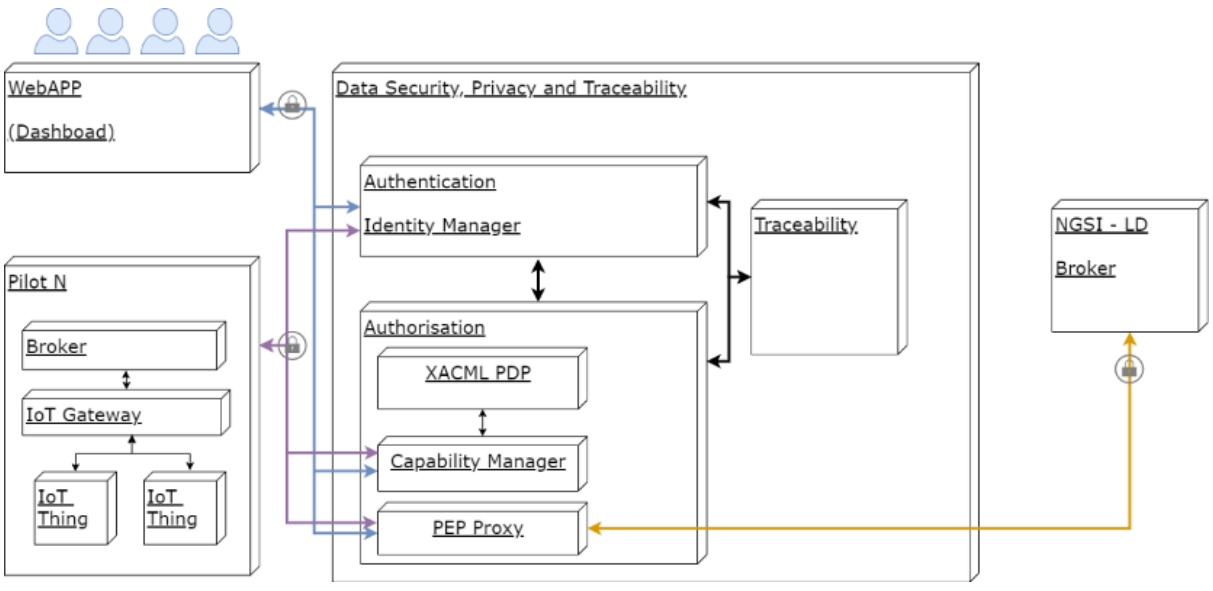


Figure 66: Security architecture components

The communication between components will be secured by the used of Transport Layer Security protocol provided the Confidentiality component/enabler.

A user trying to access a DEMETER resource should first get authenticated at the Identity Manage in order to obtain and authentication token. Once the user is authenticated, the authentication token will be used to request access to DEMETER resources through the authorization component, as described in the following sequence diagram in Figure 67.

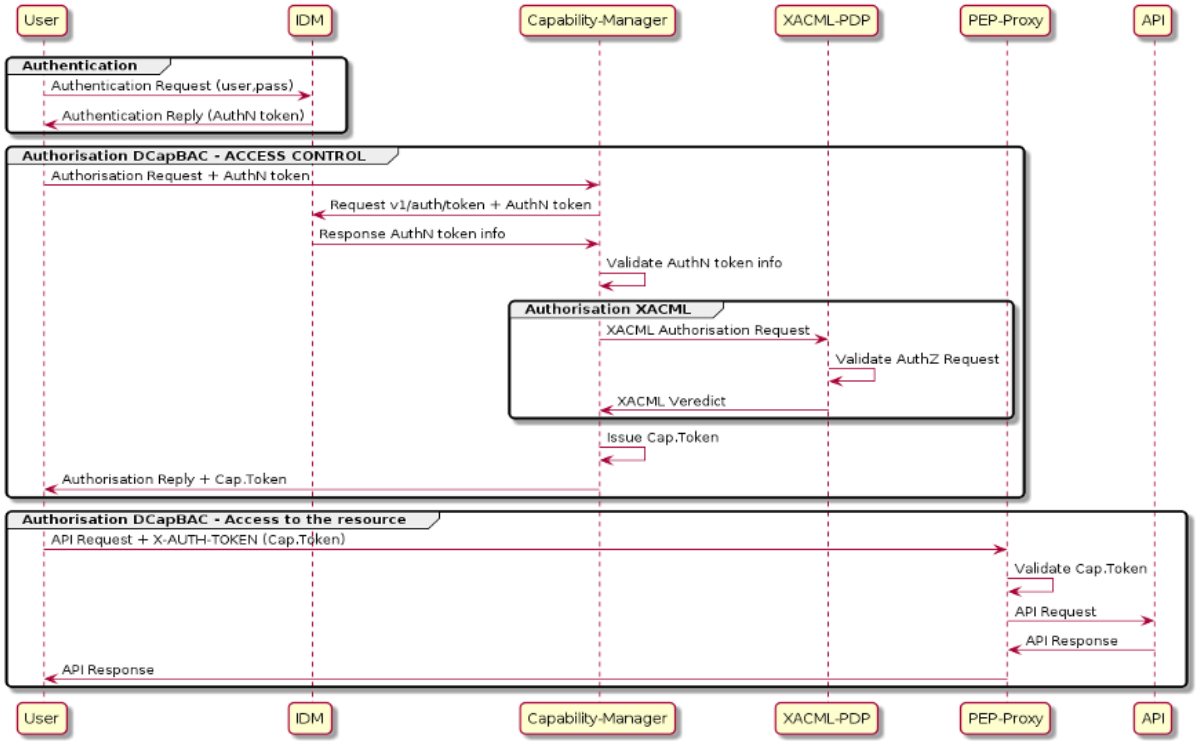


Figure 67: Authentication and authorization sequence diagram

12.2.2 Authentication

The Demeter Identity Manager (IdM) Enabler 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[176] [177] is defined in the RFC6749 standard, where is described as: "The OAuth 2.0 authorization 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

176 https://auth0.com/docs/protocols/oauth2

177 https://tools.ietf.org/html/rfc6749

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

- **Authorization Code**: defined for apps running on a web server. The client will redirect the user to the authorization server (Keyrock GE), and then the user will then be asked to login to the authorization server and 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 are 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 groups 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 affects 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, 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**: which identifies who has logged 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 the following Figure 68:

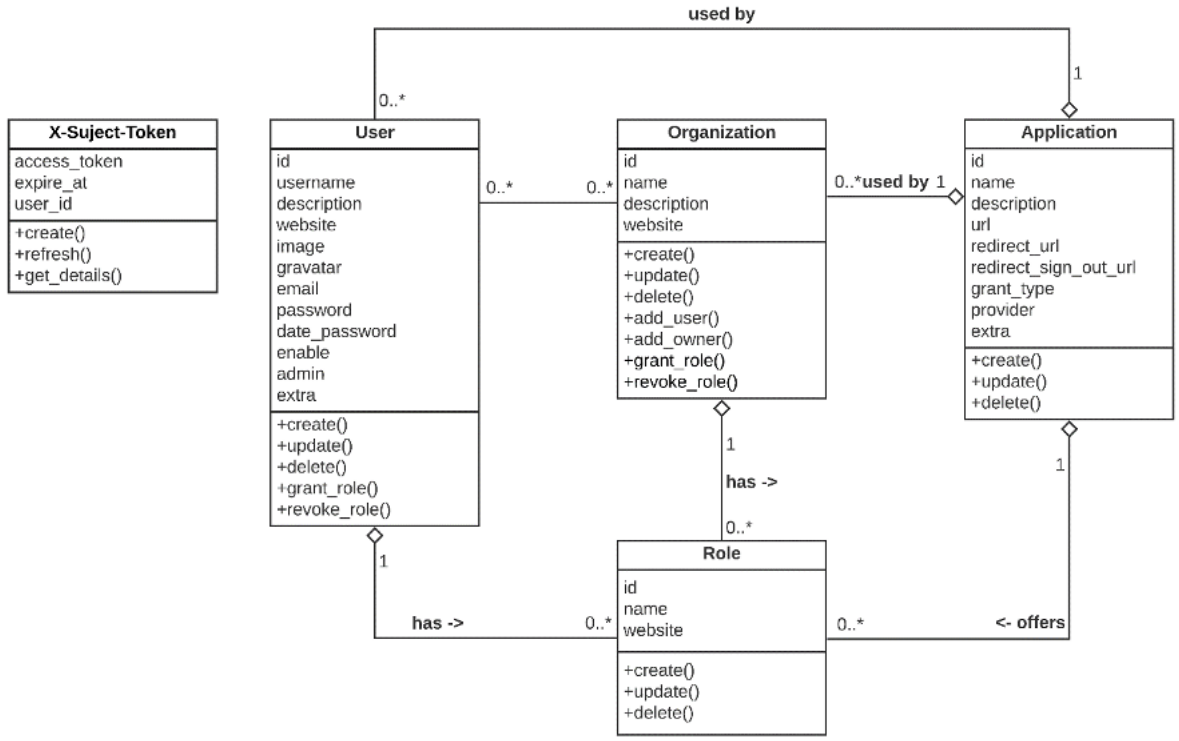


Figure 68: Relationship between Authentication objects

These IdM objects are used for the security layer to access DEMETER resources (functional layer) and are needed to provide authentication and authorization functionalities. The objects user, organization and role may be linked to objects defined within the Agriculture Information Model (AIM) and the potential relationship between them is described in later section.

The following Figure 69 shows the interaction that must be performed in order to obtain a X-Subject-Token for an authentication request with the Keyrock API:

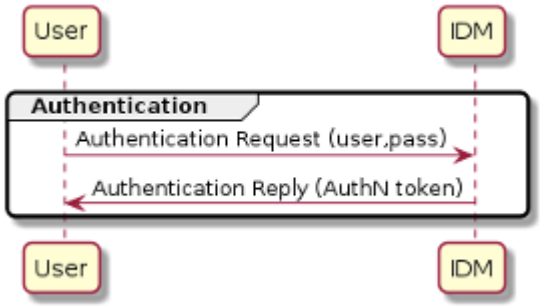


Figure 69: Authentication sequence diagram

12.2.2.1 User data model

The data user model is used to save, update and provide the user's information at 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**: and 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 to be used during registration for the 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 provide extra information.

12.2.2.2 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.
- **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 it is not configured it will be redirected to the domain indicated in the URL parameter.
- **Grant_type**: list of gran 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 the owner.
- **Extra**: field where a JSON object can be stored to provide extra information.

12.2.2.3 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.

12.2.2.4 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.

12.2.2.5 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.

12.2.2.6 Authorization 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[178], organization[179] and role[180].

12.2.2.6.1 Person

The definition of Person 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 in Table 7:

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

Table 7: Person fields mapping to Keyrock User Data Model

12.2.2.6.2 Organization

The definition of Organization 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 in Table 8:

[178] https://schema.org/Person
[179] https://schema.org/Organization
[180] https://schema.org/Role

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

Table 8: Organization fields mapped to Keyrock Organization Data Model

12.2.2.6.3 Role

The definition of Role 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 in Table 9:

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

Table 9: Role fields mapped to Keyrock Role Data Model

12.2.3 Authorization

12.2.3.1 Capability-based Access Control System

Due to heterogeneous nature of IoT devices and networks, most of recent access control proposals have been designed through centralized approaches in which a central entity or gateway is responsible for managing the corresponding authorization mechanisms, allowing or denying requests from external entities. Since this component is usually instantiated by unconstrained entities or back-end servers, standard access control technologies are directly applied. However, significant drawbacks arise when centralized approaches are considered on a real IoT deployment. On the one hand, the inclusion of a central entity for each access request clearly compromises end-to-end security properties, which are considered as an essential requirement [140][141][142] on IoT, due to the sensitivity level of potential applications. On the other hand, the dynamic nature of IoT scenarios with a potential huge number of devices complicates the trust management with the

central entity, affecting scalability. Moreover, access control decisions do not consider contextual conditions which are locally sensed by end devices.

These issues could be addressed by a decentralized approach, in which IoT devices (e.g. smartphones, sensors, actuators, etc.) are enabled with authorization logic without the need to delegate this task to a different entity when receiving an access request. In this case, end devices are enabled with the ability to obtain, process and transmit information to other entities in a protected way. However, in a fully distributed approach, the feasibility of the application of traditional access control models, such as RBAC or ABAC, has not been demonstrated so far. Indeed, as previously mentioned, such models require a mutual understanding of the meaning of roles and attributes, as well as complex access control policies, which makes challenging the application of them on IoT devices. Moreover, the impact of the potential applications of IoT in all aspects of our lives is shifting security aspects from an enterprise-centric vision to a more user-centric one. Therefore, usability is a key factor to be considered, since untrained users should be able to control how their devices and data are shared with other users and services.

As already mentioned, DCapBAC has been postulated as a feasible approach to be deployed on IoT scenarios even in the presence of devices with tight resource constraints. Inspired by SPKI Certificate Theory and ZBAC foundations, it is based on a lightweight and flexible design and that allows authorization functionality is embedded on IoT devices, providing the advantages of a distributed security approach for IoT in terms of scalability, interoperability and end-to-end security. The key element of this approach is the concept of capability, which was originally introduced by [126] as "token, ticket, or key that gives the possessor permission to access an entity or object in a computer system". This token is usually composed by a set of privileges which are granted to the entity holding the token. Additionally, the token must be tamper-proof and unequivocally identified in order to be considered in a real environment. Therefore, it is necessary to consider suitable cryptographic mechanisms to be used even on resource-constrained devices which enable an end-to-end secure access control mechanism. This concept is applied to IoT environments and extended by defining conditions which are locally verified on the constrained device. This feature enhances the flexibility of DCapBAC since any parameter which is read by the smart object could be used in the authorization process. DCapBAC will be part of the access control system and extended with an XACML in order to infer the access control privileges to be embedded into the capability token.

12.2.3.1.1 Capability Token

The format of the capability token is based on JSON. Compared to more traditional formats such as XML, JSON is getting more attention from academia and industry in IoT scenarios, since it is able to provide a simple, lightweight, efficient, and expressive data representation, which is suitable to be used on constrained networks and devices. Capability token is discussed in detail in Section 5.1.2.2.

12.2.3.1.2 DCapBAC scenario

In a typical DCapBAC scenario, an entity (subject) tries to access a resource of another entity (target). Usually, a third party (issuer) generates a token for the subject specifying which privileges it has. Thus, when the subject attempts to access a resource hosted in the target, it attaches the token which was generated by the issuer. Then, the target evaluates the token granting access to the resource. Therefore, a subject which wishes to get access to certain information from a target, requires sending the token together with the request. Thus, the target device that receives such a token can know the privileges (contained in the token) that the subject has, and it can act as a Policy Enforcement Point (PEP). This simplifies the access control mechanism, and it is a relevant feature

on IoT scenarios since complex access control policies are not required to be deployed on end devices. A detailed description of this scenario is presented already in the Section 5.3.4.4.

12.2.3.2 Sequence Diagram for capability token-based approach

The following sequence diagram in Figure 70 shows the whole interaction that must be performed in order to grant access to a user/service to a specific resource provided by a system with a specific API.

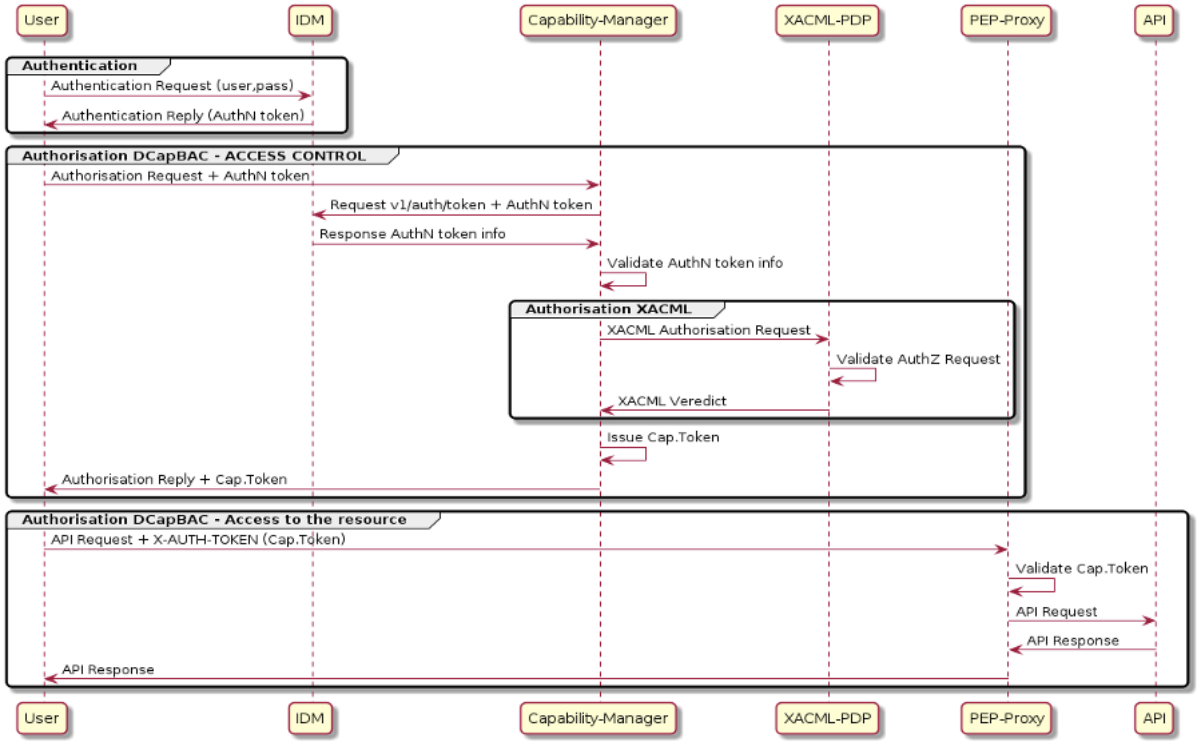


Figure 70: Authorization sequence diagram

As we can see, after an authentication process, the User receives an authentication token, this must be introduced in the authorization query which is addressed to the Capability Manager. This component is responsible for translating this request to an XACML one which is derived to the XACML-PDP component. It validates the XACML authorization query by checking the XACML policy file. If there exists a matching policy, the XACML-PDP will issue a positive verdict which is received by the Capability Manager. This, in turn, generates an authorization token, called Capability Token which is sent to the User.

This Capability Token must accompany the query issued to the system, so that the PEP-Proxy could verify that the query issued by the User is granted. This process is done without querying any other third party, so it is a straightforward task. After validating the query, the PEP-Proxy will forward the query to the system, as well as the corresponding response to the User.

12.2.4 Traceability

12.2.4.1 DEMETER Traceability Component Overview

The authentication and authorization traceability component will log the access to DEMETER resources by logging the issue and use of authentication and authorization tokens. These tokens contain the information about the user who is logged to the system and the resources the user is intended to access.

A permissioned version of blockchain has been chosen to provide the characteristics of immutability, privacy and compatibility required by the DEMETER Traceability Component. It supports both public and private transactions and smart contracts, and their states derived from a single, common, complete blockchain for transactions validated by every node in the network.

The following Figure 71 depicts an overview of the DEMETER Traceability Component and its functional components:

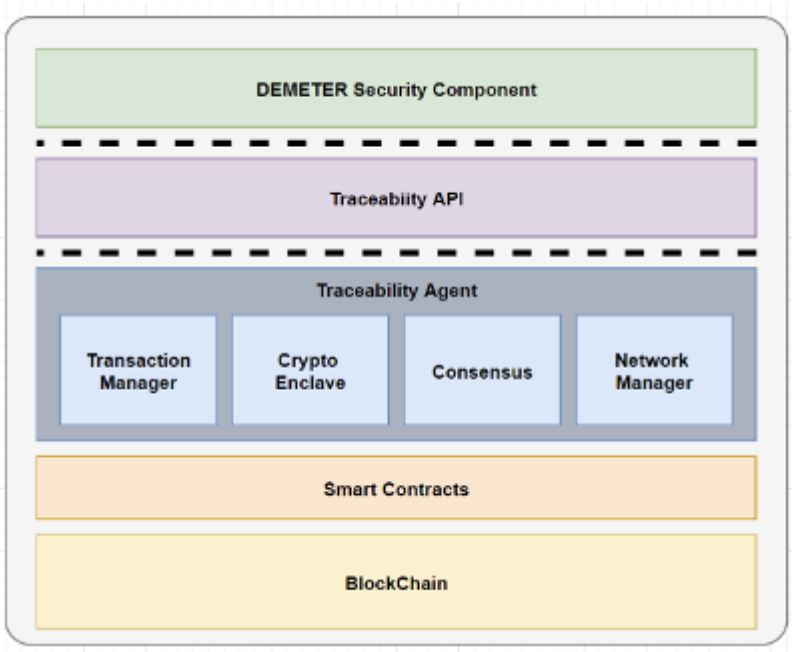


Figure 71: DEMETER Traceability functional components

- **DEMETER Security Component**: the authentication and authorization servers.
- **Traceability API**: an interface to register authentication and authorization events and retrieve their details.
- **Transaction Agent**: is responsible for Transaction privacy, allows access to encrypted transaction data for private transactions, manages local data store and communication with other transaction managers.
- **Crypto Enclave**: responsible for private key management and encryption and decryption of private transaction data
- **Consensus based on Istanbul BFT**: three phase consensus, better fault tolerance and self-verifiable blocks.
- **Network manager**: controls access to the network, enabling a permissioned network of nodes to be created.
- **Smart Contracts**: where the business logic of the Demeter Traceability Component will be defined.
- **Blockchain**: a permissioned blockchain platform such as Quorum, Corda or Hyperledger Fabric.

As an example of how the transaction are processed in a permissioned blockchain, the following Figure 72 shows the Quorum transaction sequence diagram:

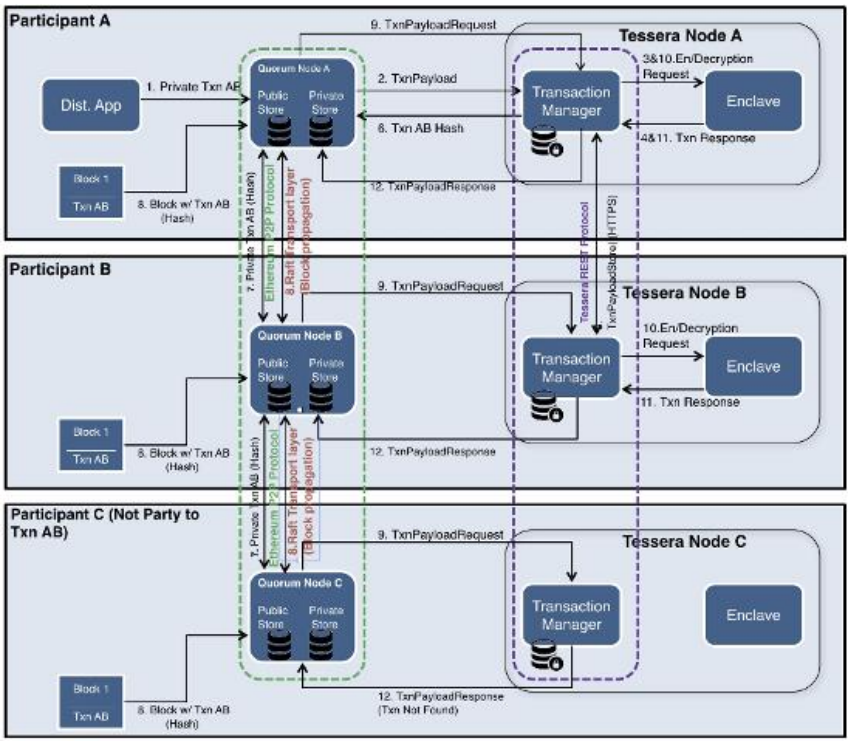


Figure 72: Quorum transaction sequence diagram

12.2.4.2 DEMETER Traceability Component Data Model

Permissioned blockchains register the transactions across the nodes of the network. The DEMETER Traceability Component will register authentication and authorization events as transactions 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.

12.2.5 Confidentiality

Data protection is a relevant aspect for the Demeter project. The prevention of readings by unauthorized users is a fundamental aspect for the robustness of the security system. The Transport Layer Security protocol is an IETF standard designed to increase security in communication between network entities providing both confidentiality and integrity of data [9].

TLS works on two phases:

- Handshake
- Securing messages

12.2.5.1 Handshake

The handshake phase provides authentication through a definition of an authentication algorithm, and the negotiation for the encryption protocol, the key-exchange protocol, the and MAC. These steps are carried out as follows in Figure 73:

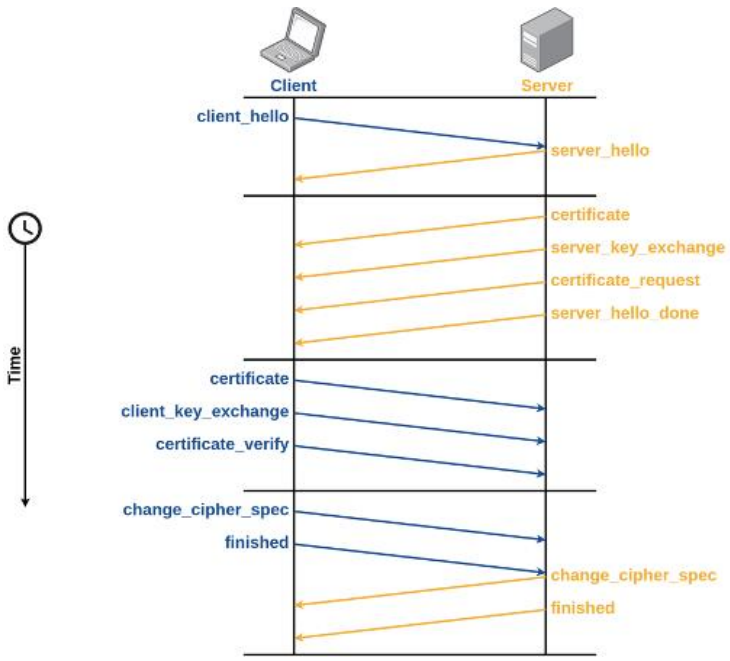


Figure 73: TLS Handshake sequence diagram

The client_hello message contains:

- version number
- optional session ID, to resume a previous session
- list of cipher suites supported, which includes key exchange algorithm, symmetric algorithm and MAC algorithm

The server_hello message is sent in response to the client_hello and contains:

- version number
- optional session ID, to resume a previous session
- the cipher suites to be used, which include key exchange algorithm, symmetric algorithm and MAC algorithm

The certificate message contains the digital certificate used to match a public key and the identity of the owner.

- The *ServerKeyExchange* is used for the key exchange. Its meaning depends on the cipher suite chosen and it is necessary if no public key is sent in the certificate.
- The *ServerHelloDone* message marks the end of the server's steps in the handshake, it does not contain any other information.
- *ClientKeyExchange* contains the client part in the key agreement. As well as for the ServerKeyExchange, the exact format depends on the exchange algorithm agreed previously.
- The *ChangeCipherSpec* message just indicates that from this point the communication is encrypted.
- *Finished* is an encrypted message that marks the end of the handshake.
- *ChangeCipherSpec* and Finished play the same role as the client's message, thus indicating that from this point the communication is encrypted also on the server side and marks the end of the handshake.

12.2.5.2 Securing message

After the handshake is complete, the client and the server start exchanging encrypted messages.

12.3 Implementation (including interfaces)

12.3.1 Security architecture

This part is already mentioned in the section 12.2.1.

12.3.2 Authentication

The FIWARE Keyrock Identity Manager (IdM) API specifications comply with 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

12.3.2.1 IDs within Keyrock

IDs and tokens within Keyrock are generally universally unique identifiers (UUID). The following Table 10 provides a description of the referenced Keyrock IDs and a sample value:

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

Table 10:UUID within Keyrock[181]

The FIWARE Keyrock API provides functionality to manage:

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

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

12.3.2.2 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 endpoints used to obtain, delete and get token details, along with the http verb to be used, are described in the following Table 11:

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

Table 11: API functionality of IDM

12.3.2.3 Applications

In order manage the application that are registered to the IdM and are allowed to get access to DEMETER resources, the following endpoints are provided in Table 12:

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

Table 12: IdM Endpoints

[181] https://fiware-tutorials.readthedocs.io/en/latest/identity-management/index.html

12.3.2.4 Users

To manage the users that are registered to the IdM, the following endpoints are provided in Table 13:

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

Table 13: IdM user management endpoints

12.3.2.5 Roles

In order to manage the roles that are defined for an application, the following REST API endpoints are provided in Table 14:

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

Table 14: Role management REST API endpoints

12.3.2.6 Organization

In order manage the organizations that are registered to the IdM and are allowed to get access to DEMETER resources, the following endpoints are provided in Table 15:

Functionality	**Endpoint**	**Http Verb**
List organizations	http://keyrock/v1/organizations	Get

Functionality	Endpoint	Http Verb
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

Table 15: Organization management API endpoints

12.3.2.7 Relationships between Applications, Organizations, Users and Roles

The IdM offers a series of endpoints in order to create relationships between the Applications, Organization, Users and Roles and manage those relationships. The following Table 16 depicts the 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

Functionality	Endpoint	Http Verb
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

Table 16: Relationships between Applications, Organizations, Users and Roles in IdM

12.3.3 Authorization

Since the authorization enabler comprises the Capability Manager, the XACML framework (PDP and PAP), as well as the PEP_Proxy, below you can find a list of endpoints provided by them.

12.3.3.1 Interfaces definition

The main API functions to realize the access control mechanisms are:

- **Capability Manager**

```
-       Request: KeyRock token + device + action + resource (body)


POST / HTTP/1.1

Host:  <HOST-CAPMAN>:<PORT-CAPMAN>

Content-Type: application/json

{"token": "<idmToken>","de": "<pepproxy endpoint>","ac": "<action>","re":
"<resource>"}


-       Response: Capability token (body)
```

- **Access to resource (through PEP-Proxy)**

```
-      Request: API resource request to (PEP-Proxy endpoint) + Capability
token (x_auth_token header).

-      Response: API response.
```

In addition to them, other endpoints are provided by the other components so that authorization in order to manage the authorization policies.

- XACML-PAP
    - **addAuthzPolicy** (authz_policy): It adds an authz_policy to a specific policy set.
    - **getAuthzPolicy** (authz_policy_id): (AuthzPolicy) It returns the authz_policy corresponding to a given authz_policy_id.
    - **updateAuthzPolicy** (authz_policy): It updates the authorization policy corresponding to a given authz_policy_id.
    - **deleteAuthzPolicy** (authz_policy_id): It deletes the authorization policy corresponding to a given authz_policy_id.
- XACML-PDP:
    - **make_authorization_decision** (subject, action, resource,) authorization_decision: This function aims to check the permission of a subject to perform an action over a resource. Such process is based on the evaluation of XACML policies.

Finally, in the following paragraphs we also describe other functions which are fundamental for the behavior of the Authorization Enabler

Capability Manager

- ***generate_capabilityToken*** *(subject, action, resource, conditions): capabilityToken*: In case of a "Permit" authorization decision, this function is used to generate a capability token. This contains the privilege associated with the subject, action and resource which were used in the previous function. In addition, a set of contextual conditions can be included in the token, specifying certain access restrictions (e.g. regarding context or trust values) to be locally verified at the target device (producer)

PEP_Proxy

- ***enforce_authorization_decision*** *(subject, action, resource, authz_token, context): Permit/Deny*: This function is used to check the validity of an authorization token. Specifically, it should check at least:
    - The token is valid and have not expired
    - The requested action matches a specific privilege in the token
    - Conditions are fulfilled
    - Cryptographic verification (e.g. issuer's signature)

12.3.3.2 Access Control Policies

According to the previous description, the authorization enabler is based on the usage of XACML policies for making authorization decisions. The following

Figure 74 shows an example of XACML policy to be deployed by the DEMETER authorization enabler. Firstly, using the *Target* element, it is indicated that the policy will be only applied for access control requests which are intended to obtain the *"CartagenaTemperatureSensor"*. The policy has a *"PERMIT" Rule* requiring the subject *"jamartinez@odins.es"*.

```
<Policy "Demeter_example">
        <Target>
        <Resource>
                <AttributeValue DataType="String">CartagenaTemperatureSensor</AttributeValue>
                <ResourceAttributeDesignator DataType="String" AttributeId="resource-id" />
        </Resource>
        <Action>
                <AttributeValue DataType="String">get</AttributeValue>
                <ActionAttributeDesignator DataType="String" AttributeId=" action-id "
        </Action>
        </Target>
        <Rule "Permit">
        <Target>
                <Subject>
                <AttributeValue DataType="String">jamartinez@odins.es</AttributeValue>
                <SubjectAttributeDesignator AttributeId="subject-id" DataType="String"/>
                </Subject>
        </Target>
        </Rule>
        <Rule "Deny">
        </Rule>
</Policy>
```

Figure 74: XACML policy example for the Demeter authorization enabler

12.3.4 Traceability

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 17:

Functionality	Endpoint	Http Verb
Register event	http://audit_tool /v1/send	Post
Get event details	http://audit_tool/v1/transaction/{hash}	Get

Table 17: DEMETER Traceability Component endpoints

The "***send***" endpoint will register at the permissioned blockchain the transaction fields described at 11.2.4.2. 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 "**transaction**" endpoint allows us to retrieve the details of a registered transaction using the transaction ID (hash). It will return a 200 value and a payload with the transaction details.

12.3.5 Confidentiality

The confidentiality module will be implemented with OpenSSL[182] which is a well-known toolkit written in C that provides several libraries and APIs to perform some cryptographic tasks. *libssl* is the specific OpenSSL library for TLS and SSL protocol applications.

12.3.5.1 Client-server

The implementation of OpenSSL requires an installation of the component into devices, both server and client side. To configure the *libssl* library, OpenSSL uses a custom build system which allows the library to set up the recursive makefiles. Once the configuration is successful, the library will be built through a C compiler.

There are also some codes that perform an SSL/TLS client or server, which can be used for a first and simple implementation. Starting from these, ad-hoc configurations can be developed for specific requirements.

12.3.5.2 Handshake

The initial handshake can provide server authentication, client authentication or no authentication at all. This handshake phase generates a master secret, from which the secret key is derived. In OpenSSL this master_secret is kept within the SSL Session *SSL_SESSION*.

12.3.5.3 Cipher suites

OpenSSL, and TLS in general, permits to use several cryptographic protocols:

- for key exchange: RSA[183], Diffie-Hellman[184], ECDH[185], SRP[186], PSK[187]
- for symmetric-key cryptography: RC4[188], DES[189], Triple DES[190], AES[191], IDEA[192], Camellia[193]
- for authentication: RSA, DSA, ECDSA
- for integrity function: HMAC-MD5[194], HMAC-SHA[195]

[182] https://www.openssl.org/
[183] https://it.wikipedia.org/wiki/RSA_(crittografia)
[184] https://it.wikipedia.org/wiki/Diffie-Hellman
[185] https://it.wikipedia.org/w/index.php?title=Elliptic_Curve_Diffie-Hellman&action=edit&redlink=1
[186] https://it.wikipedia.org/w/index.php?title=Secure_remote_password&action=edit&redlink=1
[187] https://it.wikipedia.org/wiki/Pre-Shared_Key
[188] https://it.wikipedia.org/wiki/RC4
[189] https://it.wikipedia.org/wiki/Data_Encryption_Standard
[190] https://it.wikipedia.org/wiki/Triple_DES
[191] https://it.wikipedia.org/wiki/Advanced_Encryption_Standard
[192] https://it.wikipedia.org/wiki/International_Data_Encryption_Algorithm
[193] https://it.wikipedia.org/wiki/Camellia_(cifrario)

This suite is negotiated during the handshake phase between server and client.

12.3.5.4 Secure server-client implementation

Since OpenSSL is installed, a certificate needs to be generated. During the handshake phase, the server sends to the client several data, including its own digital certificate. If the client is requesting a server resource that requires client authentication, requests the client's digital certificate.

The certificate can be generated using the OpenSSL shell prompt. This generation includes different choices about the certificate, such as the type, the validity and the key. A previously key generation is mandatory to generate the certificate.

To run the server and the client SSL/TLS protocols just compile the C codes that provide the server and client functionalities. These codes can be made up starting from the OpenSSL examples.

Once the client and server are running, the connection should be checked. This can be made thanks to OpenSSl *s_client* tool. The tool can be launched from the OpenSSL shell prompt with a proper command that includes a hostname and port.

Once the command is typed, it can be possible to see a lot of diagnostic output or type, for example, an HTTP request. This operation confirms that the TLS communication layer is working: a connection has been established with the HTTP server, a request has been sent followed by a response back.

12.3.5.5 Objects encryption and decryption

The module will be able, in order to facilitate the encryption of part of the messages, to translate plain text JSON and XML objects, to encrypted objects such as JSE and XML-Enc.

Using python libraries such as *objcrypt* (JSON AES CBC 32 Block Encryption) and *xmlsec* (XML security library), the encryption and the decryption procedures can be implemented either as standalone scripts or in docker containers.

Example:

```
$ python3 JSON\ test_encrypt.py
JSON: {'test': 'test value'}
Encoded JSON: {"test": "HPYVd5NHhFd7yMUqoYEGEM9SZ8pE4wI1Reksnh5ZGv2FTtUEBX0bs0W2OENWhUpn"}

$ python3 JSON\ test_decrypt.py
Decoded JSON: {"test": "test value"}
```

13 Conclusions

This deliverable describes in detail the initial release of the DEMETER Data and Knowledge extraction tools, which is the outcome of tasks 2.2 (Data Management and Integration), 2.3 (Targeted data fusion, analytics and knowledge extraction) and 2.4 (Data Protection, Privacy, Traceability and Governance Management) of Work Package 2. The main results and on-going work presented in this deliverable are related to the implementation of the following DEMETER enablers: Data Preparation & Integration, Data Analytics & Knowledge Extraction, Data Management and Data Security & Governance. The first three realize the Data & Knowledge (DK) enablers, which are part of the

[194] https://it.wikipedia.org/wiki/HMAC
[195] https://it.wikipedia.org/wiki/HMAC

DEMETER advanced enablers, and which rely and use the DEMETER DK Repository that is based on the DEMETER AIM. The latter enabler provides services to both core and advanced enablers, regarding authentication, authorization, privacy and traceability.

The deliverable presented an analysis of the state of the art related to the approaches, methods and techniques, as well as an overview of existing tools relevant to the implementation of the abovementioned enablers. Subsequently, it presented the technical requirements extracted by Task 2.2, Task 2.3 and Task 2.4, which drive the design and implementation of the Data & Knowledge, and Data Security enablers. After that, the deliverable presented the place, role and relationships of these components in the general DEMETER architecture, and continued with a detailed description of the enablers. This included the description of the approach, design and on-going implementation work, including API design, of i) the Data Preparation & Integration enabler with underlying facilities and pipelines based on Linked Data; ii) the Data Management enabler realized via the Brokerage Service Environment and DEMETER Enabler Hub; iii) the Data Analytics & Knowledge Extraction enabler comprising Data Quality facilities for the assessment of Linked Data and Tabular Data, and Data Analytics & Fusion facilities for targeted analytics, fusion, training, label acquisition, algorithm selection and DSS integration; iv) data security enabler with its facilities for authentication, authorization, traceability and confidentiality. Finally, the document concludes with a summary, discusses the ongoing activities related to Data & Knowledge enablers that aim to support the first round of the DEMETER pilots and the future plans that are in place in view of the final release of these enablers. Additionally, Annex A presents in tabular form the mapping of requirements presented in Section 6 with the components presented in the components Sections (Sections 8-12), and Annex B includes the authentication endpoints documentation, including examples of http requests to the IdM Endpoints that provide authentication functionalities.

The content of this deliverable is the result of collaborative work of all partners involved in WP2, but also includes input and feedback from WPs 3, 4 and 5. For instance, the data management components, which are tightly connected with WP3, were discussed and presented including input from the WP3 perspective. The Data Analytics & Knowledge Extraction enabler also includes input and feedback from WP4 regarding the decision support & monitoring enablers. The Data Security components have also a tight connection with WP3, as they realize a horizontal service in the DEMETER Reference Architecture, supporting core and advanced DEMETER enablers. Additionally, the work in this deliverable is highly connected and relies on the results of D2.1 regarding the implementation of DEMETER AIM. The Data & Knowledge enablers rely on and use the DK repository that carries any data or extracted knowledge in AIM format. For instance, the Data Preparation & Integration enabler, transforms data into the AIM format or translates queries using AIM terms. Similarly, the data generated by the enablers, e.g., knowledge extracted, data quality information, is also represented according to AIM model.

This deliverable contributes to the achievement of Milestone 2 (DEMETER Enablers, Hub, Spaces and Applications Release 1) planned for June 2020.

Hereafter, we present some ongoing and future work that is scheduled to be completed for the final release of the DEMETER Data and Knowledge extraction tools (that will be presented in D2.4 next year).

Regarding Data Management, this document is designed to provide guidelines for the data management system in the DEMETER context. Attention has focused on various crucial aspects, such as the formal definition of the main components (that will deal with data care), monitoring and control of some fundamental criteria such as availability and reliability (to be taken into consideration when choosing technologies which will be used for the implementation phase), multi-tenancy criteria (which will greatly influence the APIs characteristics and the internal interaction mechanisms of the data management components).

The first iteration, which in some respects has been defined in this document, concerns the functional design of the components, while the second, in the near future, will mainly concern aspects related to the definition of the functionality of the enablers responsible for data management in DEMETER (it should be remembered that many details on these modules will also be added in D3.2) and to the development of the components that will have to meet the criteria of conformity of these modules to the needs of the Pilots.

Nevertheless, this process is not easy to achieve and consolidate since there are still some technical details to be discussed and resolved. In particular, the large extension of the data produced by the devices, as well as the numerous protocols used and the complexity of the DEMETER network (also linked to the cloud and IoT context), makes this process more difficult. Hence, a constant alignment must be foreseen both at the organizational and at the technical level among the main partners involved but also and above all at the pilots' level.

Regarding the Data Preparation and Integration, the pipelines already available and tested in previous projects constitute the basis for the implementation of this enabler. However, as part of DEMETER, these pipelines will have to be adapted/extended, new pipelines will need to be created and a DEMETER API will have to be implemented. In particular, the following main tasks will have to be carried out in DEMETER:

- Implement and use DEMETER AIM as the target model to represent data in the transformations/translations
- Existing pipelines will require adjustments and extensions, including updates in the mapping specifications and wrapper implementations. Additionally, existing pipelines will have to be extended, and new pipelines will have to be created in order to support the processing and integration of the different DEMETER datasets. For instance, new data collection methods and pre/post processing services will have to be implemented, new wrappers and mappings will have to be generated, and new tools will have to be integrated.
- The pipelines and their underlying facilities will comprise a data preparation & integration enabler in DEMETER, which will expose a DEMETER-enabled API allowing to reuse and automatize the provided functionalities. So, while the pipelines use many different tools/services via different interfaces and protocols (API, CLI) to prepare and integrate data, the DEMETER API will abstract these differences and provide a single and common access point to these facilities. The API will also provide access to data based on AIM model, abstracting complexities in using SPARQL queries, at least for some basic/common use case scenarios.
- Alignment with pilot needs evolving out of WP4 and WP5 results

Regarding the Analytics and Knowledge Extraction tools, future work concerning Data Quality facilities is related to the gathering of concrete data quality requirements and their implementation, either using the generic quality assessment methods or, based on the requirements, implement

additional custom methods. Regarding the Data Analytics facilities, future work is concerned with the full integration of planned 3rd party software components and the AIM data model, as well as production-grade integration of the enabler with related Core Enablers and Decision Support & Performance Monitoring enablers. In the future, the Data Analytics and Knowledge enabler shall be further aligned with pilot requirements evolving out of WP4 and WP5 results.

Regarding the Data Security components, DEMETER Reference Architecture shows the authentication, authorization and communication components functionalities, and how they can be integrated with other DEMETER components (such the DEMETER Dashboard and the BSE) and pilots.

The next steps for the data protection, privacy and traceability components and enabler will be to provide support to the pilots and other DEMETER components on how to integrate the security enablers in order to easily create secured communication channels and make use of the authentication and authorization capabilities.

The security components and enablers will be therefore further integrated and tested, and they will be fine-tuned based on the DEMETER components and pilot's needs.

Finally, we are working closely with Task 2.1 in order to extend the AIM to cover all needs arising during the implementation of these enablers, including the support of more specific pilot needs, and the communication with and between enablers.

In a year from now, the revised version of the DEMETER Data and Knowledge extraction tools will be released in D2.4 to be delivered in April 2021..

14 References

[1] Wood, D., Lanthaler, M. & Cyganiak, R. (2014). RDF 1.1 Concepts and Abstract Syntax [W3C Recommendation]. (Technical report, W3C)

[2] Harris, S. and Seaborne, A. (2013). SPARQL 1.1 Query Language. W3C Recommendation. W3C.

[3] Hyland, B., Atemezing, G., Villazón-Terrazas, B. (2014). Best Practices for Publishing Linked Data. W3C Working Group Note 09 January 2014. URL: https://www.w3.org/TR/ld-bp/

[4] Heath, T. and Bizer, C. (2011) Linked Data: Evolving the Web into a Global Data Space (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool.

[5] W3C OWL Working Group (2012). OWL 2 Web Ontology Language Document Overview (Second Edition) - W3C Recommendation 11 December 2012 {World Wide Web Consortium (W3C)}

[6] Bikakis, Nikos & Tsinaraki, Chrisa & Gioldasis, Nektarios & Stavrakantonakis, Ioannis & Christodoulakis, Stavros. (2013). The XML and Semantic Web Worlds: Technologies.

[7] Echterhoff et al. (2018) , OGC Testbed-14: Application Schemas and JSON Technologies Engineering Report, http://www.opengis.net/doc/PER/t14-D022-2

[8] Svensson, Atkinson,Car (Eds). (2019) ,Content Negotiation by Profile (W3C working draft)

[9] Atkinson, Car (Eds). (2019), Profile Descriptions ontology. (W3C working draft)

[10] Albertoni, R & Isaac, A (2019), Introducing the Data Quality Vocabulary (DQV), Draft under review http://www.semantic-web-journal.net/system/files/swj2320.pdf

[11] For the basics of how ODRL enables usage control in a data space, see C. Jung, A. Eitel. “Data Usage Control”. Webinar, 2019. (https://www.internationaldataspaces.org/wp-content/uploads/2019/07/Data-usage-control.pdf)

[12] Sandhu, R. S. (1993). Lattice-based access control models. Computer, 26(11), 9-19

[13] Moffett, J., Sloman, M., & Twidle, K. (1990). Specifying discretionary access control policy for distributed systems. Computer Communications, 13(9), 571-580.

[14] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. Computer, 29(2), 38-47.

[15] Vincent C. et al. NIST Special Publication 800-162. Guide to Attribute Based Access Control (ABAC) Definition and Considerations http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf

[16] From ABAC to ZBAC: The Evolution of Access Control Models http://www.hpl.hp.com/techreports/2009/HPL-2009-30.pdf

[17] Hernández-Ramos, J. L., Jara, A. J., Marín, L., & Skarmeta Gómez, A. F. (2016). DCapBAC: embedding authorization logic into smart things through ECC optimizations. International Journal of Computer Mathematics, 93(2), 345-366.

[18] Vassiliadis P., Simitsis A. (2009) Extraction, Transformation, and Loading. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA

[19] Semantic Integration and Interoperability among Portals - http://what-when-how.com/portal-technologies-and-applications/semantic-integration-and-interoperability-among-portals/

[20] Semantic Integration & Interoperability Using RDF and OWL. W3C Editor's Draft 3 November 2005. https://www.w3.org/2001/sw/BestPractices/OEP/SemInt/

[21] JSON-LD 1.1 A JSON-based Serialization for Linked Data. W3C Candidate Recommendation 12 December 2019.

[22] Hitzler P, Krotzsch M, Rudolph S. Foundations of Semantic Web Technologies. Chapman and Hall/CRC; 2009.
[23] Villazón-Terrazas, B., et al.. Methodological Guidelines for Publishing Government Linked Data. In Linking Government Data, chapter 2, pages 27–49. Springer New York, New York, NY, 2011. URL: http://link.springer.com/chapter/10.1007/978-1-4614-1767-5_2
[24] Hyland, B. and Wood, D. (2011). The Joy of Data - A Cookbook for Publishing Linked Government Data on the Web. Linking Government Data. New York, NY, United States: Springer New York.3-26.https://doi.org/10.1007/978-1-4614-1767-5_1
[25] Linked Data Applied: A Field Report from the Netherlands. SEMANTiCS Conference 2019. https://2019.semantics.cc/linked-data-applied-field-report-netherlands
[26] Why Linked Data for data.gov.uk? https://www.jenitennison.com/2010/01/26/why-linked-data-for-data-gov-uk.html
[27] Australian Government Linked Data Working Group https://agldwg.github.io/website/
[28] Spanish LinkedData.es http://linkeddata.es/index.html
[29] FOODIE. D2.2.3 Service Platform Specification. Miguel Ángel Esbrí. 2016
[30] Palma R., Reznik T., Esbri M., Charvat K., Mazurek C., An INSPIRE-based vocabulary for the publication of Agricultural Linked Data. Proceedings of the OWLED Workshop: collocated with the ISWC-2015, Bethlehem PA, USA, October 11-15, 2015
[31] Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. Journal on Semantic Web and Information Systems (in press), 2009.
[32] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ull-man, and J Widom. The TSIMMIS project: Integration of heterogeneous information sources. InProceedings of the 10th Meeting of the Information Processing Society ofJapan, pages 7–18, Tokyo, Japan, October 1994
[33] A.Y. Levy, A. Rajaraman, and J.J. Ordille. Querying heterogeneous information sources using source descriptions. InProceedings of the Twenty-second InternationalConference on Very Large Data Bases (VLDB'96), pages 251–262, Mumbai (Bom-bay), India, September 1996.
[34] Andriy Nikolov, Peter Haase, Johannes Trame, and Artem Kozlov. 2017. Ephedra: Efficiently Combining RDF Data and Services Using SPARQL Federation KESW 2017. 246--262
[35] Liao, Yongxin, et al. "Semantic annotation model definition for systems interoperability." On the Move to Mean-ingful Internet Systems: OTM 2011 Workshops. Springer Berlin Heidelberg, 2011.
[36] Lohmann, S., Díaz, P., Aedo, I.: MUTO: The Modular Unified Tagging Ontology. Proceedings of the 7th International Conference on Semantic Systems (I-SEMANTICS 2011), pp. 95-104.
[37] R. Navigli and S. Ponzetto. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence, 193, Elsevier, 2012, pp. 217-250
[38] Unal, O., & Afsarmanesh, H. (2009). Schema matching and integration for data sharing among collaborating organizations. Journal of Software, 4(3), 248-261. https://doi.org/10.4304/jsw.4.3.248-261
[39] Rahm, E. & Bernstein, P (2001). "A survey of approaches to automatic schema matching". The VLDB Journal 10, 4.
[40] Unal, O., Afsarmanesh, H. Semi-automated schema integration with SASMINT. Knowl Inf Syst 23, 99–128 (2010) doi:10.1007/s10115-009-0217-z

[41] W. Li and C. Clifton, "SEMINT: A tool for identifying attribute correspondence in heterogeneous databases using neural networks", Journal of Data and Knowledge Engineering, 2000. 33 (1), pp. 49-84.
[42] J. Madhavan, P.A. Bernstein, and E. Rahm, "Generic Schema Matching with Cupid", Proc. of Very Large Data Bases, 2001.
[43] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, "S-match: an algorithm and an implementation of semantic matching", Proc. of ESWS, 2004.
[44] D. Aumueller, H.H. Do, S. Massmann, and E. Rahm, "Schema and ontology matching with COMA++", Proc. of SIGMOD, 2005.
[45] Pavel Shvaiko, Jérôme Euzenat. Ontology matching: state of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering, Institute of Electrical and Electronics Engineers, 2013, 25 (1), pp.158-176. ff10.1109/TKDE.2011.253f
[46] Euzenat, J., & Shvaiko, P. (2007), Ontology matching. Springer, 2007.
[47] Euzenat, J., & Shvaiko, P. (2013). Ontology matching (2nd ed.). Heidelberg (DE): Springer-Verlag. URL:http://book.ontologymatching.org
[48] Otero-Cerdeira, L., Rodríguez-Martínez, F.J., & Gómez-Rodríguez, A.M. (2015). Ontology matching: A literature review. Expert Syst. Appl., 42, 949-971.
[49] J. Euzenat, "Alignment infrastructure for ontology mediation and other applications," in Proc. International Workshop on Mediation in Semantic Web Services (MEDIATE), 2005, pp. 81–95.
[50] Cruz, I. F., Antonelli, F. P., Stroe, C., Keles, U. C., & Maduko, A. (2008). Using agreementmaker to align ontologies for oaei 2009: Overview, results, and outlook. In P. Shvaiko, J. Euzenat, F. Giunchiglia, H. Stuckenschmidt, N. F. Noy, & A. Rosenthal (Eds.), OM, CEUR-WS.org. URL: <http://dblp.uni-trier.de/db/conf/semweb/om2009.html#CruzASKM08>.
[51] Wang, Z., Zhang, X., Hou, L., Zhao, Y., Li, J., Qi, Y., & Tang, J. (2010). RiMOM results for OAEI 2010. In P. Shvaiko, J. Euzenat, F. Giunchiglia, H. Stuckenschmidt, M. Mao, & I. F. Cruz (Eds.) OM, CEUR-WS.org. (pp. 195–202).
[52] David, J. (2011). AROMA results for OAEI 2011. In P. Shvaiko, J. Euzenat, T. Heath, C.Quix, M. Mao, & I. F. Cruz (Eds.), OM, CEUR-WS.org. (pp. 122–125).
[53] Maßmann, S., Raunich, S., Aumüller, D., Arnold, P., & Rahm, E. (2011). Evolution of the COMA match system. In P. Shvaiko, J. Euzenat, T. Heath, C. Quix, M. Mao, & I.F. Cruz (Eds.), OM, CEUR-WS.org. (pp. 49–60).
[54] Aumueller, D., Do, H., Massmann, S., and Rahm, E. Schema and ontology matching with COMA++. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data (SIGMOD '05). Association for Computing Machinery, New York, NY, USA, 2005 906–908. DOI:https://doi.org/10.1145/1066157.1066283.
[55] Quix, C., Gal, A., Sagi, T., & Kensche, D. (2010). An integrated matching system GeRoMeSuite and SMB: results for OAEI 2010. In P. Shvaiko, J. Euzenat, F. Giunchiglia, H. Stuckenschmidt, M. Mao, & I. F. Cruz (Eds.), OM, CEUR-WS.org. (pp. 166–171).
[56] Charvat K., Reznik T., Lukas V., Charvat Junior K., Jedlicka K., Palma R., Berzins R. Advanced Visualization of Big Data for Agriculture as part of DataBio Development. Proceedings of the 2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2018. Valencia, Spain
[57] T. Reznik, K. Charvat jr., K. Charvat, S. Horakova, V. Lukas, M. Kepka, "Open Data Model for (Precision) Agriculture Applications and Agricultural Pollution Monitoring," In Proceedings of

Enviroinfo and ICT for Sustainability 2015, ACSR-Advances in Computer Science Research, 22, Atlantis Press, Paris, France, pp. 97-107
[58] T. Van Hertem, D. Berckmans, "Appropriate data visualization is key to Precision Livestock Farming acceptance.
[59] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (January 2008), 107-113. DOI=http://dx.doi.org/10.1145/1327452.1327492
[60] A survey of open source tools for machine learning with big data in the Hadoop ecosystem, Sara Landset, Taghi M. Khoshgoftaar, Aaron N. RichterEmail author and Tawfiq Hasanin. Journal of Big Data 2015 DOI: 10.1186/s40537-015-0032-1
[61] Hall, D., McCool, C., Dayoub, F., Sunderhauf, N., & Upcroft, B. (2015). Evaluation of Features for Leaf Classification in Challenging Conditions. 2015 IEEE Winter Conference on Applications of Computer
[62] Lee, S. H., Chan, C. S., Wilkin, P., & Remagnino, P. (2015). Deep-plant: Plant identification with convolutional neural networks. 2015 IEEE International Conference on Image Processing (ICIP).
[63] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Computational Intelligence and Neuroscience, 2016, 1–11.
[64] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 7.
[65] Reyes, A.K., Caicedo, J.C., Camargo, J.E., 2015. Fine-Tuning Deep Convolutional Networks for Plant Recognition. CLEF (Working Notes), Toulouse
[66]Kuwata, K., & Shibasaki, R. (2015). Estimating crop yields with deep learning and remotely sensed data. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS).
[67] Rußwurm, M., Körner, M., 2017. Multi-temporal land cover classification with long short term memory neural networks. Int. Arch. Photogramm., Remote Sens. Spatial Inform. Sci. 42.
[68] Rahnemoonfar, M., Sheppard, C., 2017. Deep count: fruit counting based on deep simulated learning. Sensors 17 (4), 905.
[69] McCool, C., Perez, T., & Upcroft, B. (2017). Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics. IEEE Robotics and Automation Letters, 2(3), 1344–1351.
[70] Milioto, A., Lottes, P., Stachniss, C., 2017. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. Proceedings of the International Conference on Unmanned Aerial Vehicles in Geomatics. Bonn, Ger
[71] Potena, C., Nardi, D., Pretto, A., 2016. Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In: International Conference on Intelligent Autonomous Systems. Springer, Cham, Shanghai, China, pp. 105–121.
[72] Rebetez, J. et al., 2016. Augmenting a convolutional neural network with local histograms—a case study in crop classification from high-resolution UAV imagery. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learn
[73] Sehgal, G., Gupta, B., Paneri, K., Singh, K., Sharma, G., Shroff, G., 2017. Crop Planning using Stochastic Visual Optimization. arXiv preprint arXiv: 1710.09077.
[74] Namin, S.T., Esmaeilzadeh, M., Najafi, M., Brown, T.B., Borevitz, J.O., 2017. Deep Phenotyping: Deep Learning For Temporal Phenotype/Genotype Classification. BioRxiv, 134205.

[75] Minh, D.H., Ienco, D., Gaetano, R., Lalande, N., Ndikumana, E., Osman, F., Maurel, P., 2017. Deep Recurrent Neural Networks for mapping winter vegetation quality coverage via multi-temporal SAR Sentinel-1. arXiv preprint arXiv: 1708.03694.
[76] Luus, F.P., Salmon, B.P., van den Bergh, F., Maharaj, B.T., 2015. Multiview deep learning for land-use classification. IEEE Geosci. Remote Sens. Lett. 12 (12), 2448–2452.
[77] Dyrmann, M., Karstoft, H., Midtiby, H.S., 2016a. Plant species classification using deep convolutional neural networks. Biosyst. Eng. 151, 72–80.
[78] Amara, J., Bouaziz, B., Algergawy, A., 2017. A Deep Learning-Based Approach for Banana Leaf Diseases Classification. BTW workshop, Stuttgart, pp. 79–88
[79] Grinblat, G.L., Uzal, L.C., Larese, M.G., Granitto, P.M., 2016. Deep learning for plant identification using vein morphological patterns. Comput. Electron. Agric. 127, 418–424.
[80] Douarre, C., Schielein, R., Frindel, C., Gerth, S., Rousseau, D., 2016. Deep learning based root-soil segmentation from X-ray tomography. BioRxiv, 071662.
[81] Santoni, M.M., Sensuse, D.I., Arymurthy, A.M., Fanany, M.I., 2015. Cattle race classification using gray level co-occurrence matrix convolutional neural networks. Procedia Comput. Sci. 59, 493–502.
[82] M. Jhuria, A. Kumar, and R. Borse, “Image processing for smart farming: Detection of disease and fruit grading,” in Proc. IEEE 2nd Int. Conf. Image Inf. Process. (ICIIP), Dec. 2013, pp. 521–526.
[83] B. Suksawat and P. Komkum, “Pineapple quality grading using image processing and fuzzy logic based on Thai agriculture standards,” in Proc. Int. Conf. Control Autom. Robot., May 2015, pp. 218–222.
[84] A. Kapoor, S. I. Bhat, S. Shidnal, and A. Mehra, “Implementation of IoT (Internet of Things) and image processing in smart agriculture,” in Proc. Int. Conf. Comput. Syst. Inf. Technol. Sust. Solutions (CSITSS), Oct. 2016, pp. 21–26.
[85] Gutierrez Jaguey, J., Villa-Medina, J. F., Lopez-Guzman, A., & Porta-Gandara, M. A. (2015). Smartphone Irrigation Sensor. IEEE Sensors Journal, 15(9), 5122–5127.
[86] Pérez-Ortiz, M., Peña, J. M., Gutiérrez, P. A., Torres-Sánchez, J., Hervás-Martínez, C., & López-Granados, F. (2015). A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method. Applied Soft Computing, 37, 533–544
[87] Balducci, F., Impedovo, D., & Pirlo, G. (2018). Machine Learning Applications on Agricultural Datasets for Smart Farm Enhancement. Machines, 6(3), 38. doi:10.3390/machines6030038
[88] M. R. Bendre, R. C. Thool, and V. R. Thool, “Big data in precision agriculture: Weather forecasting for future farming,” in Proc. 1st Int. Conf. Next Gener. Comput. Technol. (NGCT), Sep. 2015, pp. 744–750.
[89] Frelat, R., Lopez-Ridaura, S., Giller, K. E., Herrero, M., Douxchamps, S., Djurfeldt, A. A., van Wijk, M. T. (2015). Drivers of household food availability in sub-Saharan Africa based on big data from small farms. Proceedings of the National Academy of Sciences, 113(2), 458–463. doi:10.1073/pnas.1518384112
[90] Schuster, E., Kumar, S., Sarma, S., Willers, J., & Milliken, G. (2011). Infrastructure for data-driven agriculture: identifying management zones for cotton using statistical modeling and machine learning techniques. 2011 8th International Conference & Expo on Emerging Technologies for a Smarter World.
[91] Garvin, D. A. (1984). What does product quality really mean? Sloan management review, 26(1).

[92] Kan, S. H. (2002). Metrics and models in software quality engineering. Addison-Wesley Longman Publishing Co., Inc.
[93] ISO (2015). ISO/IEC 25024:2015 Measurement of Data Quality.
[94] ISO (2011). ISO /TS 8000-1:2011 Data Quality – Part1: Overview.
[95] Buck, D. (2012). Datenqualität, K.o.-Kriterium für Business Intelligence, Computerwoche. Available: http://www.cowo.de/a/1938325
[96] Eckerson, W. (2005). Data Quality and the Bottom Line: Achieving Business Success through Commitment to High Quality Data. TDWI's Data Quality Report, Chatsworth 2002.
[97] Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. ACM Computing Surveys (CSUR), 41(3), 16.
[98] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, S. Auer. Quality assessment for linked data: A survey. Semantic Web, 7(1):63–93, 2015. 316 citations according to Google Scholar on 22 October 2019.
[99] For the underlying design, see J. Debattista, C. Lange, S. Auer. Representing Dataset Quality Metadata using Multi-Dimensional Views. SEMANTiCS 2014. URL : https://arxiv.org/abs/1408.2468
[100] J. Debattista, S. Auer, C. Lange. Luzzu – A Methodology and Framework for Linked Data Quality Assessment. Data and Information Quality, 8(1), 2016 . URL : https://dl.acm.org/citation.cfm?id=2992786
[101] J. Debattista, S. Londoño, C. Lange, S. Auer. Quality Assessment of Linked Datasets using Probabilistic Approximation. Extended Semantic Web Conference 2015. URL: https://arxiv.org/abs/1503.05157
[102] G. Sejdiu, A. Rula, J. Lehmann, H. Jabeen. A Scalable Framework for Quality Assessment of RDF Datasets. International Semantic Web Conference 2019. URL: http://jens-lehmann.org/files/2019/iswc_dist_quality_assessment.pdf
[103] J. B. Bernabe, A. Skarmeta, N. Notario, J. Bringer, and M. David, "Towards a privacy-preserving reliable European identity ecosystem," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2017.
[104] J. B. Bernabe, M. David, R. T. Moreno, J. P. Cordero, S. Bahloul, and A. Skarmeta, "ARIES: Evaluation of a reliable and privacy-preserving European identity management framework," Futur. Gener. Comput. Syst., 2020.
[105] T. M. Fernández-Caramés, O. Blanco-Novoa, I. Froiz-Míguez, and P. Fraga-Lamas, "Towards an Autonomous Industry 4.0 Warehouse: A UAV and Blockchain-Based System for Inventory and Traceability Applications in Big Data-Driven Supply Chain Management," Sensors (Basel)., vol. 19, no. 10, 2019.
[106] S. Pearson et al., "Are Distributed Ledger Technologies the panacea for food traceability?," Glob. Food Sec., vol. 20, no. November 2018, pp. 145–149, 2019.
[107] J. Lin, A. Zhang, Z. Shen, and Y. Chai, "Blockchain and IoT based food traceability for smart agriculture," ACM Int. Conf. Proceeding Ser., pp. 1–6, 2018.
[108] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in Agri-Food supply chain management: A practical implementation," 2018 IoT Vert. Top. Summit Agric. - Tuscany, IOT Tuscany 2018, pp. 1–4, 2018.
[109] W. Hong, Y. Cai, Z. Yu, and X. Yu, "An Agri-product Traceability System Based on IoT and Blockchain Technology," Proc. 2018 1st IEEE Int. Conf. Hot Information-Centric Networking, HotICN 2018, no. HotICN, pp. 254–255, 2019.

[110] S. Wingreen, R. Sharma, P. Jahanbin, S. Wingreen, and R. Sharma, “A Blockchain Traceability Information System for Trust Improvement in Agricultural Supply Chain,” Res. Pap., pp. 5–15, 2019.
[111] M. Kim, B. Hilton, Z. Burks, and J. Reyes, “Integrating Blockchain, Smart Contract-Tokens, and IoT to Design a Food Traceability Solution,” 2018 IEEE 9th Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON 2018, no. Figure 1, pp. 335–340, 2019.
[112] T. Sermpinis and C. Sermpinis, “Traceability decentralization in supply chain management using blockchain technologies,” pp. 2–9, 2018.
[113] S. Paavolainen and P. Nikander, “Security and privacy challenges and potential solutions for DLT based IoT systems,” 2018 Glob. Internet Things Summit, GIoTS 2018, no. June, pp. 1–6, 2018.
[114] M. L. Pardal and J. Alves Marques, “Cost model for RFID-based traceability information systems,” 2011 IEEE Int. Conf. RFID-Technologies Appl. RFID-TA 2011, pp. 486–493, 2011.
[115] A. Elsts, E. Mitskas, and G. Oikonomou, “Distributed ledger technology and the internet of things: A feasibility study,” BlockSys 2018 - Proc. 1st Blockchain-Enabled Networked Sens. Syst. Part SenSys 2018, pp. 7–12, 2018.
[116] K. R. Özyilmaz and A. Yurdakul, “Work-in-progress: Integrating low-power IoT devices to a Blockchain-Based Infrastructure,” Proc. 13th ACM Int. Conf. Embed. Softw. 2017 Companion, EMSOFT 2017, 2017.
[117] Z. Guan, J. Li, Y. Zhang, R. Xu, Z. Wang, and T. Yang, “An efficient traceable access control scheme with reliable key delegation in mobile cloud computing,” Eurasip J. Wirel. Commun. Netw., vol. 2016, no. 1, 2016.
[118]I. E. T. F. (IETF), “OAuth.” [Online]. URL: https://oauth.net/.
[119] Internet Engineering Task Force (IETF), “The OAuth 2.0 Authorization Framework,” 2012. [Online]. URL: https://tools.ietf.org/html/rfc6749.
[120] “OASIS Standard. eXtensible Access Control Markup Language (XACML) Version 3.0,” 2013. [Online]: http://docs.oasis-open.org/xacml/3.0/.
[121] “OASIS SAML Wiki,” 2019. [Online]. Available: https://wiki.oasis-open.org/security/FrontPage.
[122] D. Ferraiolo, J. Cugini, and D. R. Kuhn, “Role-based access control (RBAC): Features and motivations,” Proc. 11th Annu. Comput. Secur. Appl. Conf., 1995.
[123] E. Yuan and J. Tong, “Attributed Based Access Control (ABAC) for web services,” in Proceedings - 2005 IEEE International Conference on Web Services, ICWS 2005, 2005.
[124] S. Gusmeroli, S. Piccione, and D. Rotondi, “A capability-based security approach to manage access control in the Internet of Things,” Math. Comput. Model., 2013.
[125] J. B. Dennis and E. C. Van Horn, “Programming Semantics for Multiprogrammed Computations,” Commun. ACM, 1983.
[126] D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2001.
[127] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Lecture Notes in Computer Science, 2005.
[128] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in Proceedings - IEEE Symposium on Security and Privacy, 2007.

[129] H. Tai, A. Celesti, M. Fazio, M. Villari, and A. Puliafito, “An integrated system for advanced water risk management based on cloud computing and IoT,” 2015 2nd World Symp. Web Appl. Networking, WSWAN 2015, 2015.
[130] A. Jacobsson, M. Boldt, and B. Carlsson, “A risk analysis of a smart home automation system,” Futur. Gener. Comput. Syst., vol. 56, pp. 719–733, 2016.
[131] H. F. Atlam, A. Alenezi, R. J. Walters, and G. B. Wills, “An Overview of Risk Estimation Techniques in Risk-based Access Control for the Internet of Things Developing an adaptive Risk-based access control model for the Internet of Things View project A Framework to secure the shared Virtual machine Image in clou,” no. May, 2017.
[132] M. Rak, V. Casola, A. De Benedictis, and U. Villano, Automated Risk Analysis for IoT Systems, vol. 1, no. ii. Springer International Publishing, 2019.
[133] P. Radanliev, D. De Roure, J. Nurse, R. M. Montalvo, and P. Burnap, “Standardisation of cyber risk impact assessment for the Internet of Things (IoT),” pp. 1–50, 2019.
[134] P. Radanliev et al., “Definition of Internet of Things (IoT) Cyber Risk – Discussion on a Transformation Roadmap for Standardization of Regulations, Risk Maturity, Strategy Design and Impact Assessment,” Sensors, 2019.
[135] P. Radanliev, “Design principles for cyber risk impact assessment from Internet of Things (IoT),” no. March, pp. 1–8, 2019.
[136] Basili, V., Caldiera, G. & Rombach, D. (1994). The Goal Question Metric Approach. In J. J. Marciniak (Hrsg.), Encyclopedia of Software Engineering (Bd. 1, 528-532). New York: John Wiley and Sons, Inc.
[137] Basili, V., Trendowicz, A., Kowalczyk, M., Heidrich, J., Seaman, C., Münch, J., & Rombach, D. (2014). Aligning Organizations Through Measurement: The GQM+ Strategies Approach. Springer.
[138] Kuka, Christian; Nicklas, Daniela (2014): Enriching sensor data processing with quality semantics. In : PERCOM WORKSHOPS. 2014 IEEE International Conference on Pervasive Computing and Communication Workshops : 24-28 March 2014. 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS). Budapest, Hungary, 24.03.2014 - 28.03.2014. New York: IEEE, pp. 437–442, checked on 2/28/2020.
[139]]JDL, Data Fusion Lexicon. Technical Panel For C3, F.E. White, San Diego, Calif, USA, Code 420, 1991
[140] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” Computer Networks, vol. 57, no. 10, pp. 2266–2279, 2013.
[141] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, “Security challenges in the ip-based internet of things,” Wireless Personal Communications, vol. 61, no. 3, pp. 527–542, 2011.
[142] H. Yu, J. He, T. Zhang, P. Xiao, and Y. Zhang, “Enabling end-to-end secure communication between wireless sensor networks and the internet,” World Wide Web, pp. 1–26, 2013.
[143] Kanter, J. M., & Veeramachaneni, K. (2015, October). Deep feature synthesis: Towards automating data science endeavors. In 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 1-10). IEEE.
[144]Bach, S. H., Rodriguez, D., Liu, Y., Luo, C., Shao, H., Xia, C., ... & Kuchhal, R. (2019, June). Snorkel drybell: A case study in deploying weak supervision at industrial scale. In Proceedings of the 2019 International Conference on Management of Data (pp. 362-375).

[145]Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. SPSS inc, 9, 13.
[146] Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., & Grafberger, A. (2018). Automating large-scale data quality verification. Proceedings of the VLDB Endowment, 11(12), 1781-1794.

ANNEXES

Annex A Requirements Mapping

Data Preparation & Integration
Data Management
Data Analytics
Data Fusion
Data Quality
Data Security

DK3

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK3.1	Guarantee interoperability between communicating entities in the platform	PSNC, m2xpert, TECNALIA	NO	8.3, 8.4	AIM + Data Preparation & Integration Components
DK3.2	Integration of heterogeneous data types	PSNC, TECNALIA	NO	8.3	Data Preparation & Integration Components
DK3.3	Access to linked (integrated) datasets	PSNC, ICCS	NO	8.4.2	Data Preparation & Integration Components
DK3.4	Methods and tools for data integration	PSNC, ICCS, m2xpert, tecnalia	NO	8.4.1	Data Preparation & Integration Components
DK3.5	Select suitable tools for the semantic annotation of datasets	PSNC, m2xpert, TECNALIA	NO	8.4.1	Data Preparation & Integration Components
DK3.6	Linked Data exploration/visualization interfaces	PSNC, TECNALIA	NO	8.4.1	Data Preparation & Integration Components

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK3.7	Query translation with interoperable API	PSNC, m2xpert, TECNALIA	NO	8.4.1, 8.4.2	Data Preparation & Integration Components

DK4

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK4.1	Data management lifecycle	ENG, ATOS, PSNC	NO	7.1	Data Management Components
DK4.2	Data availability	ENG, ATOS, PSNC	NO	7.2.2	Data Management Components
DK4.3	Data integration mechanisms	ENG, ATOS, PSNC	NO	7.3.1	Data Management Components
DK4.4	API framework data and semantic interoperability	ENG	NO	7.3.1	Data Management Components
DK4.5	Services documentation and logging	ATOS, PSNC	NO	7.3.1, 7.3.2	Data Management Components
DK4.6	Data storage availability for heterogenous datasets	ENG, ATOS, PSNC	NO	7.3.1, 7.3.2	Data Management Components
DK4.7	Data storage deployment	ENG, ATOS, PSNC	NO	7.3.1, 7.3.2	Data Management Components

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK4.9	High availability of data storage	ENG, ATOS, INTRA, PSNC	NO	7.2.2	Data Management Components
DK4.9	Data storage of related metadata	ENG, ATOS, PSNC	NO	12	Data Analytic Components
DK4.10	Data synchronization	ENG	NO	7.3.1	Data Management Components
DK4.11	Data synchronization frequency	ENG, ATOS, PSNC	NO	7.3.1	Data Management Components
DK4.12	Data synchronization (batch and real-time)	ENG	NO	7.3.1	Data Management Components
DK4.13	Data synchronization stability	ENG	NO	7.3.1	Data Management Components
DK4.14	Data synchronization in a pub-sub fashion	ENG, INTRA, LESPROJEKT	NO	7.3.1	Data Management Components
DK4.15	Data access methods	ENG, PSNC	NO	7.3.1	Data Management Components
DK4.16	Connection caching mechanisms	ENG	NO	7.3.1	Data Management Components
DK4.18	Data preprocessing for advanced analytics	ENG	NO	7.3.1	Data Management Components

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK4.19	Business Intelligence client Tool	ENG, ATOS, PSNC	NO	12	Data Analytic Components
DK4.20	Data discovery tools and technologies	ENG	NO	7.3.1	Data Management Components
DK4.21	Data Aggregation Tools	ENG	NO		Data Analytic Components
DK4.22	Data Flow/Pipeline Procedure Support	ATOS	NO	7.3.1	Data Management Components
DK4.23	Data Grouping, Filtering and Aggregation Function Set	ENG, m2Xpert	NO	12	Data Analytic Component
DK4.24	Data Warehousing Support	ENG	NO	12	Data Analytic Components
DK4.25	Multi-tenancy	ENG	NO	7.2.1, 7.3.1, 7.3.2	Data Management Components

DK5, DK6

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK5.1	Fusion of data from (content-wise) heterogeneous data sources	ICCS	no	12.4	Targeted Data Fusion Modules

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK5.2	Selection of information extracted from a metadata model	ROT	no	12.4	Targeted Data Fusion Modules
DK5.3	Data fusion support for multiple file formats	ICCS	no	8.3.4	Data Access Facilities
DK5.4	Data Fusion Knowledge extraction (Agriculture Information Model - based)	ICCS	no	12.4	Targeted Data Fusion Modules
DK5.5	Data fusion techniques for distributed (big) data	ICCS, FhG.FIT, FhG.IESE, ROT	no	8, 12.4	Data Management Components, Targeted Data Fusion Modules
DK5.7	Standardized data fusion	ICCS, FhG.FIT, ROT	no	12.2	Data Fusion API
DK5.8	Fusing data from different systems	ICCS, ROT	no	10	Data Integration Components
DK5.11	Perform measures/actions based on assessment results	FhG.IESE	no	9.3.2	Data Cleaning / Validation (Data Preparation and Integration Pipeline API)
DK5.12	Support selection of appropriate data analysis techniques	FhG.IESE	no	12.6	Algorithm Selection Component

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK5.13	Support for interoperability and interchangeability of data quality assessment	ICCS, Fraunhofer	no	11.4	Data Quality Metadata and Provenance
DK5.14	Decision-support for data source selection	ICCS, FhG.FIT, FhG.IESE, ROT	no	12.6	Feature Engineering Component
DK5.15	Optimum value extraction	ICCS	no	9.3.2	Optimum Value Extraction (Data Preparation and Integration Pipeline API)
DK5.17	Different metrics for measuring data quality	ICCS, FhG.FIT, FhG.IESE	no	11.3	Data Quality Assessment Components
DK5.19	Handling of missing or contradicting data	ICCS	no	9, 12.6	Data Preparation, Feature Engineering Component
DK5.20	Quality-preserving collection and fusion	ATOS, ICCS	no	12.4	Targeted Data Fusion Modules
DK5.21	Data provenance to ensure fused data quality	ICCS, ATOS	no	11.4	Data Quality Metadata and Provenance
DK5.22	Data analysis and process actions in a timely manner	ICCS, FhG.FIT, FhG.IESE, ROT	no	8, 12.2, 12.3	Data Management, Targeted Data Analytics Modules, Data Analytics and Fusion API

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK5.23	Ensure quality of data streams	ICCS, ROT	no		
DK5.25	Involvement of domain experts and stakeholders	FhG.IESE	no	11.3.2.2	Specialized Data Quality Assessment Approaches
DK5.26	Access to (real) data and metadata	FhG.IESE	no	11.3.2.2	Specialized Data Quality Assessment Approaches
DK5.29	Easily understandable and machine-executable data quality metrics	FhG.FIT	no	11.4	Data Quality Metadata and Provenance
DK6.1	Data analytics support for data in various formats	ICCS, m2Xpert	no	8.3.4	Data Access Facilities
DK6.3	Data analytics should provide appropriate input for the Decision Support Systems	ICCS	no	12.9	DSS Integration
DK6.6	Avoid analysis bias	ATOS, Fraunhofer	no	12.7	Auditable, Explainable and Fair Analytics
DK6.8	Support for Exploratory Numerical Data Analysis	m2Xpert	no	8.3.4	Data Access Facilities
DK6.9	Support for Semantic Cross-Referencing	m2Xpert	no	10	Knowledge Discovery Component

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK6.10	Statistical Model Storage and Modification	m2Xpert	no	12.8	Model Management Component
DK6.11	Ethical data analytics	ATOS	no	12.7	Auditable, Explainable and Fair Analytics
DK6.12	Dataset should facilitate the generation or acquisition of datasets for training ML models.	ATOS	no	12.6	Feature Engineering Component
DK6.13	Accurate, consistent and timely analytics	ICCS	no	8, 12.2, 12.3	Data Management, Targeted Data Analytics Modules, Data Analytics and Fusion API
DK6.14	Data Analytics Performance and Accuracy Requirement	ICCS	no	12.2, 12.3	Targeted Data Analytics and Fusion Components
DK6.15	Data Analytics should deal with data quality issue(s)	ICCS, Fraunhofer IESE	no	11	Data Quality Components
DK6.16	Auditable DSS and Analytics	ICCS	no	12.7	Auditable, Explainable and Fair Analytics
DK6.17	Flexible architecture for analytics	ICCS		12.1	Data Analytics and Fusion Components Introduction
DK6.19	Computer vision for pattern extraction	ATOS	yes	12.2.1	Data Analytics Computer Vision Pattern Extraction

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
					Module
DK6.20	Analytics using EO and environmental data for plant monitoring	ICCS	yes	12.3.1	Fusion of Satellite, Spectral and UAV Imagery for Rice and Maize Fields
DK6.21	Cross-farm data processing	ICCS	yes	8, 10	Data Management, Data Integration
DK6.22	Environmental data analytics	ICCS	yes	12.4.1	Fusion of Satellite, Spectral and UAV Imagery for Rice and Maize Fields
DK6.23	Analytics for optimized fertilizer use	ICCS	yes	12.3.3	Pattern Extraction for N. Uptake, Biomass and Chlorophyll of Rice/Maize
DK6.24	Analytics over UAV data for disease forecasting	ICCS	yes	12.3.2	Pattern Extraction for Fruit Fly Counting
DK6.25	Integrating heterogeneous data for automation purposes	ICCS	yes	10	Data Integration
DK6.26	Analytics on animal properties for milk quality reasoning	ICCS	yes	12.3.x/WP4 DSS Enabler	

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK6.27	Analytics for optimal water quality	ICCS	yes	12.3.4	Data Analysis for Water Salinity and Plant Toxicity (salt) in Maize Fields
DK6.28	Analytics for optimal usage of pesticides	ICCS	yes	12.3.x/WP4 DSS Enabler	
DK6.29	Linking of Analytics Data to the DSS to enable cross-sectoral interoperability of the platforms	ICCS	yes	12.9	DSS Integration
DK6.30	Data analytics over GPS and map data	ICCS	yes	12.3.x/WP4 DSS Enabler	
DK6.31	Event triggering via real time analytics	ICCS	yes	WP4	DSS Performance Monitoring & Alerting Facilities
DK6.32	Data analytics for predictability of resource/input needs	ICCS	yes	12.3.x/WP4 DSS Enabler	

DK7

Requirement	Description	Reported	Pilot-specific	Chapter	DEMETER Module/Component
DK7.1a	Lightweight messaging protocols handling encryption	VICOM ROT	NO	14.2.5 14.3.5	Data Security Communication & Networking Enabler

DK7.1b	Secured communication	VICOM ROT	NO	14.2.5 14.3.5	Data Security Communication & Networking Enabler
DK7.1c	Secured transport layer	VICOM ROT	NO	14.2.5 14.3.5	Data Security Communication & Networking Enabler
DK7.1d	Encryption should begin at sensor level	VICOM ROT	NO	14.2.5 14.3.5	Data Security Communication & Networking Enabler
DK7.1e	Secure way to handle resource constrained devices communication	VICOM ROT	NO	14.2.5 14.3.5	Data Security Communication & Networking Enabler
DK7.1f	Monitoring of intrusion detection	ROT	NO	14.2 14.3	Dasta Communication Enabler
DK7.1g	Management of alarms intrusions	ROT	NO	14.2 14.3	Dasta Communication Enabler
DK7.2a	Common formats and standards for information exchange that are also secure	VICOM UMU ODINS ROT	NO	14.2 14.3	Dasta Security Auth(n), Auth(z) and Communication & Networking, Components and Enablers
DK7.2b	Formats and standards must allow cryptography	VICOM ROT	NO	14.2.5 14.3.5	Data Security Communication & Networking Enabler
DK7.3a	Distributed, capability and attribute based access control system	VICOM UMU ODINS	NO	14.2.2 14.2.3 14.3.2 14.3.3	Dasta Security Auth(n) and Auth(z) Components and Enablers

DK7.3b	Authentication and authorization, traceability: Secure transport layer for authn/authz	VICOM UMU ODINS ROT	NO	14.2 14.3	Dasta Security Auth(n), Auth(z) and Communication & Networking, Components and Enablers
DK7.3c	Policy language for defining the access to resources	VICOM UMU ODINS	NO	14.2.3 14.3.3	Dasta Security Auth(z) Component and Enabler
DK7.3d	Data handling policy language to set how requested data is handled and passed on.	VICOM UMU ODINS	NO	14.2.3 14.3.3	Dasta Security Auth(z) Component and Enabler
DK7.3e	Define which users and devices will have access to what, and when and how	VICOM UMU ODINS	NO	14.2.2 14.2.3 14.3.2 14.3.3	Dasta Security Auth(n) and Auth(z) Components and Enablers
DK7.3f	Appropriate traceability for heterogeneous datasets	VICOM	NO	14.2.4 14.3.4	Dasta Security Traceability Component
DK7.3g	Capability of the data owner to specify who can access, process and store its data	VICOM UMU ODINS	NO	14.2.2 14.2.3 14.3.2 14.3.3	Dasta Security Auth(n) and Auth(z) Components and Enablers
DK7.4a	Content encryption/decryption and encoding of data	VICOM ROT	NO	14.2.5 14.3.5	Data Security Communication & Networking Enabler

DK7.4b	Protect personal data.	VICOM UMU ODINS ROT	NO	14.2 14.3	Dasta Security Auth(n), Auth(z) and Communication & Networking, Components and Enablers
DK7.4c	Protect sensitive data.	VICOM UMU ODINS ROT	NO	14.2 14.3	Dasta Security Auth(n), Auth(z) and Communication & Networking, Components and Enablers
DK7.5	Comply with GDPR technical requirements	VICOM UMU ODINS ROT	NO	14.2 14.3	Dasta Security Auth(n), Auth(z) and Communication & Networking, Components and Enablers
DK7.6	Perform Court-proof logging and audit logs.	VICOM	NO	14.2.4 14.3.4	Dasta Security Traceability Component
DK7.7	Preservation of data access rights	VICOM UMU ODINS ROT	NO	14.2 14.3	Dasta Security Auth(n), Auth(z) and Communication & Networking, Components and Enablers

Annex B Authentication endpoints documentation

This annex provides examples of http requests to the IdM Endpoints that provide authentication functionalities for the management of User, Organization, Roles and Applications for the components.

B.1. Create token with Password

- Request

```
curl --include \
       --request POST \
       --header "Content-Type: application/json" \
       --data-binary "{
  \"name\": \"alice@test.com\",
  \"password\": \"passw0rd\"
}" \
'http://keyrock/v1/auth/tokens'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Subject-Token: 04c5b070-4292-4b3f-911b-36a103f3ac3f Content-Length:74 ETag:W/"4a-jYFzvNRMQcIZ2P+p5EfmbN+VHcw" Date:Mon, 19 Mar 2018 15:05:35 GMT Connection:keep-alive

Body	{
	"token": {
	"methods": ["password"],
	"expires_at": "2018-03-20T15:05:35.697Z"
	}
	}

B.2. Refresh token

- Request

```
curl --include \
  --request POST \
  --header "Content-Type: application/json" \
  --data-binary "{
  \"token\": \"token_id\"
}" \
'http://keyrock/v1/auth/tokens'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Subject-Token:04c5b070-4292-4b3f-911b-36a103f3ac3f Content-Length:74 ETag:W/"4a-jYFzvNRMQcIZ2P+p5EfmbN+VHcw" Date:Mon, 19 Mar 2018 15:05:35 GMT Connection:keep-alive

Body	{
	"token": {
	"methods": ["password"],
	"expires_at": "2018-03-20T15:05:35.697Z"
	}
	}

B.3. Get token details

- Request

```
curl --include \
       --header "X-Auth-token: auth_token" \
       --header "X-Subject-token: subj_token" \
  'http://keyrock/v1/auth/tokens'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:278 ETag:W/"116-JwJR8Y5eFV2SDon0j1GE5yWyNx4" Date:Tue, 20 Mar 2018 15:23:21 GMT

	Connection:keep-alive

Body	{
	"access_token": "f0a0a067-6341-4943-8225-45f794b3d94b",
	"expires": "2018-03-21T15:22:33.000Z",
	"valid": true,
	"User": {
	"id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	"username": "alice_new",
	"email": "alice@test.com",
	"date_password": "2018-03-20T12:12:09.000Z",
	"enabled": true,
	"admin": false
	}
	}

B.4. Applications

B.4.1. **List Applications**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/applications'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:303 ETag:W/"12f-64q7xcRDNfw7c2xrc1r2S5EFS6k" Date:Tue, 20 Mar 2018 10:14:30 GMT Connection:keep-alive
Body	{ "applications": [{ "id": "0fbfa58c-e5b6-41c3-b748-ab29f1567a9c", "name": "Test_application 2", "description": "Description", "image": "default", "url": "http://localhost", "redirect_uri": "http://localhost/login", "grant_type": "client_credentials,password,implicit,authorization_code,refresh_token", "response_type": "code,token", "token_types": "bearer,jwt,permanent", "client_type": null }, {

```
        "id": "fd7fe349-f7da-4c27-b404-74da17641025",
        "name": "Test_application 1",
        "description": "description",
        "image": "default",
        "url": "http://localhost",
        "redirect_uri": "http://localhost/login",
        "grant_type": "password,authorization_code,implicit",
        "response_type": "code,token",
        "token_types": "bearer",
        "client_type": null
        }
       ]
}
```


B.4.2. **Create an Application**

- Request

```
curl --include \
      --request POST \
      --header "Content-Type: application/json" \
      --header "X-Auth-token: token" \
      --data-binary "{
  \"application\": {
      \"name\": \"Test_application 1\",
      \"description\": \"description\",
      \"redirect_uri\": \"http://localhost/login\",
      \"url\": \"http://localhost\",
      \"grant_type\": [
      \"authorization_code\",
      \"implicit\",
      \"password\"
      ],
      \"token_types\": [
      \"jwt\",
      \"permanent\"
      ]
  }
}" \
'http://keyrock/v1/applications'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:329 ETag:W/"149-yQORsIAntQ0YHR5uCyLZhk7jTkg" Date:Tue, 20 Mar 2018 10:16:13 GMT Connection:keep-alive

Body	{
	"application": {
	"id": "fd7fe349-f7da-4c27-b404-74da17641025",
	"secret": "9dc463cf-8318-4f65-bc02-778424fdfd77",
	"image": "default",
	"name": "Test_application 1",
	"description": "description",
	"redirect_uri": "http://localhost/login",
	"url": "http://localhost",
	"grant_type": "password,authorization_code,implicit",
	"token_types": "jwt,permanent",
	"jwt_secret": "3f1164da20d50c62",
	"response_type": "code,token"
	}
	}
	}

B.4.3. **Read Application details**

- Request

```
curl --include \
     --header "X-Auth-token: token" \
  'http://keyrock/v1/applications/application_id'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:407 ETag:W/"197-biN1lSH/UM4L+dEtIoLYyOGteWE" Date:Tue, 20 Mar 2018 10:25:33 GMT Connection:keep-alive

Body	{
	"application": {
	"id": "0fbfa58c-e5b6-41c3-b748-ab29f1567a9c",
	"name": "Test_application 2",
	"description": "Description",
	"secret": "61f5def7-bcf9-45b1-9c69-d0887e403737",
	"url": "http://localhost",
	"redirect_uri": "http://localhost/login",
	"image": "default",
	"grant_type":
	"client_credentials,password,implicit,authorization_code,refresh_token",
	"response_type": "code,token",
	"token_types": "bearer,jwt,permanent",
	"jwt_secret": "3f1164da20d50c62",
	"client_type": null,
	"scope": null,
	"extra": null
	}
	}

B.4.4. **Update an Application**

- Request

```
curl --include \
      --request PATCH \
      --header "Content-Type: application/json" \
      --header "X-Auth-token: token" \
      --data-binary "{
      \"application\": {
      \"name\": \"new name\",
      \"description\": \"new description\",
      \"redirect_uri\": \"new redirect uri\",
      \"token_types\": [\"permanent\"]
      \"grant_type\": [
            \"authorization_code\",
            \"password\"
      ]
      }
}" \
'http://keyrock/v1/applications/application_id'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:170 ETag:W/"aa-XCFRn1K54wlgPzun8PluFjYwCO0" Date:Tue, 20 Mar 2018 10:29:04 GMT Connection:keep-alive

Body	{
	"values_updated": {
	"name": "new name",
	"description": "new description",
	"redirect_uri": "new redirect uri",
	"grant_type": "password,authorization_code",
	"token_types": "bearer,permanent",
	"jwt_secret": null,
	"response_type": "code"
	}
	}

B.4.5. **Delete an Application**

- Request

```
curl --include \
      --request DELETE \
```

```
                --header "X-Auth-token: token" \
            'http://keyrock/v1/applications/application_id'
```

- Response

Value	204
Headers	Content-Type:application/json
Body	

B.5. Users

B.5.1. **List Users**

- Request

```
        curl --include \
                --header "X-Auth-token: token" \
          'http://keyrock/v1/users'
```

- Response

Value	204
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:378 ETag:W/"17a-/TnzfhPjjd4IG4D1u38zPZiSIL0" Date:Tue, 20 Mar 2018 10:08:27 GMT Connection:keep-alive
Body	{ "users": [{ "id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb", "username": "alice", "email": "alice@test.com", "enabled": true, "gravatar": false, "date_password": "2018-03-20T09:31:07.000Z", "description": null,

```
                "website": null
        },
        {
                "id": "admin",
                "username": "admin",
                "email": "admin@test.com",
                "enabled": true,
                "gravatar": false,
                "date_password": "2018-03-20T08:40:14.000Z",
                "description": null,
                "website": null
        }
         ]
}
```


B.5.2. **Create a User**

- Request

```
curl --include \
        --request POST \
        --header "Content-Type: application/json" \
        --header "X-Auth-token: token" \
        --data-binary "{
\"user\": {
        \"username\": \"alice\",
        \"email\": \"alice@test.com\",
        \"password\": \"passw0rd\"
}
}" \
'http://keyrock/v1/users'
```

- Response

Value	204
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:205 ETag:W/"cd-hbCwPDJrO5NvpsY2pqY6Qhlf/do" Date:Tue, 20 Mar 2018 09:31:07 GMT Connection:keep-alive

Body	{
	  "user": {
	  "id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	  "image": "default",
	  "gravatar": false,
	  "enabled": true,
	  "admin": false,
	  "username": "alice",
	  "email": "alice@test.com",
	  "date_password": "2018-03-20T09:31:07.104Z"
	  }
	}

B.5.3. **Read User´s details**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/users/user_id'
```

- Response

Value	204
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:239 ETag:W/"ef-346BFly5PUZzqso7/DhcynciNPs" Date:Tue, 20 Mar 2018 10:34:20 GMT Connection:keep-alive

Body	{
	"user": {
	"id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	"username": "alice",
	"email": "alice@test.com",
	"enabled": true,
	"admin": false,
	"image": "default",
	"gravatar": false,
	"date_password": "2018-03-20T09:31:07.000Z",
	"description": null,
	"website": null
	}
	}

B.5.4. **Update User´s details**

- Request

```
curl --include \
       --request PATCH \
       --header "Content-Type: application/json" \
       --header "X-Auth-token: token" \
       --data-binary "{
\"user\": {
       \"username\": \"new_alice\",
       \"password\": \"passw0rd_new\"
}
}" \
'http://keyrock/v1/users/user_id'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:72 ETag:W/"48-NygV9BIfH5juflD+icW6fPFuBkA" Date:Tue, 20 Mar 2018 10:38:59 GMT Connection:keep-alive

Body	{
	"values_updated": {
	"username": "alice_new",
	"email": "alice_new@test.com"
	}
	}

B.5.5. **Delete User´s details**

- Request

```
curl --include \
       --request DELETE \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/users/user_id'
```

- Response

Value	204
Headers	Content-Type:application/json
Body	

B.6. Roles

B.6.1. **List roles**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/applications/application_id/roles '
```

- Response

Value	200
Headers	Content-Type:application/json

Body	{
	"roles": [
	{
	"id": "purchaser",
	"name": "Purchaser"
	},
	{
	"id": "provider",
	"name": "Provider"
	},
	{
	"id": "ee2ec16f-694b-447f-b61a-e293b6fe5f7b",
	"name": "role 2"
	},
	{
	"id": "33fd15c0-e919-47b0-9e05-5f47999f6d91",
	"name": "role 1"
	}
	]
	}

B.6.2. **Create a role**

- Request

```
curl --include \
        --request POST \
        --header "Content-Type: application/json" \
        --header "X-Auth-token: token" \
        --data-binary "{
  \"role\": {
 \"name\": \"role 1\"
  }
}" \
'http://keyrock/v1/applications/application_id/roles'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:147 ETag:W/"93-gmWMlkuHssLlhgsbkRsYJjI1ZpE"

	Date:Tue, 20 Mar 2018 10:56:16 GMT Connection:keep-alive

Body	{
	"role": {
	"id": "33fd15c0-e919-47b0-9e05-5f47999f6d91",
	"is_internal": false,
	"name": "role 1",
	"oauth_client_id": "fd7fe349-f7da-4c27-b404-74da17641025"
	}
	}

B.6.3. **Read role details**

- Request

```
curl --include \
        --header "X-Auth-token: token" \
  'http://keyrock/v1/applications/application_id/roles/role_id'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:147 ETag:W/"93-jWdaZFK7V/AOFD7WOIM0aZePmqg" Date:Tue, 20 Mar 2018 11:21:19 GMT Connection:keep-alive

Body	{
	"role": {
	"id": "33fd15c0-e919-47b0-9e05-5f47999f6d91",
	"name": "role 1",
	"is_internal": false,
	"oauth_client_id": "fd7fe349-f7da-4c27-b404-74da17641025"
	}
	}

B.6.4. **Update role details**

- Request

```
curl --include \
        --request PATCH \
        --header "Content-Type: application/json" \
        --header "X-Auth-token: token" \
        --data-binary "{
  \"role\": {
\"name\": \"test_role_new\",
\"application_id\": \"2222\"
  }
```

```
              }" \
'http://keyrock/v1/applications/application_id/roles/role_id'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:43 ETag:W/"2b-pM4WlfxN6zpTuxeiYjs7ZBMAimM" Date:Tue, 20 Mar 2018 11:24:07 GMT Connection:keep-alive

Body	{
	"values_updated": {
	"name": "new role name"
	}
	}

B.6.5. **Delete a role**

- Request

```
              curl --include \
                   --request DELETE \
                   --header "X-Auth-token: token" \
  'http://keyrock/v1/applications/application_id/roles/role_id'
```

- Response

Value	204
Headers	Content-Type:application/json

Body	

B.7. Organizations

B.7.1. **List organizations**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/organizations'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:355 ETag:W/"163-0eDqBVBTEbKRQmAm6AtzDLXVigI" Date:Tue, 20 Mar 2018 10:45:36 GMT Connection:keep-alive

Body	{
	"organizations": [
	{
	"role": "owner",
	"Organization": {
	"id": "33cf4d3c-8dfb-4bed-bf37-7647f45528ec",
	"name": "Test organization 2",
	"description": "description 2",
	"image": "default",
	"website": null
	}
	},
	{
	"role": "owner",
	"Organization": {
	"id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	"name": "Test organization",
	"description": "description",
	"image": "default",
	"website": null
	}
	}
	]
	}

B.7.2. **Create an organization**

- Request

```
curl --include \
       --request POST \
       --header "Content-Type: application/json" \
       --header "X-Auth-token: token" \
       --data-binary "{
\"organization\": {
       \"name\": \"Test organization\",
       \"description\": \"description\"
}
} " \
'http://keyrock/v1/organizations'
```

- Response

Value	201
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:135 ETag:W/"87-mQPZWVCWNw1jTeBUp2u6DfQUBXg" Date:Tue, 20 Mar 2018 10:41:14 GMT Connection:keep-alive

Body	{
	"organization": {
	"id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	"image": "default",
	"name": "Test organization",
	"description": "description"
	}
	}

B.7.3. **Read organization's details**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/organizations/organization_id'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:150 ETag:W/"96-Gd8UZNti5b/l9AK7zLVUgoPzoSg" Date:Tue, 20 Mar 2018 10:46:47 GMT

	Connection:keep-alive

Body	{
	"organization": {
	"id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	"name": "Test organization",
	"description": "description",
	"website": null,
	"image": "default"
	}
	}

B.7.4. **Update organization´s details**

- Request

```
curl --include \
       --request PATCH \
       --header "Content-Type: application/json" \
       --header "X-Auth-token: token" \
       --data-binary "{
  \"organization\": {
\"description\": \"new description\",
\"website\": \"http://test.com\"
  }
}" \
'http://keyrock/v1/organizations/organization_id'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 Content-Length:80 ETag:W/"50-SLhFvXhKe+sDolh13Q+2u5eotYw" Date:Tue, 20 Mar 2018 10:48:45 GMT Connection:keep-alive

Body	{
	"values_updated": {
	"description": "new description",
	"website": "http://test.com"
	}
	}

B.7.5. **Delete an organization**

- Request

```
curl --include \
       --request DELETE \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/organizations/organization_id'
```

- Response

Value	204
Headers	Content-Type:application/json
Body	

B.8. Applications, Organizations, Users and Roles: Relationships

B.8.1. **Add a user as to an organization**

- Request

```
curl --include \
       --request PUT \
       --header "Content-Type: application/json" \
       --header "X-Auth-token: token" \

'http://keyrock/v1/organizations/organization_id/users/user_id/organization_roles/organization_role_id'
```

- Response

Value	201
Headers	Content-Type:application/json; charset=utf-8 Content-Length:125 ETag:W/"7d-lt1JE+QQNKxkfaswZ+ZRWXj2vdk" Date:Tue, 20 Mar 2018 12:55:58 GMT Connection:keep-alive
Body	{ "user_organization_assignments": { "role": "owner", "user_id": "admin", "organization_id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd" } }

B.8.2. **List users within an organization**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/organizations/organization_id/users'
```

- Response

Value	201
Headers	Content-Type:application/json; charset=utf-8 Content-Length:240 ETag:W/"f0-3G8Dv2NqT3KMGuO+jRta+nJMnJk" Date:Tue, 20 Mar 2018 12:47:26 GMT Connection:keep-alive

Body	<pre>{
	  "organization_users": [
	  {
	  "user_id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	  "organization_id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	  "role": "owner"
	  },
	  {
	  "user_id": "admin",
	  "organization_id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	  "role": "member"
	  }
	  ]
	}</pre>

B.8.3. **Remove a user from an organization**

- Request

```
curl --include \
       --request DELETE \
       --header "X-Auth-token: token" \

              'http://keyrock/v1/organizations/organization_id/users/user_id/organization
              _roles/organization_role_id'
```

- Response

Value	204
Headers	

Body	

B.8.4. **Read user´s role within an organization**

- Request

```
curl --include \
       --header "X-Auth-token: token" \

             'http://keyrock/v1/organizations/organization_id/users/user_id/organization
             _roles'
```

- Response

Value	201
Headers	Content-Type:application/json; charset=utf-8 Content-Length:114 ETag:W/"72-/fZonAf/VJOEZ94bshoBRSVQzmE" Date:Tue, 20 Mar 2018 12:51:28 GMT Connection:keep-alive

Body	{
	"organization_user": {
	"user_id": "admin",
	"organization_id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	"role": "member"
	}
	}

B.8.5. **Grant a role to an organization**

- Request

```
curl --include \
       --request PUT \
       --header "Content-Type: application/json" \
       --header "X-Auth-token: token" \

'http://keyrock/v1/applications/application_id/organizations/organization_i
d/roles/role_id/organization_roles/organization_role_id'
```

- Response

Value	201
Headers	X-Powered-By:Express Content-Type:application/json; charset=utf-8 Content-Length:198 ETag:W/"c6-6eTVB6vyiwH6XKLZ3pLYYDXFMm4" Date:Tue, 20 Mar 2018 12:41:47 GMT Connection:keep-alive

Body	{
	"role_organization_assignments": {
	"role_id": "provider",
	"organization_id": "33cf4d3c-8dfb-4bed-bf37-7647f45528ec",
	"oauth_client_id": "fd7fe349-f7da-4c27-b404-74da17641025",
	"role_organization": "owner"
	}
	}

B.8.6. **List granted organization roles**

- Request

```
curl --include \
        --header "X-Auth-token: token" \

'http://keyrock/v1/applications/application_id/organizations/organization_id/roles'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:332 ETag:W/"14c-Itun6PzsweI15qednKd5+tG7Oh0" Date:Tue, 20 Mar 2018 12:39:20 GMT Connection:keep-alive

Body	{
	"role_organization_assignments": [
	{
	"organization_id": "33cf4d3c-8dfb-4bed-bf37-7647f45528ec",
	"role_id": "purchaser"
	},
	{
	"organization_id": "33cf4d3c-8dfb-4bed-bf37-7647f45528ec",
	"role_id": "ee2ec16f-694b-447f-b61a-e293b6fe5f7b"
	},
	{
	"organization_id": "33cf4d3c-8dfb-4bed-bf37-7647f45528ec",
	"role_id": "33fd15c0-e919-47b0-9e05-5f47999f6d91"
	}
	]

	}

B.8.7. **Revoke a role from an organization**

- Request

```
curl --include \
        --request DELETE \
        --header "X-Auth-token: token" \

'http://keyrock/v1/applications/application_id/organizations/organization_id/roles/role_id/organization_roles/organization_role_id'
```

- Response

Value	204
Headers	
Body	

B.8.8. **Grant a role to a user**

- Request

```
curl --include \
        --request PUT \
        --header "Content-Type: application/json" \
        --header "X-Auth-token: token" \

'http://keyrock/v1/applications/application_id/users/user_id/roles/role_id'
```

- Response

Value	201
Headers	X-Powered-By:Express

	Content-Type:application/json; charset=utf-8 Content-Length:155 ETag:W/"9b-PTfLJchMB1s6mXINkQh6OJQnSgY" Date:Tue, 20 Mar 2018 12:31:58 GMT Connection:keep-alive

Body	{
	"role_user_assignments": {
	"role_id": "purchaser",
	"user_id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	"oauth_client_id": "fd7fe349-f7da-4c27-b404-74da17641025"
	}
	}

B.8.9. **List granted user roles**

- Request

```
curl --include \
        --header "X-Auth-token: token" \
  'http://keyrock/v1/applications/application_id/users/user_id/roles'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:199 ETag:W/"c7-kciYHLq/V20GkqNyzgr7HDzrWI4" Date:Tue, 20 Mar 2018 12:29:34 GMT Connection:keep-alive

Body	{
	"role_user_assignments": [
	{
	"user_id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	"role_id": "provider"
	},
	{
	"user_id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	"role_id": "ee2ec16f-694b-447f-b61a-e293b6fe5f7b"
	}
	]
	}

B.8.10. **Revoke a role to a user**

- Request

```
curl --include \
        --request DELETE \
        --header "X-Auth-token: token" \

'http://keyrock/v1/applications/application_id/users/user_id/roles/role_id'
```

- Response

Value	204
Headers	
Body	

B.8.11. **Read user roles within an organization**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/applications/application_id/users/user_id/roles'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:199 ETag:W/"c7-kciYHLq/V20GkqNyzgr7HDzrWI4" Date:Tue, 20 Mar 2018 12:29:34 GMT Connection:keep-alive

Body	<pre>{
	  "role_user_assignments": [
	  {
	  "user_id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	  "role_id": "provider"
	  },
	  {
	  "user_id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	  "role_id": "ee2ec16f-694b-447f-b61a-e293b6fe5f7b"
	  }
	  ]
	}</pre>

B.8.12. **List authorized organizations for an application**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/applications/application_id/organizations'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:798 ETag:W/"31e-yAFZkeaD7GA4Xo4F6dAyMc/FISM" Date:Tue, 20 Mar 2018 12:37:51 GMT Connection:keep-alive

Body	{
	"role_organization_assignments": [
	{
	"organization_id": "33cf4d3c-8dfb-4bed-bf37-7647f45528ec",
	"role_organization": "owner",
	"role_id": "purchaser"
	},
	{
	"organization_id": "33cf4d3c-8dfb-4bed-bf37-7647f45528ec",
	"role_organization": "owner",
	"role_id": "ee2ec16f-694b-447f-b61a-e293b6fe5f7b"
	},
	{
	"organization_id": "33cf4d3c-8dfb-4bed-bf37-7647f45528ec",
	"role_organization": "member",
	"role_id": "33fd15c0-e919-47b0-9e05-5f47999f6d91"
	},
	{
	"organization_id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	"role_organization": "owner",
	"role_id": "provider"
	},
	{
	"organization_id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	"role_organization": "owner",
	"role_id": "33fd15c0-e919-47b0-9e05-5f47999f6d91"
	},
	{
	"organization_id": "3e20722f-d420-422d-89ba-3ae87bc1c0cd",
	"role_organization": "member",
	"role_id": "ee2ec16f-694b-447f-b61a-e293b6fe5f7b"
	}
	]
	}

B.8.13. **List authorized users for an application**

- Request

```
curl --include \
       --header "X-Auth-token: token" \
  'http://keyrock/v1/applications/application_id/users'
```

- Response

Value	200
Headers	Content-Type:application/json,application/json; charset=utf-8 X-Powered-By:Express Content-Length:310 ETag:W/"136-lW/H57r3jyK+ObyFOlWMB5tMuNU" Date:Tue, 20 Mar 2018 12:27:09 GMT Connection:keep-alive

Body	{
	"role_user_assignments": [
	{
	"user_id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	"role_id": "provider"
	},
	{
	"user_id": "admin",
	"role_id": "purchaser"
	},
	{
	"user_id": "2d6f5391-6130-48d8-a9d0-01f20699a7eb",
	"role_id": "ee2ec16f-694b-447f-b61a-e293b6fe5f7b"
	},
	{
	"user_id": "admin",
	"role_id": "ee2ec16f-694b-447f-b61a-e293b6fe5f7b"
	}
	]
	}