D3.1 DEMETER Reference Architecture (Release 1)

Dissemination Level: Public
Submission Date: 28/02/2020

Contents

1 Executive Summary

DEMETER aims to lead the Digital Transformation of the European Agrifood 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. It aims to 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 starts by delivering a Reference Architecture (RA) that is suitable to address these challenges in the agrifood domain. It builds upon a number of related state-of-the-art solutions aiming to demonstrate all of their advantages combined and is based on a thorough requirements analysis, imposed by the pilot stakeholders and guided by the DEMETER vision, embracing an holistic multi-actor approach. This deliverable introduces the initial release of the DEMETER Reference Architecture that will be used to guide the development of the technologies in all work packages that will be used for the first round of the DEMETER pilots. This architecture specification follows the analysis of the State of the Art and the first elicitation of the requirements for the technical work packages. In this respect, it initially discusses the methodology employed to design the RA and reviews the related state-of-the-art work, justifying why specific related frameworks that have been taken into account. It provides an overview of the technical requirements that should be addressed by the DEMETER RA and elaborates on the main RA concepts. It then describes in detail the actual DEMETER RA through a number of viewpoints (i.e., high-level, functional, process, data, deployment and business views), gives an overview of the main interactions among the core RA building blocks and presents the specific RA instantiations that will be implemented to serve the 20 DEMETER pilots. Finally, the document concludes with an elaboration on the GDPR concerns and guidelines that will need to be enforced in order to protect the data and privacy of the various DEMETER stakeholders.

2 Acronyms

ABE	Attribute-Based Encryption
AI PPP	AI Public Private Partnership
AIOTI	Alliance for the Internet of Things Innovation
AIS	Agricultural Interoperability Space
API	Application Programming Interface
ARIES	ReliAble euRopean Identity EcoSystem
ARM	Architectural Reference Model
BDVA	Big Data Value Association
CA	Certificate Authority
CEP	Complex Event Processing
CIM	Context Information Management
CoAP	Constraint Application Protocol
CP-ABE	Cyphertext-policy Attribute-based Encryption
CPS	Cyber-Physical Systems
DAE	DEMETER Advanced Enabler
DDS	Data Distribution Service
DEH	DEMETER Enabler HUB
DOM	Document Object Model
DP	Digital Platforms
DS	Decision Support
DSS	Decision Support System
ECC	Elliptic Curve Cryptography
ECDH	Elliptic-Curve Diffie–Hellman
ECMQV	Elliptic Curve Menezes-Qu-Vanstone
ECSEL	Electronic Components and Systems for European Leadership
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
FC	Functional Components
FG	Functionality Groups
FMIS	Farm Management Information System
GDPR	General Data Policy Regulations
GE	Generic Enablers
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HLA	High Level Reference Architectures
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IBE	Identity-Based Encryption
IC	Integration Component
ICT	Information and communications technology
IDAS	Integrated Data Acquisition System
IdM	Identity Management

IDS	Industrial Data Space
IDSA	International Data Spaces Association
IDS-RAM	IDS Reference Architecture Model
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IIC	Industrial Internet Consortium
IIOT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
IOT	Internet of Things
IP	Internet Protocol
ISG	Industry Specification Group
ISO	International Organization for Standardisation
IT	Information Technology
ITU	International Telecommunication Union
JSON	Java Script Object Notation
KPI	key performance indicators
LAN	Local Area Network
LSP	Large-Scale Pilots
MES	Manufacturing Execution System
MMT	Mission Management Tool
MQTT	Message Queuing Telemetry Transport
NBDRA	NIST Big Data Reference Architecture
NGSI	Next Generation Sensors Initiative
NGSI-LD	Next Generation Sensors Initiative - Linked Data
NIST	National Institute of Standards and Technology
NI-ZKP	Non-Interactive Zero Knowledge Proof
OMA-LWM2M	Open Mobile Alliance - Lightweight Machine to Machine
OneM2M	One Machine to Machine
OSI	Open Systems Interconnection
OT	Operational Technology
OWL	Web Ontology Language
PAN	Personal Area Network
PKC	Public Key Cryptography
PLC	Programmable Logic Controller
RAMI	Reference Architectural Model Industrie
RAS	Reliability, Availability and Serviceability
RDF	Resource Description Framework
REST	Representational state transfer
RFID	Radio Frequency Identification Device
RPC	Remote Procedure Calls
RSA	Rivest-Shamir-Adleman
RTPS	Real Time Publish Subscribe
S&P	Security and Privacy
SCADA	Supervisory Control And Data Acquisition

SCALE	Security, Cognition, Agility, Latency, Efficiency
SDK	Software Development Kit
SKC	Symmetric Key Cryptography
SKOS	Simple Knowledge Organization System
SOCS	Stakeholders Open Collaboration Space
SPS	Speicherprogrammierbare Steuerung
SWRL	Semantic Web Rule Language
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAN	Wide Area Network
WSN	Wireless Sensor Network
XML	Extensible Markup Language

3 List of Authors and Reviewers

Organisation	Author
ICCS	Ioanna Roussaki (Editor)
	Ioannis Vetsikas
	George Routis
	Marios Paraskevopoulos
INTRA	Thanasis Poulakidas
	Ioannis Oikonomidis
ENG	Angelo Marguglio
	Antonio Caruso
UMU	Antonio Skarmeta
	Manuel Mora
TECNALIA	Sonia Bilbao
	Belén Martínez
	Fernando Jorge Hernandez
ATOS	Tomas Pariente Lobo
SINTEF	Arne Berre
	Bjørn Marius von Zernichow
SIVECO SA	Mihai Angheloiu
	Octavian Goicea
DNET	Nenad Gligoric
PSNC	Raul Palma
	Soumya Brahma
	Szymon Mueller
Vicomtech	Inés Goikoetxea
TSSG/WIT	Nithin Padmaprabhu
ICE	John Beattie
Prospeh BGD	Branimir Rakić

Organisation	Reviewer
ATOS	Tomas Pariente Lobo
	Javier Garcia Hernandez
	Sergio Salmeron
ICE	John Beattie

4 Introduction

This deliverable presents the first release of the DEMETER Reference Architecture that will be used to guide the development of the technologies in all the work packages that will be instantiated for the first round of the DEMETER pilots. This architecture specification builds upon a thorough analysis of the related State of the Art and is guided by the initial requirements extracted by the technical work packages and of course by the DEMETER vision and targeted objectives.

More specifically, the rest of the document is structured as follows.

Section 5 presents the methodology used in order to design the Reference Architecture of DEMETER; this is based partly on the ISO/IEC/IEEE 42010 International Standard and is useful in order to present the architecture in this document and in particular its various viewpoints.

Section 6 examines the issues that should be considered and introduces the principles and main examples of the Reference Model and Reference Architectures, coming mainly from the agrifood domain (and other relevant ones), thus having a baseline representation of the relevant European (and worldwide) architecture, before these are presented in detail in the next section.

Section 7 provides an analysis of the state of the art on related Reference Architectures that are in place for sectors and domains linked to DEMETER. This analysis, together with the technical requirements extracted (briefly discussed in section 11), drive the design and development of the DEMETER Reference Architecture.

Section 8 draws links between DEMETER and the Reference Architectures presented in the state of the art, presenting a short summary of the reasons why selected frameworks have been taken into account to build the DEMETER Reference Architecture.

Section 9 gives an overview of the technical requirements extracted by WP2, WP3 and WP4. In this section, only a summary of the various requirement classes is provided, distilled from the detailed work done on cataloguing the requirements by the corresponding Work Packages. It is left to the corresponding WP2/3/4 deliverables to present the exhausting list of specific technical requirements elicited.

Section 10 presents the main concepts of the DEMETER architecture. Key among these are the DEMETER Stakeholders Open Collaboration Space and Agricultural Interoperability Space, both accessed by users through the DEMETER Dashboard, as well as the DEMETER Enabler HUB which provides all resources needed for the development of DEMETER-enabled applications, as well as the integration and deployment of the DEMETER tools and enabling technologies.

Section 11 presents the DEMETER Reference Architecture through a number of viewpoints:

- subsection 11.1 gives a high-level view of the entire architecture elaborating on an indicative instantiation example;
- subsection 11.2 presents the functional viewpoint of the RA and described the main components of the DEMETER system;
- subsection 11.3 presents the RA's process view and explains the system processes and how the components communicate to deliver the foreseen features, functionality elements and mechanisms;
- subsection 11.4 describes the data viewpoint, highlighting the data flows and the

components needed to manage the main data processes, such as storage, retrieval, processing and security management;
- subsection 11.5 presents the deployment viewpoint mainly dealing with the topology and connections of software components on the physical layer when applications are deployed;
- subsection 11.6 presents the business viewpoint of the architecture, which will guide the application development and support the decision-making process.

Section 12 presents the main interactions and dependencies between the core components of the Reference Architecture previously detailed and attempts a high level description of the interfaces that need to be in place between these components.

Section 13 presents specific instantiations of the RA for all DEMETER pilots. As part of this section, a list of the various DEMETER entities that need to be implemented in order to enable the delivery of the targeted DEMETER applications for each pilot is provided.

Section 14 presents the GDPR concerns and guidelines that will need to be enforced in order to protect the data and privacy of the various stakeholders using the DEMETER system.

Finally, Section 15 concludes the document and Section 16 provides the respective references used.

The architecture presented in this deliverable will be complemented and completed by the following deliverables as they become available:

- D2.1 DEMETER data models and semantic interoperability mechanisms (April 2020)
- D2.2 DEMETER data and knowledge extraction tools (May 2020)
- D3.2 DEMETER technology integration tools (June 2020)
- D4.1 Decision Support, Benchmarking and Performance Indicator Monitoring Tools – Release 1 (May 2020)
- D4.2 Decision Enablers, Advisory Support Tools and DEMETER Stakeholder Open Collaboration Space (June 2020)

The revised version of the DEMETER Reference Architecture is planned for release on February 2021 and will be presented in D3.3.

5 Architecture design methodology

In DEMETER a reference ICT architecture and semantic data model based on the ISO/IEC/IEEE 42010 International Standard is used for representing services and IoT entities and that is transversal to many domains. This Standard, entitled as *"Systems and software engineering - Architecture Description"*, was published in 2011 as the result of a joint of the ISO and IEEE revision of the earlier IEEE Standard 1471-2000[1]. The first edition provided a conceptual model of *architecture description* and *best practices for the same*. The current edition refines the first one and adds requirements on *architecture frameworks* and *architecture description languages*.

The ISO Standard above is based upon a meta model of the terms and concepts pertaining to an architecture description that is presented using UML class diagrams to represent classes of entities and their relationships. Essentially, it covers the output of an architecting process without providing any help on how to construct such a process.

This Standard defines an architecture framework as the conventions, principles and practices for the description of architectures established within a specific domain of an application and community of stakeholders. The fundamental goal of the architecture framework is to codify a common set of architecture practices within a community for the sake of understandability, commonality, synergy and interoperability. The Standard also defines minimum requirements on any framework that are expressed in terms of the conceptual model of Architecture Description:

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

A good architecting process should facilitate early in the overall development process the discussions about what is feasible, hard, costly, etc. In practice, it is best done concurrently with systems analysis and requirements definition activities, resulting in a set of requirements and in an architecture that meets those requirements. The inputs to this architecting process will be the system/software requirements (potentially in draft form), the system/software operational concepts, and the list of potential stakeholders. The outputs will be an architecture description that conforms to the Standard, the results from reviewing it and the architecture that it describes, and also the possibly updated requirements reflecting different architectural decisions.

DEMETER follows the four main steps in this architecting process:

1. **Stakeholder/Concern identification**. First there is a listing with all the potential stakeholders and their concerns for the system and for the architecture of the system. It will be refined and taken back to the stakeholders to be validated. As part of this process it is desirable to extract requirements that need to be addressed in order to satisfy the needs of the stakeholders.

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

2. **Viewpoint development**. The next step is to figure out how to best answer the questions listed as the stakeholders' concerns. There are two parts to this: what the answers are (*views*) and how they can be captured (*viewpoints*). Within the Standard there is an explicit separation of both of them that allows building reusable viewpoints. A well-defined set of viewpoints, reviewed by stakeholders and developers, should make it easier to capture architectural decisions.
3. **View development**. This is basically dependent on what is required by each viewpoint. As the view is developed it is important to capture rationale for the key decisions and to include them in the architecture description. While we will not explicitly separate views from viewpoints in this document, when presenting the various viewpoints of the architecture we will keep this methodology in mind.
4. **View integration and evaluation**. As part of the integration of the architecture each viewpoint should be re-evaluated, particularly, any rules about cross-view consistency. It has to be verified that each view correctly implements the viewpoint and that its contents fully cover the system as a whole from the perspective of that view. The architecture description should undergo some sort of evaluation of the architecture being described against stakeholders' concerns and related considerations. This is actually addressed in Demeter via an internal review process that will iron out any inconsistences in the original viewpoint descriptions.

Applying ISO/IEC/IEEE 42010:2011 methodology for the architecture description of systems as DEMETER itself, involves specifying the architectural viewpoints that address stakeholders' concerns formulating their requirements and creating for them consistent architectural views using architectural models. The following viewpoints, briefly outlined, are the ones chosen to best derive DEMETER´s correspondent architectural views:

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

For documenting the architecture viewpoints, it is necessary to keep a list of mandatory and optional questions that need to be answered. These concerns/topics that need to be addressed are the following:

1. Viewpoint name.
2. Viewpoint overview: a brief overview of this view and the information it presents as well as its key features or a high level view of its operations.
3. Typical stakeholders: a list of the stakeholders expected to be users of views using this viewpoint.
4. Model kinds or diagrams: identify each model kind specified by the viewpoint; alternatively provide a diagram instead of a model that outlines the components. The International Standard does not specify one style for documenting these and it may be achieved in a number of ways: with a UML diagram, a meta model, a model template, a language definition or by some combination of these methods.

In addition to the previous mandatory information that should be provided, depending on the viewpoint, the following might be added:

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

There are important lessons taken into account and learned from the development and use of the ISO Standard. They may be summarized as falling into the following general areas:

1. **Ontology-based**. The Standard used is built upon an explicit conceptual model or ontology. To be useful, an architecture framework should be useable, that is, it must be understandable and presented in a form that can be acted upon. The first ingredient of understandability is a crisp and clear conceptual foundation and the terms reflecting it.
2. **Interest-driven**. Complex entities have a multitude of interested stakeholders each one with specific interests or concerns that, once identified, can serve as a key index into a successful architecture description.
3. **Open and extensible**. Architecture frameworks should be designed to be open and extensible as a system. One important lesson learned from architecture framework development is that defining the ontology of a given domain of interest will never be finished.
4. **Framework as a foundation**. A robust architecture framework can serve as a foundation for all aspects of architecting beyond its central role in architecture description. This means that a framework has implications for methods, processes, and tooling.
5. **Governance**. Robust conformance is a key means to achieving the usability and interoperability of architecture descriptions and should support extension, reusable model kinds and methods, as well as end-product architecture descriptions. Currently conformance is defined in terms of meta model consistency.

6 Master the complex Digital Platforms and Reference Architecture landscape

Any initiative promoting Digital Transformation in the agrifood sector (such as in many other industries) needs to specify and provide digital frameworks, to foster vendor-neutral data exchange, business-oriented organisation of information, and the assignment of responsibilities for data and service management. This is a pillar for value chain information sharing and exploitation practices, with relevant economic and legal implications involved in data ownership and confidentiality.

Digital Platforms (DP) should provide data-driven mechanisms and solutions to ease access and exploitation of data, fostering data economy and digital business. They should also provide vertical and horizontal interoperability[2] to boost technological diffusion, in order to create new services and applications, throughout the whole value chain, and potentially create new markets or extend/generalise the current ones.

There are hundreds of Digital Platforms available for the development of intelligent systems, mainly supporting both vertical and horizontal interoperability among datasets, services and applications. The choice dilemma is characterizing the decision-making processes of many business owners and system designers, evaluating the right platform(s) to solve the underpinned business needs. The dimensions of these needs that must be considered are: their scope, maturity, ownership of their components, and standards supported, implied business models, and many others. In such a scenario, more and more technical solutions are being brought to the market by Open Source Software communities and eco-systems.

DEMETER, as many other Large-Scale Pilots (LSP) in Europe, participates in and navigates quite a wide landscape of existing Digital Platforms, both in terms of Reference Models and Reference Architectures. This landscape is characterized by a wide set of stakeholders, being both technology and solutions providers. DEMETER will master this complexity, trying to create a strong synergy among data, service and platform providers, targeting both vertical and horizontal interoperability.[2] As defined given the direction taken by the DEM, the requirements for interoperability specify user-centric approaches and thus use cases (i.e., the 20 pilots involved in DEMETER) are the first-hand buyers and main drivers for those data interoperable specifications and available/developed solutions.

This section introduces the principles and main examples of Reference Model and Reference Architectures, coming mainly from the agrifood domain (and other relevant ones). The objective is to provide a baseline representation of the European (and worldwide) domain within which DEMETER is acting. This will be useful later on (together with the SotA presented in Section 7) in the design of the DEMETER Reference Architecture.

6.1 Main Rationale

In order to achieve both vertical and horizontal interoperability, a lot of concepts (e.g., model, vocabularies, viewpoints, and so on) need to be agreed upon by the main stakeholders in order to ensure a common understanding of such concepts; this is also compliant with the methodology presented in the previous section. Moreover, given the need to be able to deal with a very large

[2] Meaning respectively that applications share data automatically and that data can move from one entity to another entity at the same or higher level of the application.

range of pilot systems architectures, it is also necessary to define High Level Reference Architectures (HLA), which are meant to act as a blueprint for the design and development of pilots' systems and components, while at the same time supporting various purposes such as:

- To communicate on a common view and language;
- To support the analysis and evaluation of different actual implementations of the same abstract Reference Architecture;
- To integrate various existing state-of-the-art approaches and technologies into one model;
- To support the transition from an existing legacy architecture to a new architecture;
- To help assessing conformance to identified standards or interoperability requirements;
- To document decisions taken during the development process of a system.

6.2 Considerations for platforms selection and usage

There are hundreds of Digital Platforms available for the development of complex and distributed systems. The question of a choice of platform(s) by system designers is complex and may be related to the several functionalities the platform is providing. The following dimensions should be considered:

- **IoT Reference Models**. Some of the existing IoT platforms may address a specific problem or a limited technical environment, offering a point solution addressing only a part of the IoT stacks. On the other hand, some platforms can be very general purpose and integrate the IoT system in a larger (enterprise) system. These models prescribe platforms providing services for the IoT (devices and data) management.
- **Big Data Frameworks**. These frameworks address large and complex data sets, where the issues come from having large volumes of data with a lot of variety, that must be collected, updated and processed very fast, while also verifying their quality and trustworthiness. To address all these issues, we need a range of technologies for capturing, managing, processing, analysing, visualising and communicating the data. These frameworks are therefore designed in such a way as to manage the big data collection, processing and analysis chain.
- **Interoperability Platforms**. These platforms combine a number of technologies together, such as IoT, Big Data or Cloud architectures (e.g., FIWARE). They also deal with data exchanges and management, including how to monetize data services. In fact, some platforms (like IDSA) focus and put emphasis on information ownership, with the aim of enabling clear and fair exchanges between data sources and consumers.
- **Sector-specific Reference Architectures**. These architectures come from the IT and Internet industry and put especial emphasis on the business viewpoint which sits on top of the rest of the architecture. Overall, in every other respect of note, they follow similar principles to the rest of the models in the previous frameworks, with physical devices sitting at the bottom of the architecture, integration and communication on top of them linking to the software and its functional components, and at the top having the final application which implements and satisfies the business goals of the whole system.
- **LSP Reference Architectures**. The Reference Architectures come from large scale recent projects with many pilot applications each. They adapt their research architecture from RA previously available (and which are presented in previous sections of the State-of-the-Art review in this deliverable), such as those coming from IoT reference models and from the big data

frameworks. Then, they customize these architectures, as their goal is to deploy systems based on these RA to tackle large scale (and many in number) pilot applications.

In the next section, we present the main state-of-the-art projects in these aforementioned dimensions. These architectures have been developed in various Projects, Standardisation organisations or Alliances.

7 Related State of the Art Review

7.1 IoT Reference Models

7.1.1 IoT-A

In order to solve the interoperability problem, which is essential in IoT based systems, IoT-A (Carrez, 2013) proposed the Architectural Reference Model (ARM) concept. The IoT-A ARM model consists of five different views of sub-problems.

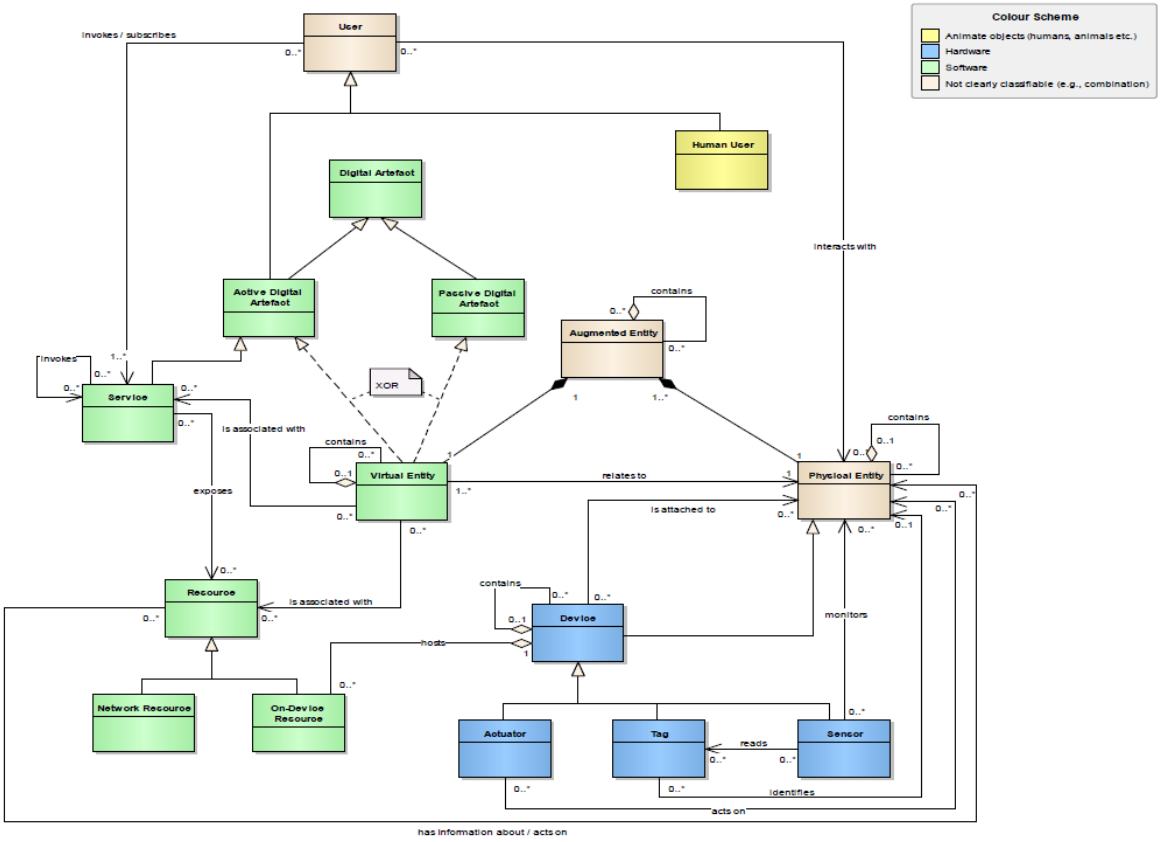


Figure 1. IoT-A Domain Model

The IoT-A domain model defines a set of high-level components or actors. These components are common to most IoT use-cases and are depicted in the figure above.

The information model analyses specific common attributes for each of the entities of the domain model and their semantics, such as defining that a VirtualEntity could include properties like entityType or the identifier and attributes including name, type and several values.

The IoT-ARM, from the domain model, then introduces the FGs (Functionality Groups) as part of the functional model. Each of these FGs is further decomposed into several Functional Components (FC). These are all presented in the functional view diagram (depicted in the figure below). This figure depicts the Functional Components (FCs) for each Functionality Group (FG) of the IoT-A ARM, which is called the IoT Functional View.

In order to handle the specific characteristics of IoT communication flows, specific FCs and FGs are included. The Communication Model relies on the previous described components of the architecture and on ISO OSI Reference model, in order to address interoperability at the stack or network level.

A number of security issues, such as Trust, Security and Privacy Model are also depicted.

Trust must be enforced in all layers and elements of the system. IoT-A ARM established concepts such as Trust-model domains, Trust-evaluation mechanisms, Behavioural policies, Trust anchors, Federation of trust or M2M support.

Security is considered at 2 different levels:

- Communication, which provides an abstract approach that constrained devices are specifically targeted. These devices are the most challenging elements of an IoT based system. Security for them can be assigned to the Corresponding Gateway, which has to implement a constrained device security feature.

- Application. IoT-ARM model covers the basic steps in order to make the analysis that will communicate with the identification of the design policies and actions for mitigating, such as description of elements for protection, categorize risk sources, proposal of design choices and risk assessment.

Privacy issues are enforced, for example, through a Functional Component (FC) that manages the identities of all the actors of the platform. This element in collaboration with the Authentication component enhances security policies in the different resources and endpoints of the system.

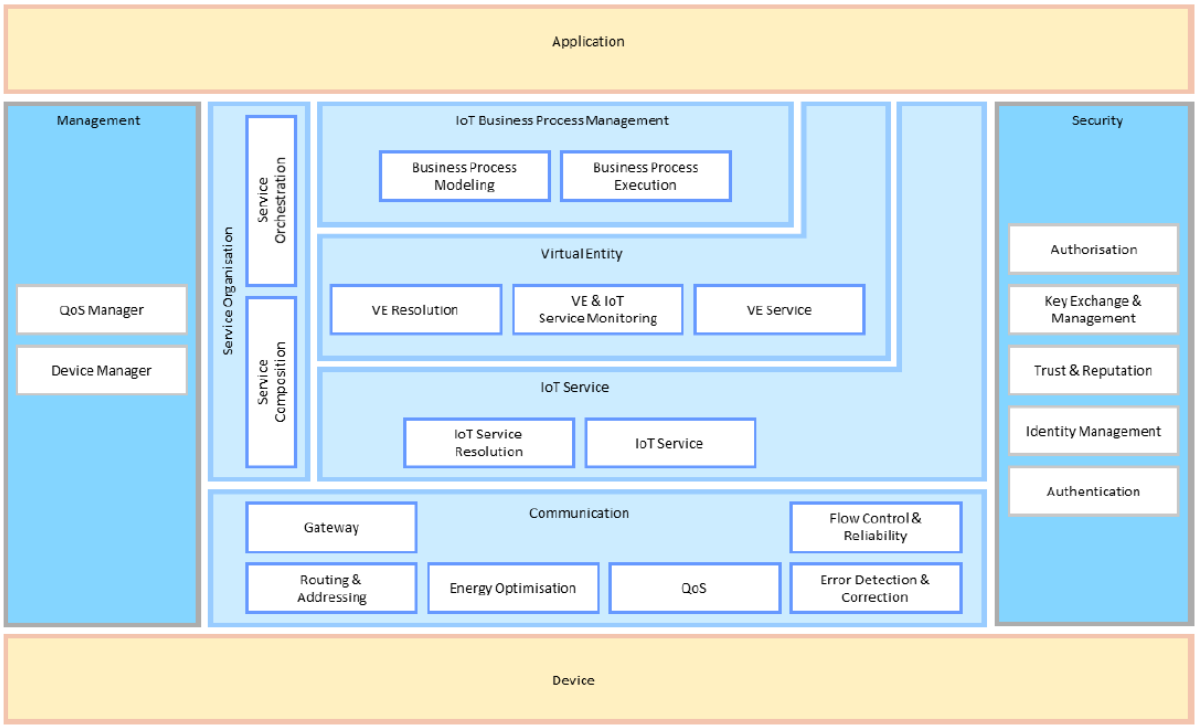


Figure 2. Functional View of IoT-A Architectural Reference Model

7.1.2 AIOTI

The Alliance for the Internet of Things Innovation (AIOTI)[3] has specified a high level Reference Architecture (AIOTI, 2018) that maps to several other dominant and/or standardised IoT architectural approaches, such as ITU-T[4], oneM2M[5], Industrial Internet Consortium (IIC)[6], RAMI 4.0[789], Big Data Value Association (BDVA)[10], National Institute of Standards and Technology (NIST)[11], etc.

Based on the IoT-A domain model described in section 7.1.1, they have derived the AIOTI Domain Model depicted in the figure below.

[3] Alliance for the Internet of Things Innovation (AIOTI): https://aioti.eu/
[4] ITU-T FG-DPM, ITU-T Focus Group on Data Processing and Management to support IoT and Smart Cities & Communities, https://www.itu.int/en/ITUT/focusgroups/dpm/Pages/default.aspx
[5] oneM2M, "oneM2M Functional Architecture Baseline Draft", oneM2M-TS-0001, 2014.
[6] Industrial Internet Reference Architecture, http://www.iiconsortium.org/IIRA.htm
[7] VDI/VDE GMA, ZVEI: Status Report - Reference Architecture Model Industrie 4.0 (RAMI 4.0), July 2015, https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report__Reference_Archtitecture_Model_Industrie_4.0__RAMI_4.0_/GMA-Status-Report-RAMI-40-July-2015.pdf
[8] DIN SPEC 91345:2016-04 – Referenz architektur modell Industrie 4.0 (RAMI 4.0), April 2016, http://www.din.de/de/ueber-normen-und-standards/din-spec/din-specveroeffentlichungen/wdc-beuth:din21:250940128
[9] IEC PAS 63088:2017 Smart manufacturing - Reference architecture model industry 4.0 (RAMI 4.0), March 2017, https://webstore.iec.ch/publication/30082
[10] Big Data Value Association, European Big Data Value Strategic Research and Innovation Agenda, http://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf
[11] NIST big data interoperability framework, http://bigdatawg.nist.gov/V1_output_docs.php

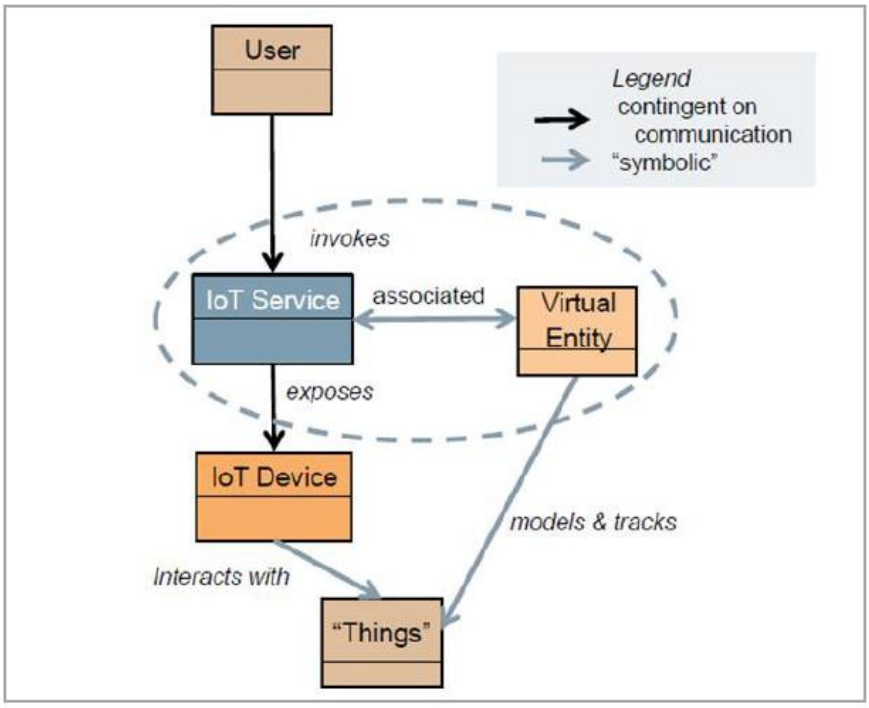


Figure 3. The AIOTI Domain Model

The main concepts and relationships are captured by the domain model at the highest level. Identification and naming of these concepts and relationships make a common dictionary for the domain, moreover they are functional for all other taxonomies and models. There is interaction of the user, which can be human or otherwise, with a physical entity, a Thing. This kind of interaction can be intervened by an IoT Service, which is associated with a Virtual Entity, a physical entity or a digital representation. The Thing and the IoT Service interact with each other, though an IoT Device that exposes the capabilities of the actual physical entity.

As it is depicted in the figure below, the functional model of AIOTI consists of three basic layers. We have to note in this point that the term "*layer*" refers to software architecture logic. So, by saying layer, we mean a group of modules that offer a tight set of services.

The AIOTI functional model describes, as depicted below, interfaces and functions between functions of the IoT system. As it shown, functions are: **1)** *App Entity* is an entity existing in the application layer that implements IoT application logic. **2)** *IoT Entity*, which is an entity with the aim to expose IoT functions to App Entities via the interface 2 or to other IoT entities via interface 5. **3)** *Networks function* that consists of different network technologies (such as PAN, LAN, WAN). It includes different interconnected administrative network domains.

A Device can include an App Entity and a Network interface. For example, it can use an IoT Entity. This example represents a constrained device. But other devices can implement an App Entity, an IoT Entity and a Network interface. The interfaces that are shown in the figure above are:

1. it specifies the structure of the data that are exchanged between App Entities.
2. it enables access to services exposed by an IoT Entity.
3. it enables the exchanging of data across the Networks to other entities.
4. it enables the requesting of network control plane services.

5. it enables the requests and the exposures of services that coming (form)/going (to) IoT Entities.

AIOTI High Level Architecture realizes the digital representation of physical things in the IoT Entities. Those representations help in discovering things of App Entities and enable related services, for example actuation or measurements. In order to realize interoperability, the representation of things contains data and metadata.

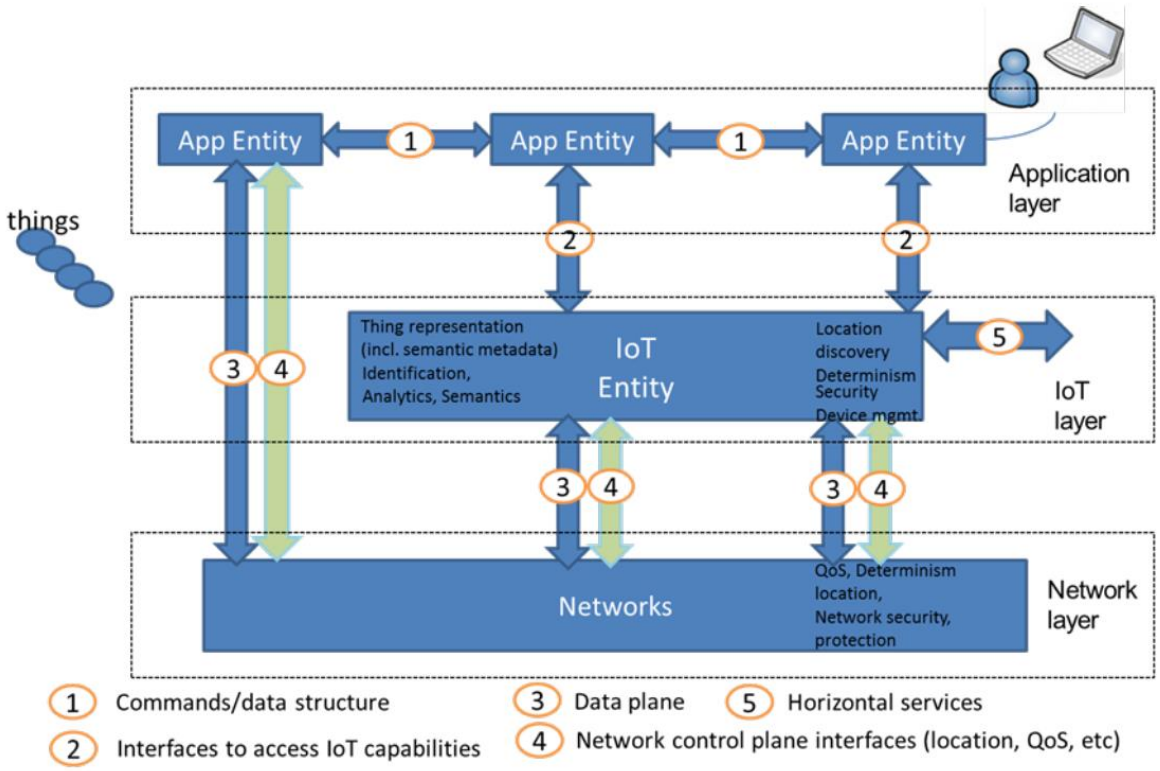


Figure 4. The AIOTI HLA functional model

7.2 Big Data Frameworks

Big Data (Big Data (Wikipedia)) refers to large and complex data sets; precision farming is the main initiative under which big data technologies are being employed in the agriculture domain and these build on data such as geo-coded maps of agricultural fields, weather data as well as the real-time monitoring of farm activities (through various sensors in the farm, cameras, data from various actors etc.) in order to increase the efficiency of resource use, and ultimately increase crop yield without wasting resources.

The main challenges when processing big data come from:

- their *volume*: global data size roughly doubles every year
- their *variety*: data can take the form of text, image, audio, video as well as fused data sources
- the need for *velocity*: data must be collected and processed in a timely manner, often in real time or near to real time.

Additional issues arise from their *veracity* (referring to the quality and trustworthiness of the data, e.g., malicious or false data may be input) and their *value* (the ability to transform them into business and get them monetized).

To address all these, we need a range of technologies for capturing, managing, processing, analysing, visualising and communicating the data. Data preparation such as collecting, curating and organising data, accounts for up to 80% of the work before any data analytics may be applied. This is the reason why it is necessary to have appropriate platforms and frameworks to manage the big data collection, processing and analysis chain.

7.2.1 BDVA Reference Architecture

One such generic big data system architecture is presented in the figure below, which depicts the BDVA Reference Architecture (European BDVA Strategic Research and Innovation Agenda v4.0, 2017); this is a reference framework made by the European BDVA (Big Data Value Association) that describes logical components of a generic big data system.

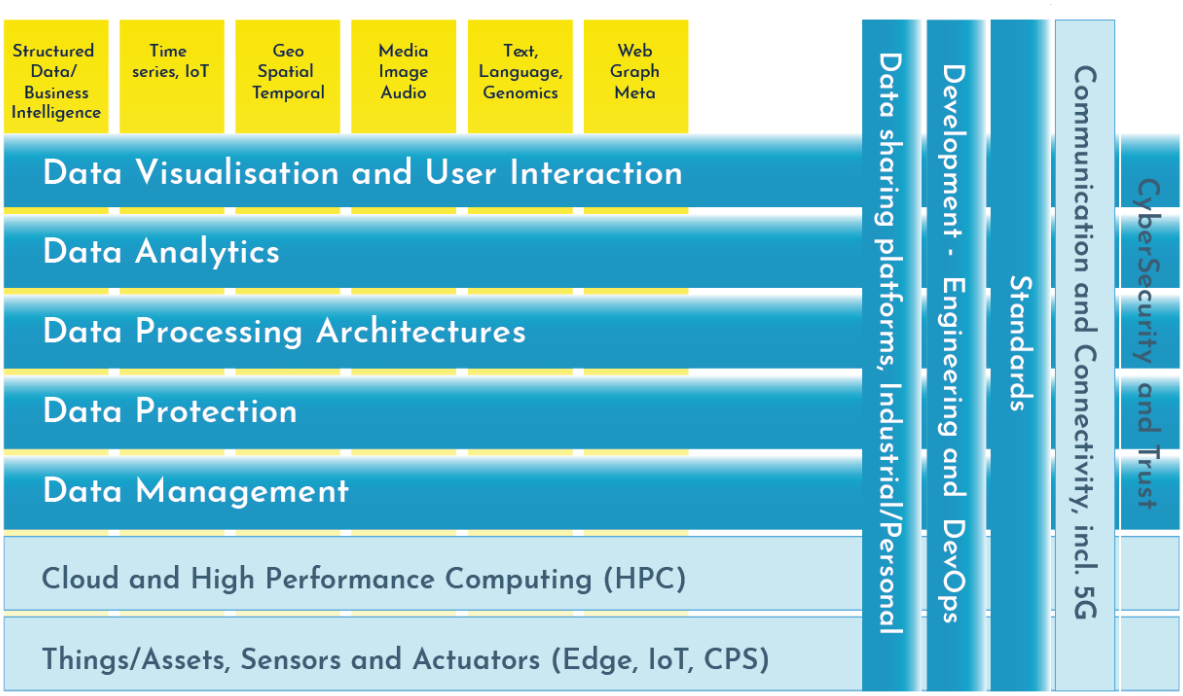


Figure 5. The BDVA Reference Architecture

In fact, BDVA has proposed their initiative regarding a European Data-Driven Artificial Intelligence and their vision regarding AI and Big Data and how it can drive the European technology and economy. (Data-driven Artificial Intelligence For European Economic Competitiveness and Societal Progress. BDVA Position Statement, 2018) To realize this vision, it will be necessary to address a number of challenges: a) data-driven AI-based solutions for the industry will require new business models, b) trust in AI and its results must be established; for example one should be able to explain how AI applications came to a specific result ("Explainable AI"), which would foster responsible technological development (e.g., avoid bias) and enhance transparency in how and why an AI takes a decision, c) it is necessary to develop an AI and Big Data ecosystem, by developing data for open AI

platforms and overcoming the lack of data interoperability, and d) fuse and develop a number of technologies, as a successful industrial AI relies on the combination of a wide range of technologies, such as advanced data analytics, distributed AI, and hardware optimised for AI. To this end, recently BDVA (together with euRobotics) has proposed the creation of an AI PPP (public-private partnership). (Joint Vision Paper for an AI Public Private Partnership (AI PPP). Brussels: BDVA – euRobotics, 2019).

7.2.2 NIST Big Data Reference Architecture

The National Institute of Standards and Technology (NIST) has published another Reference Architecture for a big data interoperability framework. (NIST Special Publication 1500-6. NIST Big Data Interoperability Framework: Volume 6, Reference Architecture, 2015). This framework defines data in a broad level and service use flows between the components of the framework, signifying needs for applications interfaces. This architecture is depicted in the figure below.

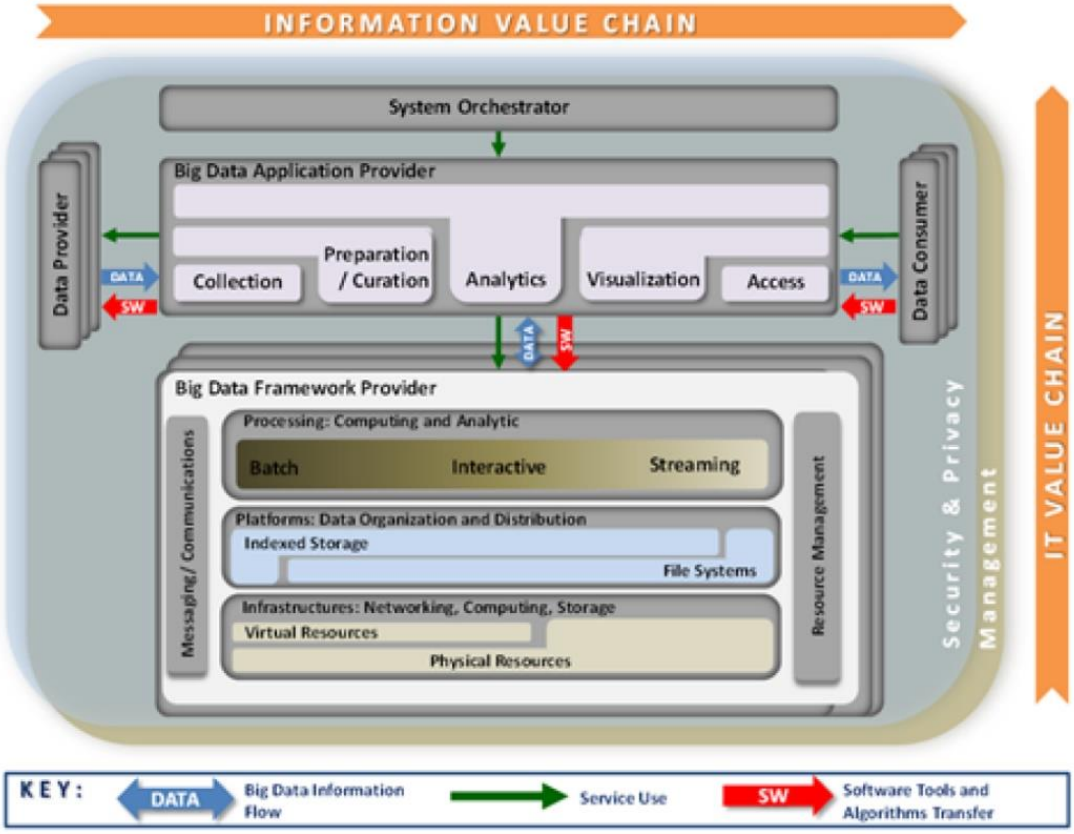


Figure 6. NIST Big Data Reference Architecture

The NBDRA (NIST Big Data Reference Architecture)[12] consists of five main components, shown in Figure 8, that represent different technical roles that exist in every Big Data system. These functional components are:

[12] NIST Big Data Interoperability Framework: Volume 6, Reference Architecture, Final Version 1, NIST Big Data Public Working Group Reference Architecture Subgroup, p.10-15, http://dx.doi.org/10.6028/NIST.SP.1500-6

- System Orchestrator: Defines and integrates the required data application activities into an operational vertical system.
- Data Provider: Introduces new data or information feeds into the Big Data system.
- Big Data Application Provider: Executes a data life cycle to meet security and privacy requirements as well as System Orchestrator-defined requirements.
- Big Data Framework Provider: Establishes a computing framework in which to execute certain transformation applications while protecting the privacy and integrity of data.
- Data Consumer: Includes end users or other systems that use the results of the Big Data

The two fabrics shown in Figure 6 encompassing the five functional components are:

- Management
- Security and Privacy

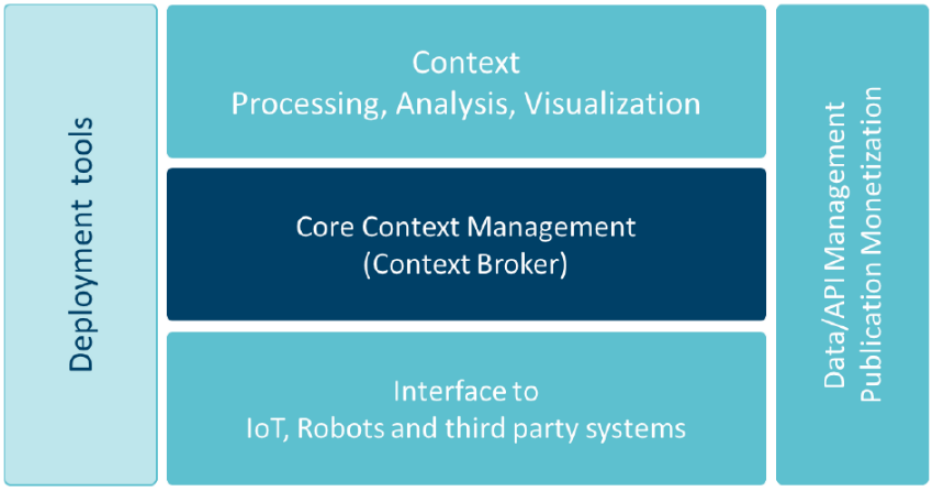
These two fabrics provide services and functionality to the five functional components in the areas specific to Big Data and are crucial to any Big Data solution.

7.3 Interoperability Platforms

7.3.1 FIWARE

FIWARE[13] aims to create a platform that uses a combination of technologies such as IoT, Big Data or Cloud architectures. FIWARE is open source and eases the creation of new applications in multiple verticals since it contains a rich set of components that can be deployed and connected in a compliant way.

FIWARE tries to distribute the data and vertical silos in many IoT based systems by using a horizontal layer so that it manages large-scale context information. FIWARE NGSI defines an information model and a RESTful interface can be constructed by context data providers and consumers. The latter can be realized by interacting with the Context Broker, which is a central component of FIWARE architecture. The Context Broker aims to enable the system to make updates and access to the current state of context.



[13] www.fiware.org

Figure 7. FIWARE overall architecture

FIWARE includes specifications of the interfaces of multiple Generic Enablers (GE) as depicted in the figure above. (FIWARE developers page) The FIWARE Catalogue (FIWARE developers catalogue) is a curated framework of open source platform components which can be assembled together and with other 3rd Party platform components to accelerate the development of Smart Solutions.

The main and only mandatory component of any "Powered by FIWARE" platform or solution is the FIWARE Orion Context Broker Generic Enabler, which brings a cornerstone function in any smart solution: the need to manage context information, enabling to perform updates and bring access to context.

Building around the FIWARE Context Broker, a rich suite of complementary FIWARE components are available, dealing with:

- **Interfacing with the Internet of Things (IoT), Robots and 3rd Party systems**, for capturing updates on context information and translating required actuations.
- **Context Data/API management, publication, and monetization**, bringing support to usage control and the opportunity to publish and monetize part of managed context data.
- **Processing, analysis, and visualisation of context information** implementing the expected smart behavior of applications and/or assisting end users in making smart decisions.

The catalogue contains a rich library of components with reference implementations that allow developers to put into effect functionalities such as the connection to the Internet of Things or Big Data analysis, making programming much easier. FIWARE core platform model facilitating IaaS and SaaS required of application domains, on this basis GEs applications achieve already defined standards, provide APIs for interoperability, represent application domains or design granularities. We can think to a GE as Macroscopes were highest level interface is a simple controller providing a wide and in scope view of operations (attributes control functions of a system); GEs from different domains are Macroscopes on the domain: implement abstract Macroscopes concretely and provide API access via REST HTTP to trigger GE behaviour. Modelling a GE is identified within UML use cases. GE specification have some properties:

- Addressing: IP address and port numbers
- Recognition: control syntax, parser, interpreter and semantic rules
- Multimodal: APIs, protocols, drivers
- Structured Data: XML, JSON, IC
- Formal Operation: state machine, dispatcher, DOM nodes
- Ad hoc Network Communication: HTTP/s, request methods, asynchronous, client/server, URIs
- Modular Design: object-oriented architecture, methods and functions, listeners, callbacks
- Behavioural: multithreaded, parallel, imperative, result combining, verifying, transformation, bidirectional coms
- Security: channel encryption, message encryption, authentication, authorization
- HCI: GUI, hardware interaction, multimodal UI, accessibility, human actors (Doctors, Patients, Staff, …)

- Interoperability: networked API server, configuration parameters, legacy system integrators, RPC, REST

The ETSI Industry Specification Group for cross-cutting Context Information Management (ISG CIM) (Industry Specification Group (ISG) cross cutting Context Information Management (CIM)) has just released a preliminary specification of an API considered to be cornerstone in the development of Smart Cities, Smart Agrifood or Smart Industry applications.

In any smart solution there is a need to gather and manage context information, processing that information and informing external actors, enabling them to actuate and therefore alter or enrich the current context. Group Specification CIM 004, referred to as the ETSI NGSI-LD API specification, defines a simple way to update or query context data within a Smart Application, including factors such as the source, meaning, licensing, or related information describing that data.

The ETSI ISG CIM has decided to give the name "NGSI-LD" to the Context Information Management API to reinforce the fact that it leverages on the former OMA NGSI 9 and 10 and FIWARE NGSI specifications, incorporating the latest advances from Linked Data. The FIWARE Context Broker component (Orion), the core component of any "Powered by FIWARE" platform or solution, provides the open source reference implementation of the FIWARE NGSI API and will evolve to work as open source reference implementation of the new ETSI NGSI-LD API specifications.

The implementations of NGSI-LD which are available are: Orion-LD[14], Scorpio[15], and Djane[16].

In the figure below, the FIWARE architecture on Smart Agriculture is presented, which is based on a platform powered by FIWARE. (Smart Agrifood - FIWARE Foundation Open Source Platform)

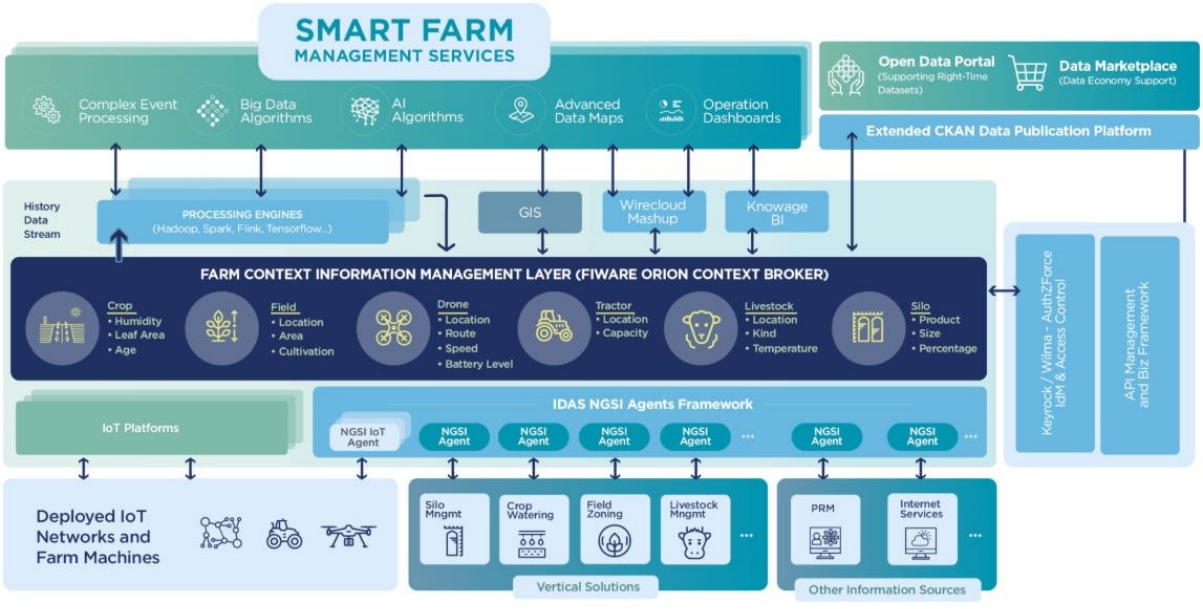


Figure 8. Reference architecture for Smart Farm Management System powered by FIWARE.

[14] https://github.com/Fiware/context.Orion-LD
[15] https://github.com/ScorpioBroker/ScorpioBroker
[16] https://github.com/sensinov/djane/

More specifically, the Orion Context Broker collects data from sensors, drones, vertical smart solutions and information systems. In this way, the broker breaks information silos. Sensors are connected to IDAS IoT Agents, so that they can handle many IoT protocols such as MQTT, CoAP/OMA-LWM2M, OneM2M. Also, alternative IoT platforms can be used for this situation. Fast RTPS is used to interface ROS-2 robots, which is the main communication middleware in ROS-2. Different processing engines, such as Flink, Hadoop and Spark, are used in order to process historical data, so as to extract valuable insights or derive smart actions. Artificial Intelligence or Complex Event Processing functions can be used above the integrated processing engines. Wirecloud web mashup framework is used for Operating dashboards. Extended CKAN portal can offer to 3rd parties part of the current and historic context data. The API/Data access control functions enable access to the context data to parties that own certain privileges. The API management and business support layer can offer auditing of the system and monetize data access.

7.3.2 International Data Spaces Association – IDSA

The International Data Spaces Association (IDSA) is the evolution of IDS (Industrial Data Space) which itself was an initiative lead by Fraunhofer ISST, in cooperation with ATOS, T-Systems, and the idea is promoted by the German Federal Ministry of Education and Research. IDSA is characterized by the focus on information ownership, with the aim of enabling clear and fair exchanges between data providers and consumers. To this end it suggests a reference distributed architecture that accomplishes this goal (IDS Reference Architecture Model Version 3.0.

Broadening the perspective from an individual use case scenario to interoperability and a platform landscape view, the IDS Reference Architecture Model positions itself as an architecture that links different cloud platforms through policies and mechanisms for secure data exchange and trusted data sharing (through the principle of data sovereignty). Over the IDS Connector, industrial data clouds, individual enterprise clouds, on-premise applications and individual, connected devices can be connected to the International Data Space ecosystem (see Figure 12).

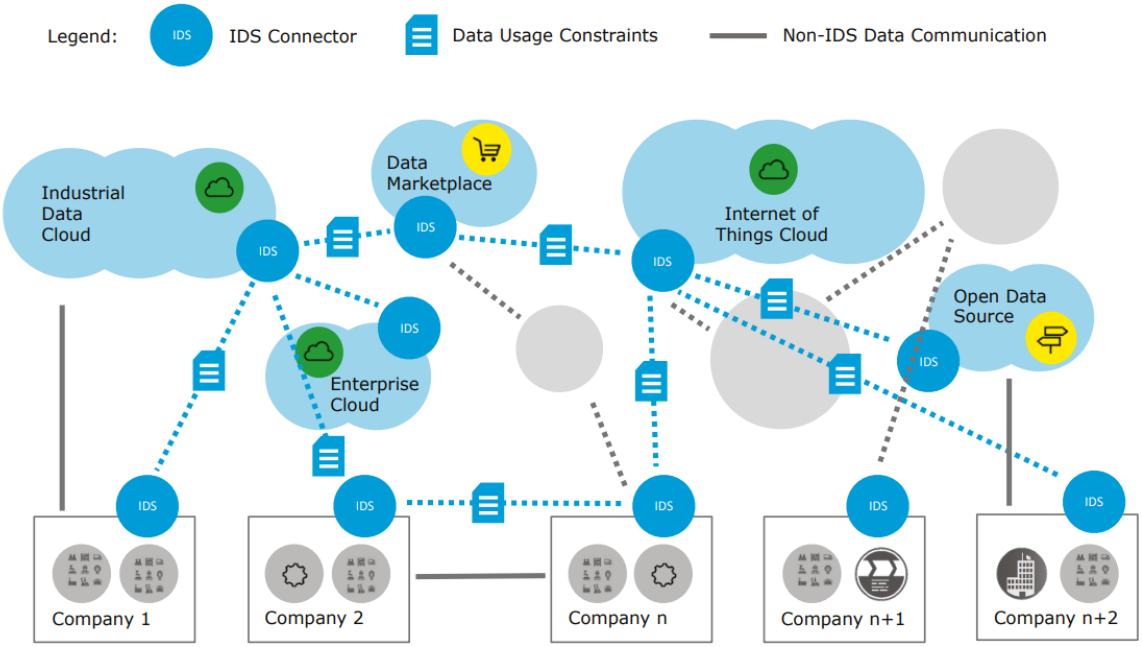


Figure 9. International Data Spaces connecting different platforms

This IDS Reference Architecture Model (IDS-RAM) is described using multiple layers, such as business, functional, process, information and system; between and common to all these layers are transversal functionalities that foster security, certification and governance, as illustrated in Figure 12. The Business Layer specifies and categorizes the different roles which the participants of IDS can assume, and it specifies the main activities and interactions connected with each of these roles. The Functional Layer defines the functional requirements of IDS, plus the concrete features to be derived from these. The Process Layer specifies the interactions taking place between the different components of IDS; using the BPMN notation, it provides a dynamic view of the Reference Architecture Model. The Information Layer defines a conceptual model which makes use of linked-data principles for describing both the static and the dynamic aspects of IDS' constituents. The System Layer is concerned with the decomposition of the logical software components, considering aspects such as integration, configuration, deployment, and extensibility of these components.

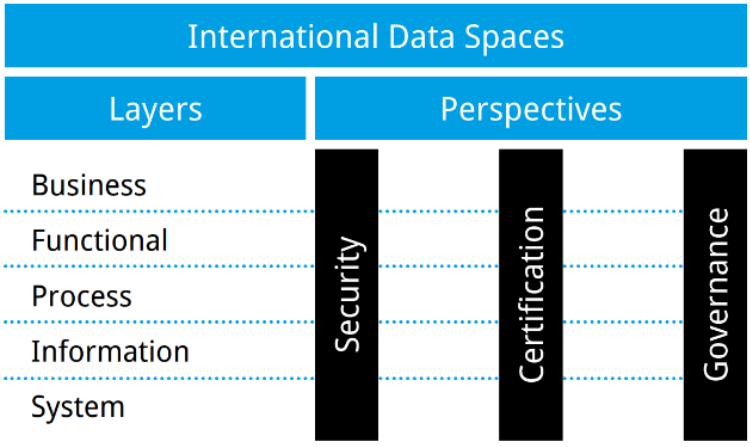


Figure 10. General structure of Reference Architecture Model

Comparing IDS to IoT-A ARM, the former focuses its specification of the roles for actors within the business layer that would govern the data flows between different domains or data spaces. As such, key participants (actors in the system) would be the Data Owner, Data Provider, Data Consumer, Data User or Broker Service provider. The complete landscape of roles, their functionalities and relationships result in a model depicted in the following Figure 13.

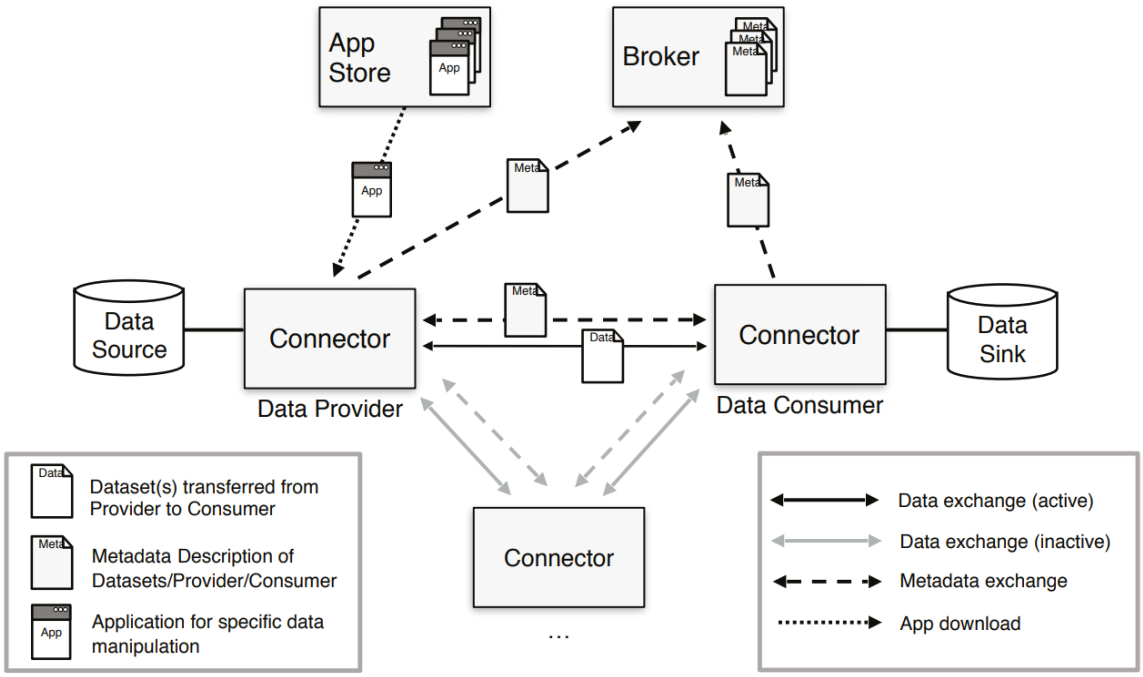


Figure 11. Interaction between technical components of IDS Reference Architecture Model

The **Connector** is the central technological building block of IDS. It is a dedicated software component allowing Participants to exchange, share and process digital content. At the same time, the Connector ensures that the data sovereignty of the Data Owner is always guaranteed. The **Broker Service Provider** is an intermediary that stores and manages information about the data sources available in IDS. The activities of the Broker Service Provider mainly focus on receiving and providing metadata that allow provider and consumer connectors to exchange data. The **App Provider** role is optional in IDS, and its main role is to develop applications that can be used by both data providers and consumers in the data space. Applications are typically downloaded from the remote app store, and run inside the containerized connector.

Establishing **trust for data sharing and data exchange** is a fundamental requirement in IDS. The IDS-RAM defines two basic types of trust: 1) Static Trust, based on the certification of participants and core technical components, and 2) Dynamic Trust, based on active monitoring of participants and core technical components. For data sharing and data exchange in the IDS, some preliminary actions and interactions are required. These are necessary for every participant, and involve a Certification Body, Evaluation Facilities, and the Dynamic Attribute Provisioning Service (DAPS). Figure 14 illustrates the roles and interactions required for issuing a digital identity in IDS, and these interactions are briefly listed here:

1. Certification request: This is a direct interaction between a participant and an evaluation facility to trigger an evaluation process based on IDS certification criteria.

2. Notification of successful certification: The Certification Body notifies the Certification Authority of the successful certification of the participant and the core component. Validity of both certifications must be provided.

3. Generating the IDS-ID: The Certification Authority generates a unique ID for the pair (participant and component) and issues a digital certificate (X.509).

4. Provisioning of X.509 Certificate: The Certification Authority sends a digital certificate (X.509) to the participant in a secure and trustworthy way and notifies the DAPS.

5. Register: After the digital certificate (X.509) is deployed inside the component, the component registers at the DAPS.

6. DTM Interaction: The Dynamic Trust Monitoring (DTM) implements a monitoring function for every IDS Component, and DTM and DAPS then exchange information on the behavior of the component, e.g., about security issues (vulnerabilities) or attempted attacks.

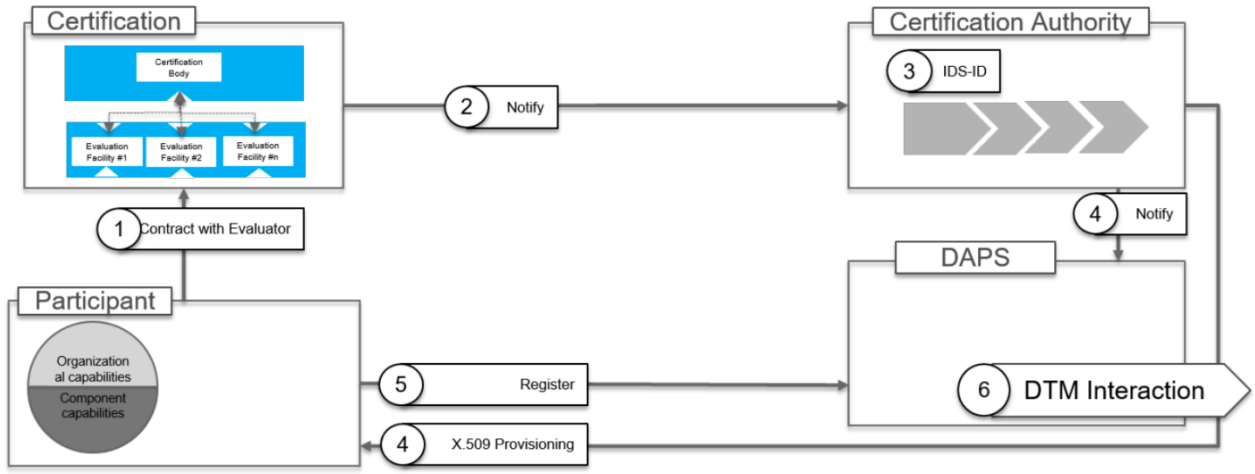


Figure 12. Interactions required for issuing a digital identity in the IDS

The IDS Reference Architecture contains an internal structure that is strongly supported by the containerization for the development of IDS connectors. It relies on IDS Communication Protocol to enforce security in data exchanges, as it is depicted in the figure below.

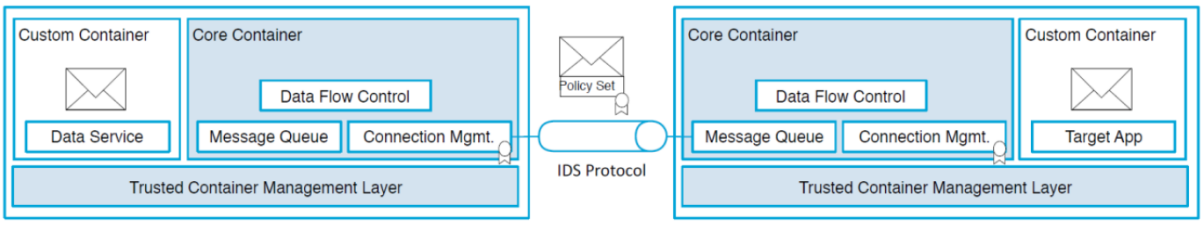


Figure 13. Enforcing security in data exchanges: the IDS Communication Protocol (IDS-RAM v3)

To sum up, the security implications that guarantee reliable and trusted transfer of information between independent entities in IDS are the following:

- **Secure communication.** The concept of Trusted Connector is introduced as depicted in the figures above.
- **Identity Management** for identification/authentication/authorization enhancing. There is use of certificates issued by a Certificate Authority (CA).
- **Trust Management** that uses Cryptographic methods such as PKI (Public Key

Infrastructures).

- **Trusted Platform** for trustworthy data exchange, which defines the minimal requirement for Security Profiles that should be verified by IDS connectors. It also defines the capacity to perform integrity verification of the rest of the involved connectors.
- **Data Access control**. IDS defines authorization criteria based on the previously defined Security Profiles.
- **Data Usage Control**. IDS checks and regulates that data processing is according the intended purposes defined by the original data owner.

7.4 Sector-specific Reference Architectures

7.4.1 Reference Architecture Model for Industry 4.0 (RAMI4.0)

The Reference Architectural Model Industrie 4.0 (RAMI 4.0) was developed by the Platform 4.0 in 2015 and focuses on the IoT and Cyber-Physical Systems (CPS) in the industrial manufacturing domain. RAMI4.0 is a three-dimensional model, which describes the Industrie 4.0 space and organises the life-cycle/value streams and the manufacturing hierarchy levels across the six layers of the IT representation of Industry 4.0.

Current Industrial Revolution driven by CPS and IoT is expected to have a major impact on the future of agriculture as well, as there is a natural relation between industry and agriculture. As an extension of Industry 4.0 a new concept can be introduced: Agriculture 4.0. Integration of machines and equipment, increased automation, efficient decisional process represents objectives of an agricultural enterprise facilitating the adaptability to climate and market dynamics and perturbations and allowing for sustainable, ecological and socially beneficial development. (Dumitrache, Sacala, Moisescu, & Caramihai, 2017)

One of the main objectives once adopted is to be able to communicate the scope and design of the system, to further collaboration and integration with other relevant initiatives by framing the developed concepts and technologies in a common model.

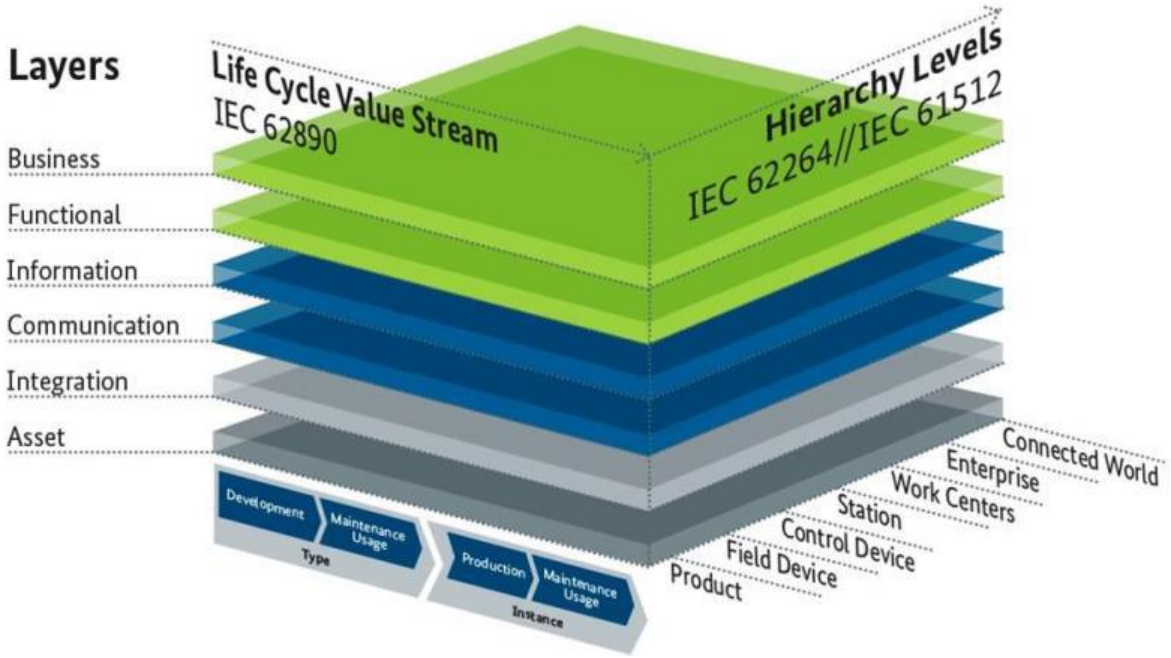


Figure 14. The three dimensions of the RAMI 4.0 (Source: Platform I4.0 and ZVEI).

The three-dimensional matrix can be used to position standards and describe use-cases. It addresses integration within and between factories, end-to-end engineering and human value-stream orchestration. This model is complemented by the Industrie 4.0 components and both have been described in DIN SPEC 91345. (Reference Architecture Model Industrie 4.0 (RAMI4.0) - DIN SPEC 91345:2016-04, 2016)

In RAMI4.0, each component consists of six layers. Starting with the lowest layer, the structure consists of asset, integration, communication, information, functional and business and represents a layered IT system structure, as shown in the figure below.

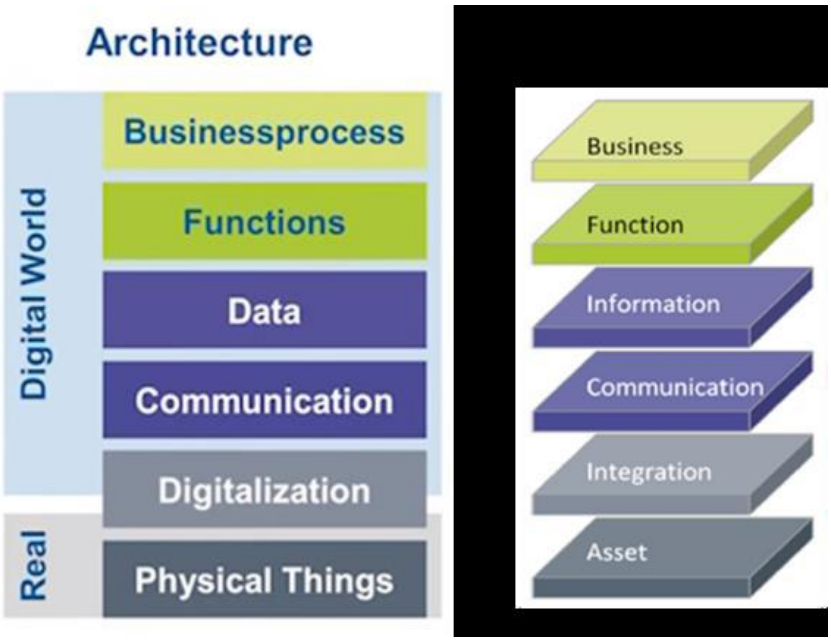


Figure 15. The IT Layers of RAMI 4.0 (Source: Platform I4.0 and ZVEI)

The function of each layer is:

- The asset layer describes physical components of a system, for example production equipment, product part, sensors, documents, as well as humans. For every asset represented in this layer there must be a virtual representation in the above layers. Among the physical assets, this layer includes the digital interface with humans and the relationship to elements in the integration layer.
- The integration layer deals with easy to process information content and can be considered as a bridge between the real and the IT world. It contains all elements associated with the IT, including field buses, HMIs, necessary to implement a function, as well as the properties and process related functions required to use an asset in the intended way and generates events based on the acquired information.
- The communication layer is responsible for the standardized communication between integration and information layer. Therefore, it performs transmission of data and files and standardizes the communication from the Integration Layer, providing uniform data formats, protocols and interfaces in the direction of the Information Layer. It also provides services to control the integration layer.
- The information layer holds the necessary data in a structured and integrated form and provides the interfaces to access this structured data from the functional layer. It is responsible for processing, integrating and persisting the data and events, as well as for describe the data related to the technical functionality of an asset. It can be considered the run-time environment for Complex Event Processing (CEP) where rule-based (pre-) processing of events takes place, data APIs and data persistence mechanisms. So, events are received from the communication layer, transformed and forwarded accordingly.
- The functional layer describes the logical and technical functions of an asset providing a digital description of its functions and a platform for horizontal integration of various functions; it also describes the business model mapping, business processes which can be adjusted based on inputs from the functional layer, providing models with runtime data of processes, functions and applications.
- The business layer is in charge to orchestrate the services provided by the functional layer. It maps the services to the business (domain) model and the business process models. It also models the business rules, legal and regulatory constraints of the system. The processes to ensure of the economy are located on this level.

In order to represent the Industry 4.0 or Agriculture 4.0 environment, the functionalities of IEC62264 have been expanded to include two new levels, at the bottom, the “product” (both the type and the instance, through the entire lifecycle) which are active elements within the production system due to their ability to communicate. They provide information on their individual properties and necessary production steps. At the top there is the “connected world”, which represents its outer networks or the ecosystem, e.g., collaboration with business partners and customers, suppliers or service providers, as well as Internet-based services.

This allows moving from the typical pyramid, with rigid hierarchical structures, to a composite of networked objects and systems as reflected in the figure below.

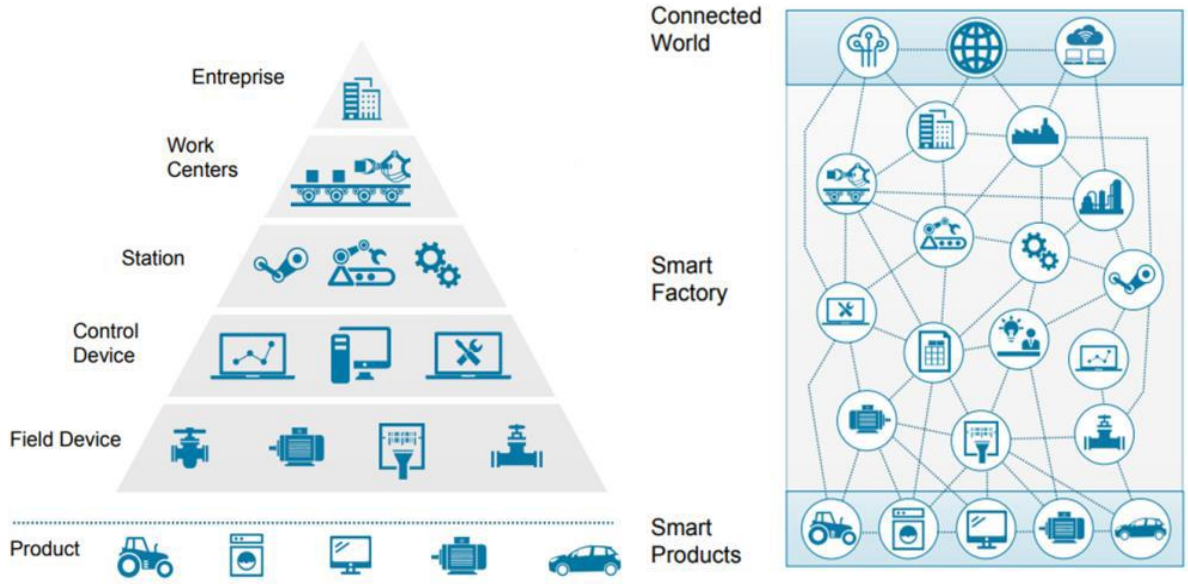


Figure 16. Hierarchy Levels of Industry 3.0 and RAMI 4.0 (Source: Platform I4.0 and ZVEI).

7.4.2 Industrial Internet Reference Architecture (IIRA)

The Industrial Internet Reference Architecture (IIRA) has been published by the Industrial Internet Consortium (IIC) in the document "The Industrial Internet of Things Volume G1: Reference Architecture" (The Industrial Internet of Things Volume G1: Reference Architecture Version 1.9, 2019) and contains architectural concepts, vocabulary, structures, patterns and a methodology for addressing design concerns. The document identifies the fundamental architecture constructs and specifies design issues, stakeholders, viewpoints, models and conditions of applicability defining a framework by adapting architectural approaches from the ISO/IEC/IEEE 42010-2011 Systems and software engineering—Architecture description standard.

This international standard outlines the requirements regarding a system, software, and enterprise level architecture. The ISO/IEC/IEEE 42010 standard recommends identifying the perspectives of the various different stakeholders that can be: system users, operators, owners, vendors, developers, and the technicians who maintain and service the systems. The aim is to describe system properties as seen from their viewpoint. Such properties include the intended use and suitability of the concept in terms of its implementation, the implementation process itself, potential risks, and the maintainability of the system over the entire lifecycle.

Essentially, the IIRA attempts to identify the most important and common architecture concerns. It then provides an architectural template and methodology that engineers can use to examine and resolve design issues. In addition, the template and methodology suggest ways of addressing the top concerns, allowing designers to glean insights by examining architecture patterns, helping Industrial Internet of Things (IIoT) system designers to avoid missing important architecture considerations and this also helps them to identify design gaps of missing important system functions or components.

The core of the IIRA's methodology lies in a set of system conceptualization tools called viewpoints that enable architects and engineers to identify and resolve key design issues. Thus, the IIRA design starts with defining the shapes and forms of an Industrial Internet of Things Architecture by starting with the viewpoints of the stakeholders. These IIRA's viewpoints are arranged in a particular order to

reflect the pattern of interactions that occurs between the four elements, because the decisions from a higher-level viewpoint impose requirements on the viewpoints below it. In this sense, the IIRA is a layer model that takes into consideration four different viewpoints (business, usage, functional, and implementation). It focuses on the capabilities from the perspective of the software and their business processes. Each of the four viewpoints outlined in IIRA can be compared with the respective layers on the vertical axis of RAMI 4.0; RAMI 4.0 supplements the model with the axes 'Lifecycle' (with types and instances) and 'Hierarchical Levels.'

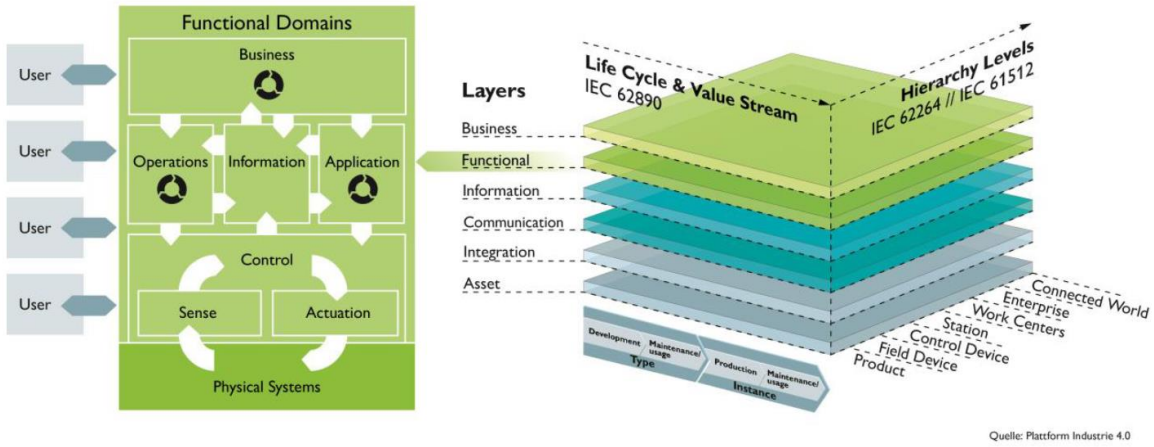


Figure 17. The viewpoints of the IIRA can be represented in the corresponding layers in the RAMI 4.0 model.

The IIoT technologies core implemented in IIRA are applicable to the depth and breadth of every small, medium and large enterprise in manufacturing, mining, transportation, energy, agriculture, healthcare, public infrastructure and virtually every other industry. In addition to IIoT system architects, the plain language of IIRA and its emphasis on the value proposition and enablement of converging Operational Technology (OT) and Information Technology (IT) enables business decision-makers, plant managers, and IT managers to better understand how to drive IIoT system development from a business perspective.

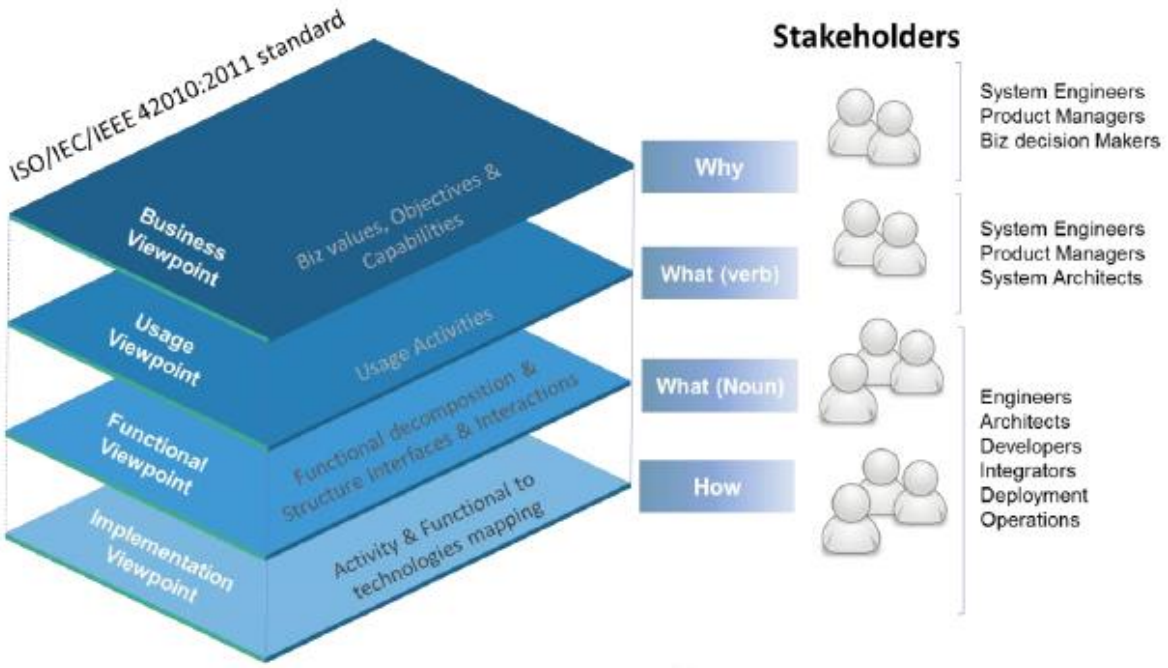


Figure 18. IIRA Architectural Framework

Security Framework (IISF)

Additionally, if the design of the IIoT solution requires considerations within the context of all the viewpoints -crosscutting concerns- as for example security and safety issues, it exists the cross-cutting functions and the system characteristics. The figure below illustrates the relationship between functional domains, cross-cutting functions and system characteristics.

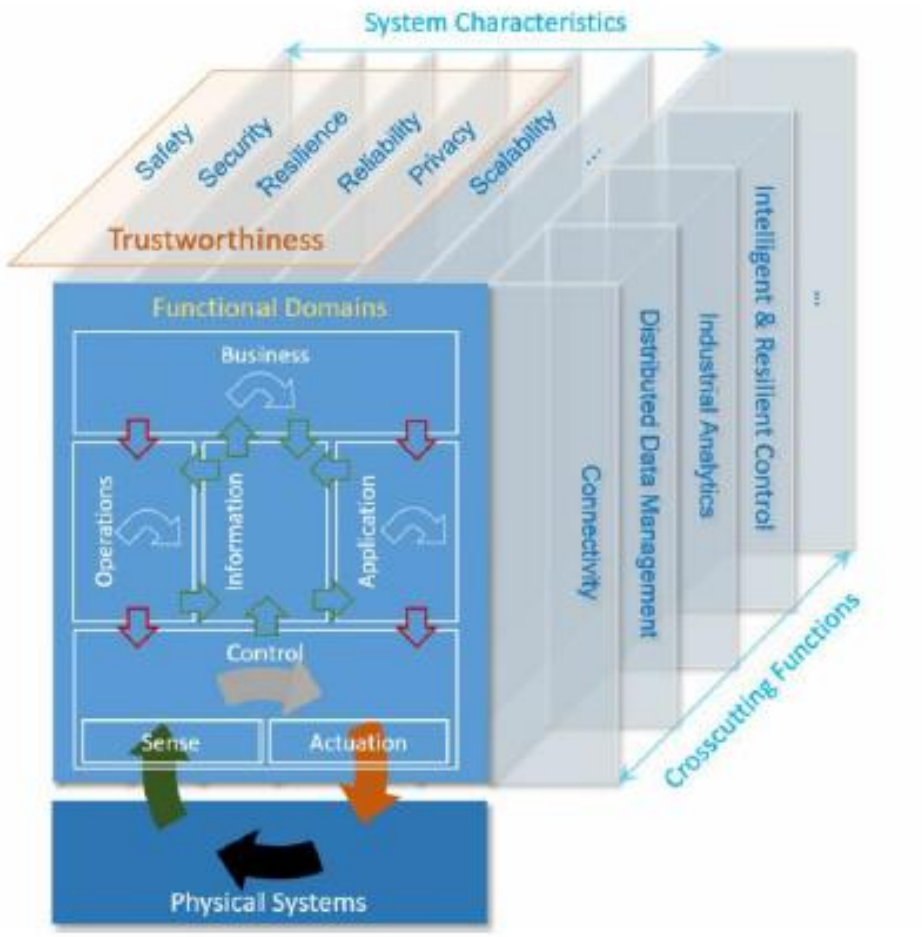


Figure 19. IIRA Functional Domain, crosscutting functions and System Characteristics (source IIC)

IIoT systems are typically systems that interact with the physical world where uncontrolled change can lead to hazardous conditions. This potential risk increases the importance of safety, reliability, privacy and resiliency beyond the levels expected in many traditional IT environments.

The “Industrial Internet of Things Volume G4: Security Framework”, (Industrial Internet of Things Volume G4: Security Framework - IIC:PUB:G4:V 1.0:PB:20160926, 2016) published by the Industrial Internet Consortium (IIC), identifies, explains and positions security-related architectures, designs and technologies, as well as identifies procedures relevant to trustworthy Industrial Internet of Things (IIoT) systems. It describes their security characteristics, technologies and techniques that should be applied, methods for addressing security and how to gain assurance that the appropriate mix of issues have been addressed to meet stakeholders' expectations.

7.4.3 OpenFog

The Internet of Things (IoT) is driving business transformation by connecting everyday objects and devices to one another and to cloud-hosted services.

Current deployment models emphasize mandatory cloud connectivity; however, this is not feasible in many real-world situations. These are two of the primary issues with connecting edge devices to the cloud for all services:

- Connected devices are creating data at an exponentially growing rate, which will drive performance and network congestion challenges at the edge of the current infrastructure.
- Performance, security, bandwidth, reliability, and many other concerns make cloud-only solutions impractical for many use cases.

Unfettered cloud-only architectural approaches cannot sustain the projected data velocity and volume requirements of the IoT. To sustain IoT momentum, the OpenFog Consortium is defining an architecture to address infrastructure and connectivity challenges by emphasizing information processing and intelligence at the logical edge. This approach is called fog computing.

Figure 20. Unfettered Cloud Computing

The fog computing model moves computation from the cloud closer to the edge, and potentially right up to the IoT sensors and actuators. The computational, networking, storage and acceleration elements of this new model are known as fog nodes. These are not completely fixed to the physical edge, but instead should be considered as fluid system of connectivity.

OpenFog architectures offer several unique advantages over other approaches, named SCALE:

- Security: Additional security to ensure safe, trusted transactions
- Cognition: awareness of client-centric objectives to enable autonomy
- Agility: rapid innovation and affordable scaling under a common infrastructure
- Latency: real-time processing and cyber-physical system control
- Efficiency: dynamic pooling of local unused resources from participating end-user devices

The OpenFog Reference Architecture (OpenFog Reference Architecturefor Fog Computing, 2017) describes a generic fog platform that is designed to be applicable to any vertical market or application. This architecture is applicable across many different markets including, but not limited to, transportation, agriculture, smart-cities, smart–buildings, healthcare, hospitality, financial services, and more, providing business value for IoT applications that require real-time decision making, low latency, improved security, and are network-constrained.

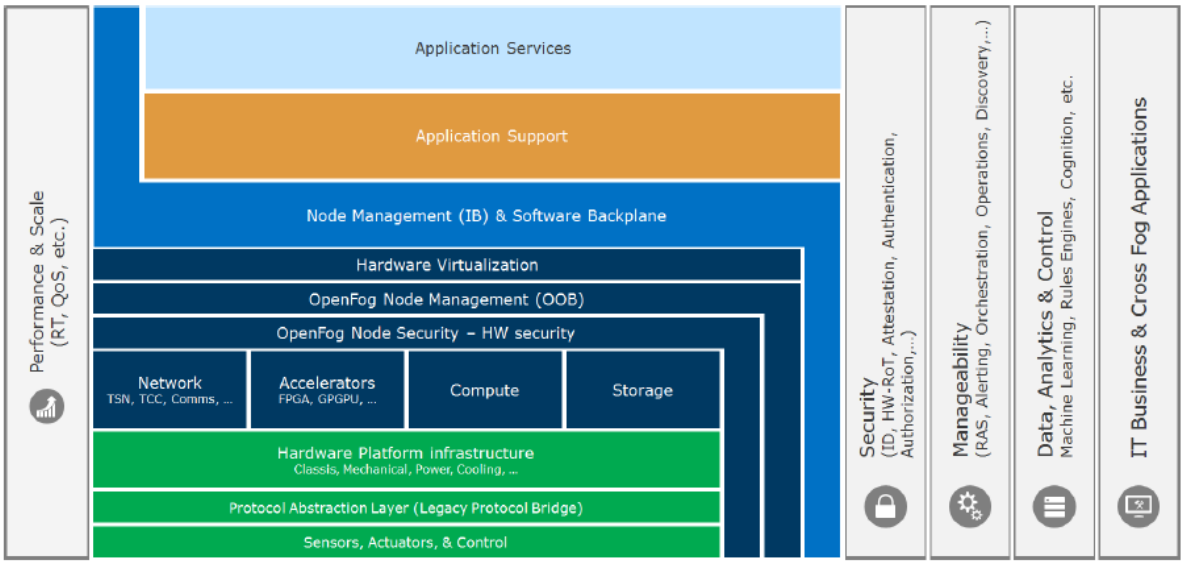


Figure 21. OpenFog Architecture with perspectives

The architecture includes perspectives, shown in grey vertical bars on the sides of the architectural description, which include:

- Performance: Low latency is one of the driving reasons to adopt fog architectures. There are multiple requirements and design considerations across multiple stakeholders to ensure this is satisfied. This includes time critical computing, time sensitive networking, network time protocols, etc. It is a cross cutting concern because it has system and deployment scenario impacts.
- Security: End-to-end security is critical to the success of all fog computing deployment scenarios. If the underlying hardware is secure but the upper layer software has security issues (and vice versa), then the solution is not secure. Data integrity is a special aspect of security for devices that currently lack adequate security. This includes intentional and unintentional corruption.
- Manageability: Managing all aspects of fog deployments, which include RAS, DevOps, etc., is a critical aspect across all layers of a fog computing hierarchy.
- Data Analytics and Control: The ability for fog nodes to be autonomous requires localized data analytics coupled with control. The actuation/control needs to occur at the correct tier or location in the hierarchy as dictated by the given scenario. It is not always at the physical edge, but instead may be at a higher tier.
- IT Business and Cross Fog Applications: In a multi-vendor ecosystem, applications need the ability to migrate and properly operate at any level of a fog deployment's hierarchy. Applications should also have the ability to span all levels of a deployment to maximize their value.

The OpenFog RA description is a composite of perspectives and multiple stakeholder views used to satisfy a given fog computing deployment or scenario. The three views that we have identified include Software, System, and Node.

- Software view: is represented in the top three layers shown in the architecture description, and include Application Services, Application Support, and Node

Management (IB) and Software Backplane.

- System view: is represented in the middle layers shown in the architecture description, which includes Hardware Virtualization down through the Hardware Platform Infrastructure.
- Node view: includes the Protocol Abstraction Layer and Sensors, Actuators, and Control.

7.5 LSP Reference Architectures in Smart Agriculture

7.5.1 DataBio

The DataBio project follows the BDVA Reference Architecture presented earlier in this document. The figure below presents the BDVA architecture together with the number of the component group used for model category. An exhaustive list of all the 91 components that make up the DataBio platform can be found in Table 2 of deliverable D4.1 of the DataBio project. (IoF2020 Deliverable 3.2 "The IoF2020 Use Case Architectures and overview of the related IoT Systems")

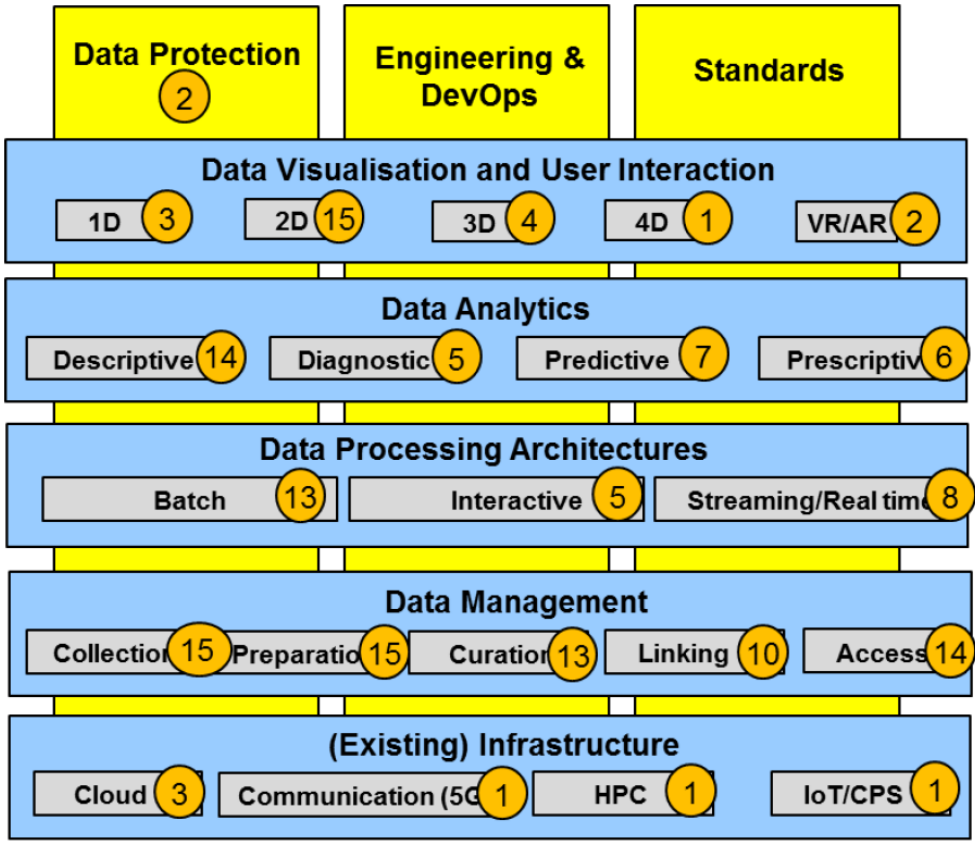


Figure 22. The BDVA architecture of DataBio with reference to the DataBio component numbers

This architecture has several layers each one made up of many possible components. The top layer has the data visualisation and UI tools (e.g., 2D, or 3D visualisation). This sits on top of the data analytics layer that generated data for the UI using techniques such as neural networks; this layer uses components for descriptive analytics, which analyse past (or historical) data to understand trends and evaluate metrics over time, predictive analytics aim to predicts future trends based on past data, and prescriptive analytics which showcase viable solutions to a problem and the impact of considering a solution on future trends. Below that sits the data processing architecture which allow

batch, interactive or real-time processing of data and include the relevant technologies and databases (e.g., Apache). The data management layers are responsible for the collection, preparation and curation of data and they include the agrifood data marketplace. At the bottom of the architecture sits the layer of the actual infrastructure used such as the cloud, 5G, IoT etc., which enables the connection to devices that provide the data used in the other layers.

7.5.2 IOF2020

Building upon the IoT reference model (see standard recommendation ITU-T Y.2060 dated 06/2012) which is presented in the figure below, and its evolution which is the functional view of the IoT-A ARM (presented previously in Section 7.1.1), the IOF2020 use case pilots utilize customized architectures, one for each specific use case.

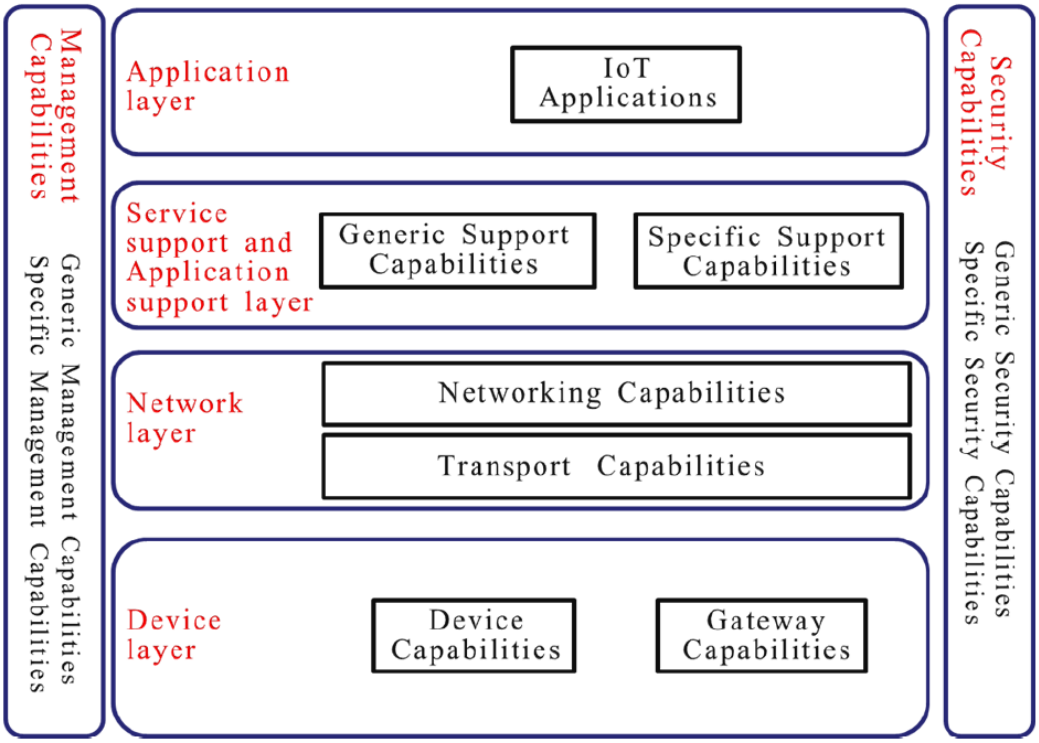


Figure 23. IoT reference model of ITU-T Y.2060

For example, in the three figures that follow, we present respectively the functional views for use cases 1.4, 3.2 and 3.3 of IOF2020 that show the connection with the functional view presented in following figures. More specifically, the application layer sits at the top of the architecture, with the device layer at the bottom. On top of the device layer sits the connectivity layer (see "Communication" layer in the IoT-A Architectural Reference Model) and between this and the application layer there is the service support and application support layer (the components of which correspond loosely to the other layers in the IoT-A Reference Model); for example clearly the IoT service layer corresponds to the equivalent layer in the IoT-A Reference Model. The only exception seems to be the information mgmt. layer (also referred as data layer in other figures) part of which seems to correspond partly to the business process management layer and part to others in the IoT-A Reference Model. Finally, Management and Security (at the sides) apply to all layers.

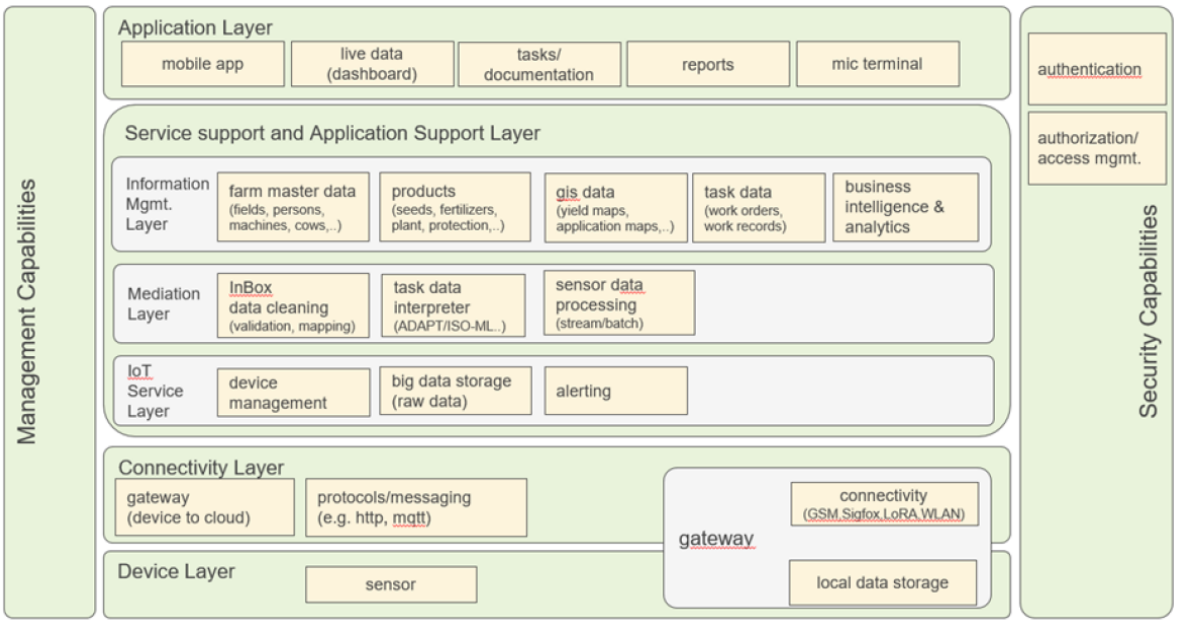


Figure 24. IOF use case 1.4

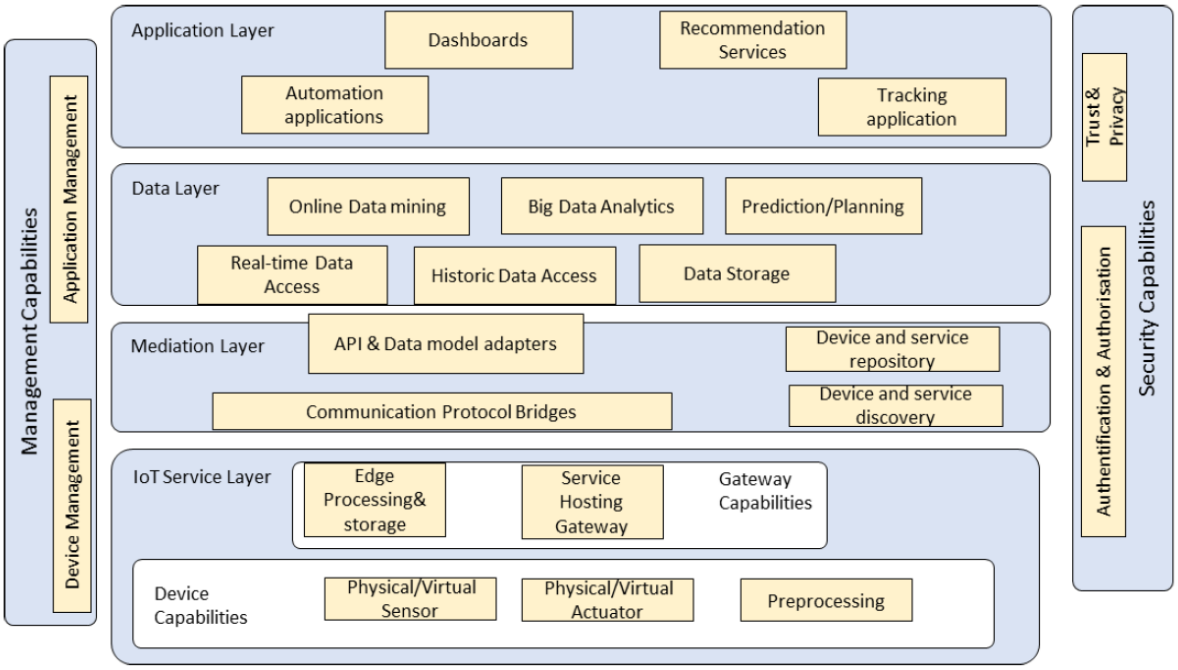


Figure 25. IOF use case 3.2

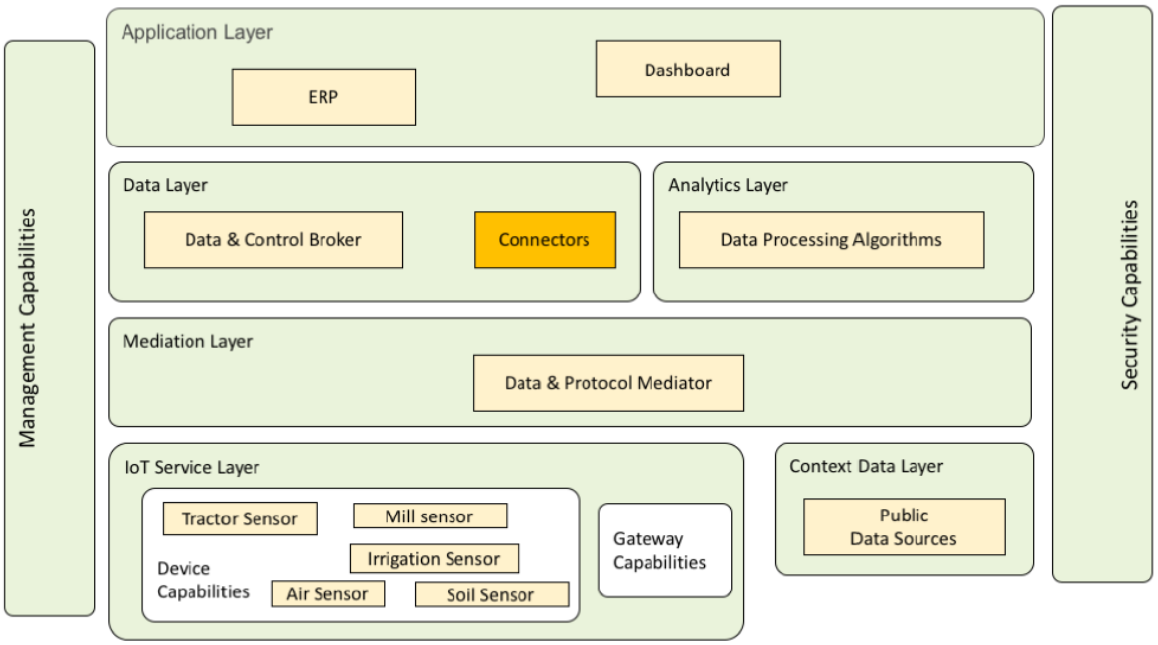


Figure 26. IOF use case 3.3

Looking at these functional views and those for a number of other use cases (see Figure below), we can observe that the same general architecture is used, however it is customized heavily to each specific use case, which should not be desirable: a lot of effort is actually devoted into this customization. Each functional view essentially requires significant customization both in the type of devices that are being connected, the communication protocols and in the way that data is stored and processed in each use case pilot.

What the IOF2020 approach and its architectures lack, their main drawback, is semantic interoperability. More specifically, it would be far more desirable that the architecture should allow different services to be used and input into the same platform on a need basis. To achieve this, the architecture should incorporate common data management tools and common data frameworks for storing, processing and transmitting the knowledge collected through raw input data, or processed data by various services and tools. It should also employ a portfolio of communication protocols as appropriate to receive the input data from various sources (mainly devices but also human user if appropriate). Furthermore, it should provide services that “translate” between data formats (e.g., those used by devices that input data into the platform) in order to maintain semantic interoperability. Finally, ideally the services used should use a common standard so that they can be used and composed together on demand for the different application domains and problems without the need for expensive and time-consuming customization. Some of these issues (e.g., semantic interoperability) are being addressed by the architecture proposed by DEMETER presented in detail in section 11.

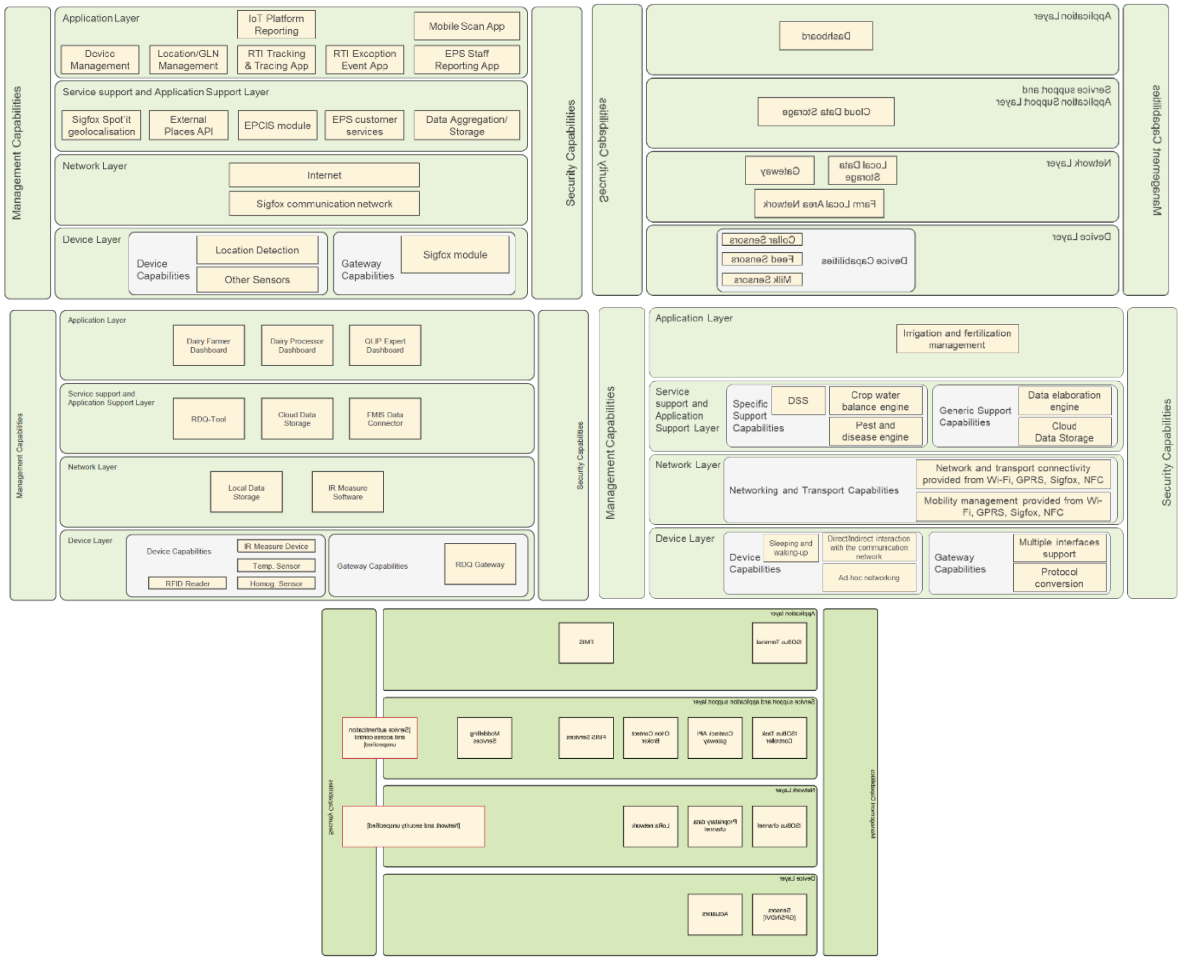


Figure 27. Additional IOF use cases

7.5.3 AFarCloud

AFarCloud (Aggregate FARming in the CLOUD) (AFarCloud D2.2 “Architecture requirements and definition (v1)”,, 2019) is an ECSEL project whose goal is to provide a distributed IoT platform for early adopter farmers and rural professionals willing to use agriculture real-time computer systems to increase efficiency, productivity, animal health and food quality, and also to reduce agricultural labour costs. This platform is integrated with farm management software and supports monitoring and decision-making solutions based on big data and real time data mining techniques.

The AFarCloud platform consists of three main functional components: (i) the *Farm Management* System, (ii) the *Semantic Middleware* and (iii) the *Deployed Hardware* layer. Besides, the AFarCloud platform interconnects with other data sources like 3rd Party data and legacy systems databases. This architecture is presented in the figure below.

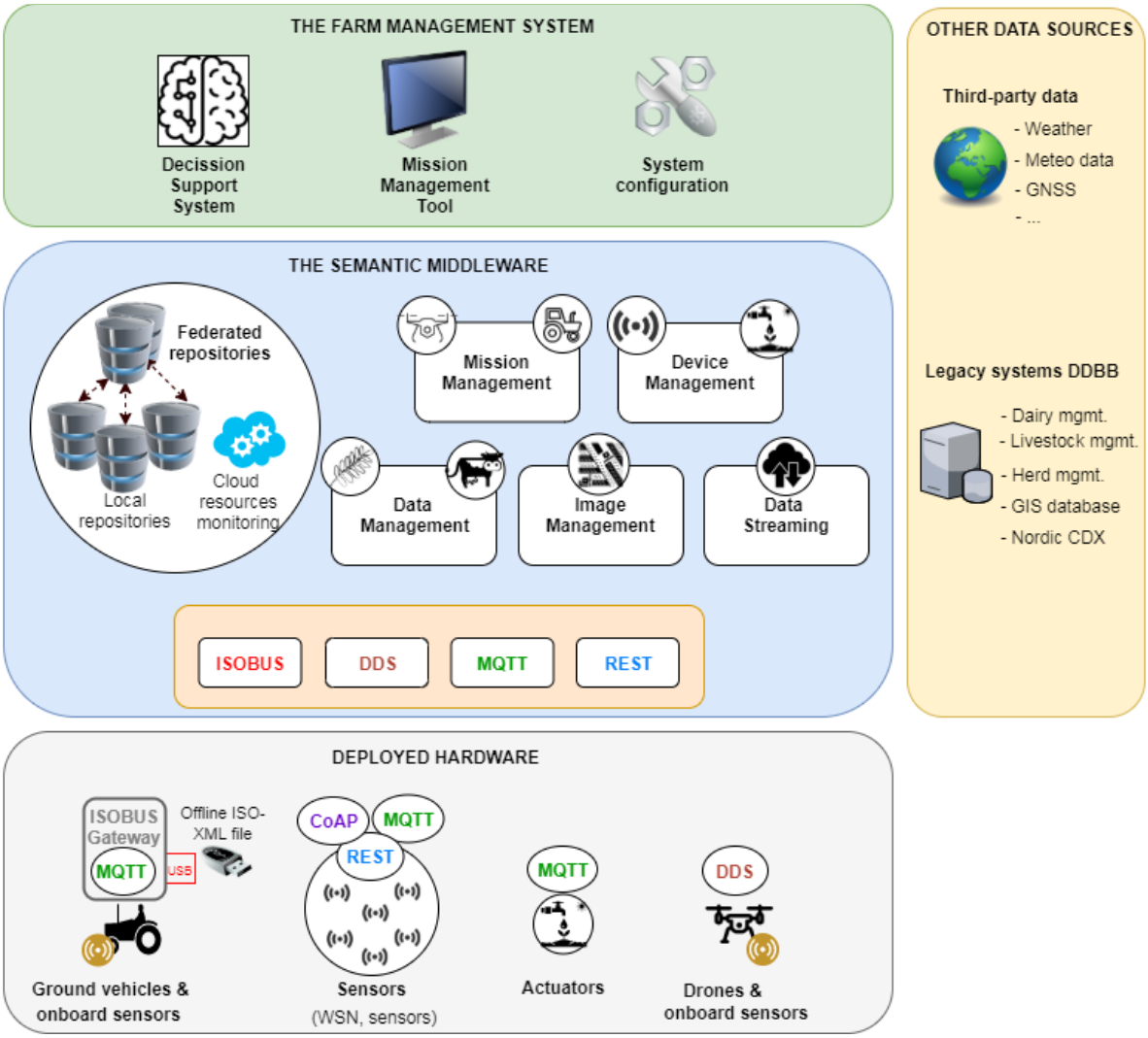


Figure 28. The AFarCloud architecture

The *Farm Management System* offers: a Mission Management Tool (MMT) to plan cooperative missions involving Unmanned Aerial Vehicles (UAV) and ground vehicles ranging from fully autonomous UGVs to legacy systems; a Decision Support System (DSS) to make decisions pre-, during- and post-mission; a system configurator to configure the above-mentioned systems, including their key hardware components (mission relevant sensors and other component important for performing a mission); and, applications for the user to manage and monitor the whole system.

The *Semantic Middleware* offers among others, components for: data storage and retrieval from the Cloud; managing and cataloguing images; registration of IoT devices, animals and vehicles in the farm; data flow management inside the platform; managing, controlling and acquiring data from IoT devices and missions involving ground and aerial vehicles; data processing and knowledge extraction. The Semantic Middleware implements a software layer that hides the underlying complexity of the deployed hardware, so that the Farm Management System can access to that hardware in a unified way. The AFarCloud middleware uses semantic models, specified by the AFarCloud ontology to abstract the heterogeneity of the underlying hardware, and to ensure that all information is stored according to a common information model that guaranties interoperability. The semantic middleware acts as a communication centralizer, disseminating messages between the

Farm Management System and the hardware deployed. The Semantic Middleware offers a set of interfaces to the deployed hardware, like: DDS, to send missions to UAVs and retrieve their feedback in real-time; ISOBUS, to manage communications with ISOBUS compatible ground vehicles; MQTT, to manage communications with standalone and vehicle onboard sensors and actuators; and REST, to retrieve data gathered by legacy systems databases.

The *Deployed Hardware* layer provides means to deploy and integrate the services and data related to unmanned aerial vehicles, semi-autonomous ground vehicles, actuators, sensors and other IoT devices.

Something that is still missing (since it was not in the scope of this project) from this architecture is the support for the interoperability between several farm platforms (FMS) and their offering services and not just repositories.

7.6 OPEN DEI CSA: Cross-domain Digital Transformation Reference Architecture

OPEN DEI is the CSA funded under the "DT-ICT-13-2019 - Digital Platforms/Pilots Horizontal Activities" topic, bullet "a) Support pilot activities and knowledge transfer across different sectors". This CSA started at June 2019 and it is an essential pillar of the implementation of EU digitisation policies by addressing in particular the "support" to the Large-Scale Pilots (LSPs) and platform projects financed by the European Commission under the Digitising European Industries (DEI) Focus Area.

OPEN DEI aims to set up a win-win collaboration with the supported Innovation Actions (all of them dealing with Digital Platforms for Digital Transformation), as well as with other ongoing endeavours in the four sectors tackled by the CSA (namely Manufacturing, Agrifood, Health and Energy). To this end, several actions are planned and undergoing to address the main needs of the projects under its umbrella.

In the agrifood domain, DEMETER is one of the reference LSPs projects, also thanks to the common beneficiaries among the two actions (i.e., ENGINEERING and ATOS). Other relevant projects targeted in such a sector are Atlas, IoF2020, SmartiAgriHubs, agROBOfood.

In terms of Platform Building, OPEN DEI aims at studying and harmonizing Reference Architectures (RA) in EU and worldwide in the different domains covered by Digital Transformation (DT) focus area. In particular, a Data-driven approach will be followed, as the Industrial Internet Consortium Reference Architecture and its layered data-buses pattern instantiated in the different layers of the Platform Industrie 4.0 RAMI. Moreover, in order to reduce the DT entry barrier for SME and to mitigate the risk of vendor lock-in, OPEN DEI is stimulating and promoting the adoption of Open Source reference implementations, based on Open Standards, such as FIWARE, OpenIOT and APACHE. Finally, OPEN DEI will support the flourishing of EU Industrial Data Platforms and Industrial Data Spaces in four different domains (Manufacturing, Agriculture, Energy and Healthcare), by supporting the identification, modelling, execution and evaluation of B2B Data Exchange / Data Sharing business processes.

At the time of writing this deliverable, the OPEN DEI working group dealing with Digital Platform and Reference Architecture in the agrifood domain is not fully operational, therefore this section is intended to provide an high level view of the potential collaboration with the CSA, assuring the work

done and the lesson learned in DEMETER will support the knowledge sharing and exploitation activities within the OPEN DEI working group.

8 DEMETER Architectural Framework and alignment process

This section creates connections among each of the Reference Architectures presented and the DEMETER project, presenting a short summary of the reasons why these frameworks have been taken into account to build the DEMETER Reference Architecture. To this end we present in the following table the key components or design features of each one of these architectures that we take into consideration when designing the Demeter RA (which will be presented in later sections of this deliverable).

Table 1. Examples of Reference Architectures analyzed as input of the DEMETER RA

Organisation	Description
IoT-A	The IoT-A (Internet of Things - Architecture) project has proposed an IoT-A Architectural Reference Model (ARM) together with the definition of an initial set of key building blocks. The IoT-A ARM is a set of best practices, guidelines, and a starting point to generate specific IoT architectures.
AIOTI	The HLA primarily introduces a domain model, which describes entities in the IoT domain and the relationships between them, and a functional model, which describes functions and interfaces (interactions) within the IoT domain. The AIOTI functional model describes functions and interfaces between functions of the IoT system. Functions do not mandate any specific implementation or deployment.
BDVA	The European Big Data Value Association (BDVA) has constructed this reference framework, which consists of the universal big data system. The following challenges should be met in order to realize BDVA: a) new business models, b) confidence in AI and its outputs, c) development of AI and Big Data ecosystem, d) AI technologies fusion.
NIST	NIST sets data in a broad level and service use flows between the dependencies of the framework. It consists of the following components: a) System Orchestrator, b) Data Provider, c) Big Data Application Provider, d) Big Data Framework Provider, e) Data Consumer All these components are covered by 1) Management and 2) Security and Privacy fabrics.
FIWARE	FIWARE is open source and uses IoT, Big Data and Cloud architectures. It distributes the data and vertical silos in many IoT systems, though the use of horizontal layers, in a way to control large-scale context information. It interacts with a Context Broker, in order to construct a RESTful interface and an information model. FIWARE components are concerned with 1) Interface with IoT, 3rd Party systems and robots, 2) Context Data/API management and 3) Processing of context information.
IDSA	IDSA focuses on who owns the information, aiming to enable transparent exchanges between data sources and consumers. In order to accomplish that, it offers a reference distributed architecture. In order to support reliability in trusted transfer of information the following are met: 1) Secure communication, 2)

	Identity Management, 3) Trust Management 4) Trusted Platform, 5) Data Access control, 6) Data Usage Control.
RAMI4.0	Industrie 4.0 aims at connecting all stakeholders involved in the business processes of the manufacturing and process industry so that all participants involved share a common perspective and develop a common understanding. The Reference Architectural Model Industrie 4.0 (RAMI 4.0) is a three-dimensional map in support of the most important aspects of Industrie 4.0.
IIRA	The Industrial Internet Consortium (IIC) has developed the Industrial Internet Reference Architecture (IIRA) in order to address the need for a common architecture framework to develop interoperable IIoT systems for diverse applications across a broad spectrum of industrial verticals in the public and private sectors.
OpenFog	OpenFog aims to deal with the exponentially growing data generated from IoT devices that can make cloud-only solutions impractical for many use cases. It defines an architecture that addresses infrastructure and connectivity challenges by emphasizing information processing and intelligence at the logical edge. This is called fog computing and moves computation from the cloud closer to the edge, and potentially right up to the IoT sensors and actuators.
DataBio	DataBio has the following layers, starting from top-to-down approach: 1) Data visualisation and User interaction, 2) Data Analytics, that uses neural networks, 3) Data Processing Architectures, in order to process interactive, batch or real-time data, 4) Data Management layer that collects, prepares and curates data, 5) Infrastructure, that uses technology solutions in order to connect devices.
IOF2020	IOF2020 builds upon the IoT reference model and its evolution the functional view of the IoT-A ARM; this is a layered architecture where the devices sit at the bottom, with communication facilities on top of them linking them to the service and app support layer, while at the very top sit the applications. Security and data management are present as vertical slices that cover all the layers. IOF2020 customizes this architecture for each of its use case pilots, therefore each one utilizes a slightly different, customized architecture, depending on the needs of the specific use case; without having generic interoperability facilities (without customization).
AFarCloud	AFarCloud provides a distributed IoT platform for farmers and professionals so that they can increase efficiency, animal health, productivity and food quality, and reduce the costs on labouring. The basic elements of the AFarCloud are the following: 1) Farm Management System, 2) Semantic Middleware, 3) Deployed Hardware. It can also connect to other data sources such as 3rd Party data or legacy systems DB.
OpenDEI	OpenDEI aims to support collaboration and knowledge sharing among Digital Platform providers acting in several industries (at least in the four domains tacked by the CSA, such as Manufacturing, Agrifood, Health and Energy). To this end,

	several technologies enabling actual Digital Transformation processes will be analysed. Regarding the definition and adoption of Reference Architecture for DT processes, OPEN DEI will provide a knowledge framework to exploit the results from closing or advanced Large Scale Pilots (LSPs) into new comers such as DEMETER. Therefore, the DEMETER Reference Architecture will be mapped toward this meta-architecture in order to ease the cross-domain knowledge exchange, acting at the same time as an input to OpenDEI and vice versa thanks to the active participation in the working groups defined by the CSA and related to the Digital Platform pillar (as per the DEI Communication back to April 2016).

-

Some commonalities have been identified across the presented Reference Architectures. The rest of the document introduces the initial release for a consolidated DEMETER Reference Architecture, on which all the architectures of the 20 project pilots can be mapped. Beyond this initial consolidation, the applicability of the proposed DEMETER Reference Architecture extends beyond the project pilots as it is quite generic and can be easily mapped to most of the reviewed approaches.

9 DEMETER Technical Requirements

In this section we present a summary of our analysis cataloguing the requirements for the various DEMETER software module blocks. This summary is distilled from the detailed work done on cataloguing the requirements by the corresponding work packages (WP2, WP3 and WP4). We leave it to the corresponding deliverables of these WPs to present in detail the full list of the specific requirements extracted:

- D2.1 DEMETER data models and semantic interoperability mechanisms (May 2020)
- D2.2 DEMETER data and knowledge extraction tools (June 2020)
- D3.2 DEMETER technology integration tools (July 2020)
- D4.2 Decision Enablers, Advisory Support Tools and DEMETER Stakeholder Open Collaboration Space (July 2020)

The abbreviations DK, IT and BDS used to identify the requirements (in the next subsections) stand for "Data & Knowledge", "Technology Integration" and "Benchmarking & Decision Support" respectively.

9.1 WP2 technical requirements overview

This subsection elaborates upon the technical requirement classes specified by WP2 addressing the data and knowledge related functionality to be delivered by the DEMETER ecosystem. The input considered to extract these requirement classes includes the following: information collected by the pilot stakeholders, related state of the art review, DEMETER targeted objectives and innovations, WP2 goals and related expertise and assets in the DEMETER consortium. The requirement classes include mainly functional but also some non-functional and some pilot-specific requirements.

DK1. Data Modelling

The technical requirements under the DK1 class are related to the establishment of a common data model (AIM) for DEMETER able to represent the wealth of data types in the agrifood value chain in general and in the DEMETER pilots in particular. These requirements represent the need for full scale semantic interoperability across standardised or dominant approaches for semantic representation of data in the agriculture sector and in general they highlight the fact that a common data model needs to be in place that may be used and reused, extended, refined, understood and governed.
These requirements will be presented in detail in D2.1 (Common data models and semantic interoperability support - Release 1), which is expected in April 2020.

DK2. Semantic mapping of AIM to dominant/standardised agrifood solutions

The technical requirements under the DK2 class complement the data model of DK1 and in particular the need for full scale semantic interoperability. To this end, they deal with the development of semantic wrappers and translators that align existing ontologies to the DEMETER AIM; they identify the necessary ontologies and data that should be included in this process and also the semantic mappings from widely used ontologies/vocabularies to AIM, enabling in this way the semantic integration of this data into the DEMETER system. They also select relevant standards and identify

the key terms that need to be aligned between these ontologies and AIM, using the appropriate mapping constructs/axioms.
These requirements will be presented in detail in D2.1 (Common data models and semantic interoperability support - Release 1), which is expected in April 2020.

DK3. Data integration

The technical requirements under the DK3 class deal with mechanisms that integrate different types of incoming data from different sources facilitating interoperability between communicating entities in the platform by using well-defined APIs and protocols. To this end, they state the requirements for appropriate storage systems enabling access to and querying of linked (integrated) datasets; they also 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, and support of suitable tools for the semantic annotation of datasets. They also specify a common query language interface (through an appropriate API).
These requirements will be presented in detail in D2.2 (DEMETER Data and Knowledge extraction tools – Release 1), which is expected in May 2020.

DK4. Data Management

The technical requirements under the DK4 class complement those of DK3. They specify requirements for a set of good practices, architectural techniques and tools able to manage the complete data lifecycle management process and to ensure the accessibility and reliability of the data used by DEMETER applications. They require an infrastructure capable of storing and retrieving data and aggregating (also synchronizing if necessary) data from heterogeneous sources as well as ensuring that data are updated regularly and kept fresh and that different databases used are kept consistent and up to date. They also store the relevant metadata in order to facilitate this process. The end goal is to keep the continuous and reliable operation of the data storage system during its operation.
These requirements will be presented in detail in D2.2 (DEMETER Data and Knowledge extraction tools – Release 1), which is expected in May 2020.

DK.5 Data Quality & Fusion

The corresponding requirements deal with issues concerning the integration of data in numerous situations, like those coming from various data sources -potentially in multiple file formats as well-, data in forms different than expected or even plain raw data, data incoming in an asynchronous manner as well as data from legacy systems that need to be used by new systems. Some possible functionalities that need to be supported are normalization, standardization and merging of data. Additionally, this class of requirements tackles data quality assessment challenges e.g., choosing the most accurate metrics for each data type, deciding the most appropriate between several available data sources, determining the action when there are non-matching, missing and/or corrupted data and ensuring the retention of this quality through all the stages of the data analysis.
These requirements will be presented in detail in D2.2 (DEMETER Data and Knowledge extraction tools – Release 1), which is expected in May 2020.

DK.6 Data Analytics & Machine Learning

Under this class of requirements lies the creation and deployment of tools for the analytical and scientific processing of the data acquired or created within DEMETER for the extraction of patterns and outputs that facilitate decision making in the agricultural domain. This includes data analytics made over heterogeneous data sources and potentially in numerous data forms, data streams or even concrete datasets that should be explored and analyzed on the fly, handling of corrupted or incoherent data. Decision support systems should be fed input from data analytics covering a wide range of agri-data (e.g., field and crop data, weather data, soil and water data) acquired with varying techniques from sensor networks, satellite and imagery data and so on. The processing of these should also follow legal and ethical restrictions. This class of requirement also details the specific analytics needed for the decision support of the various DEMETER pilots specifically.
These requirements will be presented in detail in D2.2 (DEMETER Data and Knowledge extraction tools – Release 1), which is expected in May 2020.

DK.7 Data Security & Privacy

These requirements handle all issues related to the security and privacy of the data network. They necessitate the deployment of lightweight protocols that ensure encryption within every layer of the architecture and the protection of the network and communications overall, as well as individual components, from intrusion attempts by monitoring and detecting such intrusions and triggering appropriate alarms and countermeasures. Authentication should be required during all critical data accesses. Data ownership, accessibility and traceability should be well-defined and strict. Finally, these deal with the privacy considerations for the data ensuring compliance with GDPR regulations.
These requirements will be presented in detail in D2.2 (DEMETER Data and Knowledge extraction tools – Release 1), which is expected in May 2020.

9.2 WP3 technical requirements overview

This subsection enlists the various requirement classes of WP3. These classes are mainly a categorization of the requirements that will be identified and upon which the design and implementation of DEMETER's reference implementation will be based. The requirement classes include both functional and non-functional requirements. There are few requirements that do not fall under any of these classes, being more general, and these will be detailed in general class of requirements. The process of collecting requirements is mostly completed, and the details of all the requirements will be presented in future WP3 deliverables, e.g., in D3.2 (DEMETER Technology Integration Tools – Release 1), which is expected in June 2020.

TI1. Technical and Syntactic Interoperability of pilot technologies/platform

DEMETER's target is to bridge different technologies, tools, platforms, and frameworks that already exist and are being utilized by companies and organizations across the agricultural domain. Since the project's goal is to provide this interoperability between several services, data, and devices, this requirement class focuses on identifying the important technologies, tools, platforms, and frameworks that need to interoperate. The reference that will be used for this identification process stems from the various pilots and use cases that exist within the context of WP5 (Deliverable D5.1

has already been submitted). Technical interoperability will be achieved by defining the appropriate way(s) to combine different technologies that are not currently directly interoperable between them, while syntactic interoperability will be achieved by defining a common syntactic structure to be used by the involved parties.

TI2. Environment for service discovery and provisioning

One of DEMETER's services will be to provide an environment within which DEMETER-enabled resources (services, data, devices) can be discovered and provisioned. This class aims to gather requirements that will identify the needs of such an environment and the ways with which discovery and provisioning will be achieved. These requirements should answer the question "how a resource can be discovered and provisioned in the context of DEMETER".

TI3. Networking and Communication

Networking and communication are an integral part of interoperability, hence, the need for collecting requirements that specify several aspects of this feature of the DEMETER RA and subsequent system implementation. Identifying the ways that several resources that will be taken into consideration and of the various components and entities used for the reference implementation can be interconnected is the focus of this requirement class.

TI4. Security

This requirement class deals with the security aspects that need to be considered in the design of DEMETER's reference implementation. It can be regarded as an extension to the Networking and Communication class (TI3). The various resources (such as services, data and devices) that will be involved in DEMETER already employ security mechanisms and DEMETER needs to assure that the existing security level of these resources remains or is increased if necessary.

TI5. Device/resource Management (including databases)

DEMETER aims to provide interoperability between existing resources. Resources might include services, data sources, devices, and databases, that are being offered by various organizations. Thus, the need for specifying whether these resources will be managed, how, and at which level (to which extend) are addresses by this set of requirements.

TI6. Runtime Environment, Deployment Management & Orchestration

DEMETER will provide the (technical) means to existing resources (e.g., components, devices or systems) to interoperate and subsequently be combined into final solutions. At the same time, it aims not to be a platform in the middle but to provide the capability for the resources to directly interconnect. This raises the need for defining and specifying the environments in which the setup of any deployment will take place, including its management and orchestration.

TI7. Service / application life-cycle management

There will be several services and applications that will be developed within DEMETER. Therefore, there is a need to specify the process that needs to be followed in order to successfully design, develop, test, deploy, and support all these. The requirements for these processes are the focus of the TI7 class.

TI8. APIs and Application development support

DEMETER will include several components that will provide a number of services to users as well as to external developers. External developers will also be able to develop their own Enablers (as described in later sections of this deliverable, most prominently in sections 10 and 11 which follow) in order to make their resources DEMETER-enabled. To facilitate this process, DEMETER will provide tools such as an SDK. The requirements for the available APIs available in these tools as well as any supporting material for application development are the focus of this requirement class.

TI9. Enabler registration, discovery, provision, management, composition, accounting, billing

As described in several sections and depicted in various figures in this document, DEMETER will include a component, named the DEMETER Enabler Hub, that will include several services related to the Enablers that will facilitate the process of developing/adapting DEMETER compliant to external resources. All the requirements that have been identified that are directly related to this component and, in particular, to how users and entities (e.g., services and components) can register and be discovered and consumed through the hub, are the focus of this requirement class.

TI10. Stakeholder account management

There are several stakeholders that will participate in DEMETER. Some of those stakeholders have already been identified at the beginning of the project, some others have been added later. Different stakeholders will have different roles and, hence, needs in DEMETER and this requirement class will deal with the definition of these needs.

TI11. Monitoring, Awareness, Feedback

An overall need of the project is to have the ability to monitor its usage by the stakeholders, to interact with them, and to receive their feedback in order to improve the services provided to them both as a whole but also to each individual stakeholder separately. This requirement class will include the requirements that will drive the design and implementation of DEMETER's components towards monitoring the stakeholder experience and receiving and processing their feedback.

9.3 WP4 technical requirements overview

This subsection elaborates upon the technical requirement classes defined by WP4 for the decision support and benchmarking functions in the DEMETER platform. These requirement classes are extracted based on stakeholder interactions in the DEMETER project, taking into account the

challenges, existing infrastructure and advanced decision support needs of DEMETER pilot partners and also the technical expertise and innovation support available through the technical partners. Further details on these will be given in D4.1 (Decision Support, Benchmarking and Performance Indicator Monitoring Tools – Release 1), which is expected in April 2020.

BDS1. AI-based Decision Making

The technical requirements for AI-based Decision Making are related to the delivery of innovative components/services that are capable of applying advance AI techniques as part of their data analytic functions to address the decision support needs of DEMETER pilots. This represents the need to process heterogeneous datasets (both at rest and in motion) to extract meaningful insight and leveraging AI-based reasoning and recommendation system techniques to choose the most appropriate course of action. Some examples of the data AI-based decision support components/services are: corrective, predictive & optimizing measures tailored to each pilot/domain, farm work organisation, efficient management of farm processes and machines, food travel assessment, instruction and advice for resource consumption, location assessment, and audio-video correlation analysis.

BDS2. Benchmarking

With the decision support functionalities being offered in the DEMETER platform, there is a technical requirement for a benchmarking system that can be used at farm level to evaluate the productivity and the sustainability of the decision support provided by DEMETER and to test and evaluate the efficacy of the developed digital solution in reducing costs, improving production and supporting the long-term sustainability. The benchmarking system needs to provide a set of flexible rules and a shared framework to evaluate the performance of Decision Support components/services across multiple indicators and across multiple use-cases. Some of the key indicators for benchmarking are: agronomic, economic/financial and environmental performance, as pertaining to environmental sustainability and resource consumption, as well as the yield per field or area and crop phenology.

BDS3. Adaptive User Interfacing

The technical requirements for an Adaptive User Interfacing relate to the need to visualise the outcomes of the decision support services in DEMETER. This will help the usability aspects of the DEMETER platform and enable users (with little or no technical background) to better understand the technical information provided through the DEMETER platform. The technical requirements require the User eXperience (UX) solution to be developed as a self-service dashboard framework (catering web and mobile devices) that is capable of integrating a number of service-front-ends, transformation operators and visualisation widgets that users can tune to suit their own needs. Some of the DEMETER services that will be able to take advantage of the visualisation dashboard include the following: benchmarking (current vs predicted), UX evaluation, resource consumption (feed, fertiliser, etc.), visualizations for other tools (e.g. analytics, benchmarking, enablers) as well as the ability to customize the visualization.

BDS4. Multimedia content analytics

The technical requirements for the Multimedia Content Analytics relate to the need of analysing audio and video (A/V) data and identifying specific events taking place at pilot sites. The multimedia data (i.e. real-time streams or historic data coming from DEMETER pilot partner sites) represent the diversity of data types available in the digital agriculture sector in general and in the DEMETER

platform in particular. Robust multimedia analysis functionality that can reliably identify the occurrence of key events (or events of interest) in the physical world is needed. The identification of key events should be coupled with thorough analysis to help relate the identified event with certain conditions and performance/triggering of retrospective action(s).

BDS5. Performance Monitoring and alerting

The technical requirements for performance monitoring and alerting module relate to the need for monitoring physical activities (at DEMETER pilot sites) and generating automatic alerts when certain events occur or when certain conditions are met. The performance monitoring and alerting module will make use of advance stream processing techniques to analyse data in real-time and generate relevant alerts (e.g. triggering of a service, setting off a sensor or sending a text message) when certain (pre-defined) events are detected. Some of the potential applications of performance monitoring and alerting module in DEMETER pilots are as follows: resource consumption monitoring (water, fertilizer, animal feed, fuel, engine emissions), milk production monitoring (production quality and composition), environmental condition assessment, silo condition detection, or power loss monitoring.

BDS6. Stakeholder Open Collaboration Space

The technical requirement for the Stakeholder Open Collaboration Space relate to the need for providing domain/pilot agnostic generic functionalities that can support the integration and collaboration activities in the DEMETER platform. The agri-food domain is characterised by a variety of stakeholders, these stakeholders require a secure mechanism in which information can be exchanged. The Stakeholder Open Collaboration Space addresses such a need to provide a secure environment and technology solution to support the collaborations among DEMETER users.

10 Main Concepts and Terminology

DEMETER is built around the following main concepts that enable the delivery of the project's objectives and targeted outcomes:

- The **DEMETER Stakeholders Open Collaboration Space (SOCS)**, the focus of which is on **resolving the needs of the farmers** using a structured process that converts either an individual need or the most relevant / shared need from a set of previously identified needs to a **challenge**. A challenge is then resolved through a unique **co-creation process**, in which farmers, service advisors and providers can select together the most appropriate set of tools, devices, components, data sources, etc., taking into account the existing ones already deployed at the farms as well as the farmer-defined improvement goals. The SOCS also includes a wide range of features that, together, deliver the knowledge sharing and improvement process, structuring the **human-in-the-loop dimension of DEMETER**. The SOCS is strongly inspired by the EIP Agri Social Spaces and Operational Groups, operating as a set of defined activities for multiple actors implemented through physical meetings, workshops, hackathons, etc., and supported by a dedicated online platform. The DEMETER multi-actor approach is addressed by WP7 and all related aspects are elaborated upon in detail in WP7 deliverables, e.g., D7.1.
- The **DEMETER Agricultural Interoperability Space (AIS)**, which focuses on **delivering** a full set of interoperability mechanisms to actually **develop**, **validate** and then **deploy** the solution. DEMETER does not define completely new interoperability mechanisms but instead uses (and extends) a wide range of pre-existing mechanisms at sensor, data and service levels.
- The **DEMETER Enabler HUB (DEH)**, which centralises the full description of all the components, devices, services, data sources, platforms, etc. that are accessible for exploitation and ultimately for deployment. The DEH provides, on the one side, the harmonised description that enables each component to be used in the co-creation mechanism, and on the other side its uptake in different deployments through the full set of DEMETER enabled interoperability mechanisms. The DEMETER Enabler HUB includes the **mechanisms** that ensure interoperability not only with standardized solutions, but also with dominant solutions introduced by other initiatives, such as IOF2020, ADAPT, DATABIO, or SmartAgriHubs.
- The **DEMETER Dashboard** is the sole entry point to the DEMETER ecosystem for all DEMETER Stakeholders, enabling them to access SOCS and AIS. The Dashboard also offers them user friendly interfaces to access, understand and control data related to their personal accounts, to perform basic administration tasks over their DEMETER accounts, get an overview about the usage of their data (e.g., field data or even perhaps some personal data) by external stakeholders, and to perform other related tasks.

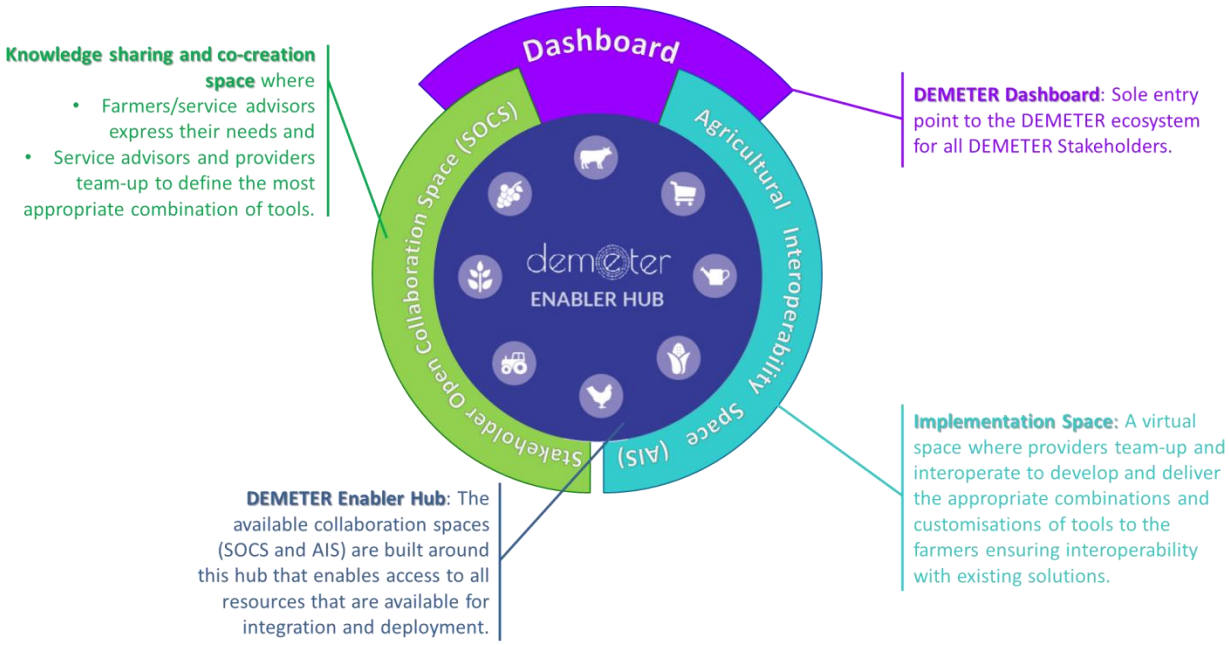


Figure 29. Overview of DEMETER showing the main concepts

A high-level overview of these concepts is presented in the figure above. The key benefits of this are that it connects a human focused interaction space on the left with the actual digital implementation space on the right. This ensures the fact that DEMETER remains fully human centric and human driven – delivering digital enablers that are fully aligned to the needs expressed by the farmers and based on the knowledge and wisdom captured through structured mechanisms.

As already mentioned, all stakeholders will access DEMETER through the dashboard which connects them to the DEMETER facilities. Depending on whether a user connects through AIS or SOCS and depending on the type of usage that the user aims for (e.g., collaboration with other stakeholders, indication of own needs, resource consumption, application development based on DEMETER resources/enablers, etc.), the dashboard will serve a different application and provide different workspaces. To this end, the dashboard will have 2 or 3 different views suitable for the usage intended. The view presented will of course depend on the user and the targeted space.

As already discussed, the *DEMETER Enabler Hub* acts as a point of reference for the interested developers and stakeholders in order to register their offered capabilities and resources and act as ***DEMETER Providers***. These offerings are semantically described and are escorted by meta-data, which include (among others) the security and data usage policies applicable, or Quality of Service metrics. ***DEMETER Consumers*** can browse the *DEMETER Enabler Hub* to discover suitable capabilities and resources matching their requirements and specified criteria. The Hub verifies the identities of the consumers and the providers and provides the support necessary for the establishment of a direct secure communication channel among them. The Enablers made available via the *DEMETER Enabler Hub* are either services developed by the project or resources offered by external stakeholders, i.e., by third party service providers, or by platform providers.

There are some more concepts that need to be introduced herewith before we proceed with the description of the DEMETER Reference Architecture design. Any platform, thing, service or application that makes itself available via the DEMETER Enabler Hub, or consumes resources available in the Hub, or both, is represented by a **DEMETER-enhanced Entity** (figure below). More

specifically, an application can consume DEMETER resources (thus acting as and implementing the DEMETER Consumer), while a service and a platform can both consume or provide resources (implementing the DEMETER Consumer and/or Provider) and finally a thing (i.e., a physical device/asset, such as a sensor or an actuator) can only be made available for consumption (implementing the DEMETER Provider). In order to allow for full scale secure interoperability and communications, there are a few specific DEMETER enablers that are mandatory and need to be available at each DEMETER-enhanced Entity. These are the **DEMETER Core Enablers** that are encapsulated, along with the DEMETER Provider and Consumer in the **DEMETER Enhancing Service**. The Entities are communicating with the DEMETER Enabler Hub via the **DEMETER Hub API**. All DEMETER-enabled platforms and individual things/resources will be registered in the **DEMETER Registry**, along with access rights/policies.

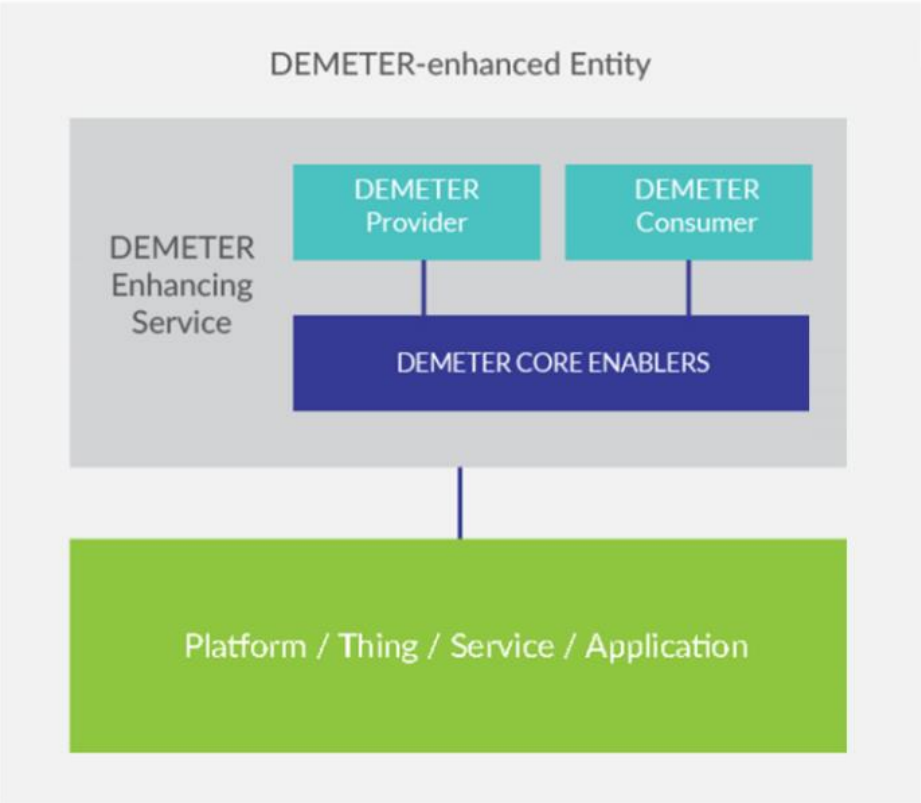


Figure 30. Necessary components of a DEMETER-enhanced entity

The various instantiations of the DEMETER-enhanced entities are presented in detail in Section 11.1 elaborating on the high-level view of the DEMETER Reference Architecture. The rest of Section 11 then presents the other views (functional, data, process, etc.) of the RA.

11 DEMETER Reference Architecture

Having already presented the main concepts of the DEMETER architecture in the previous section, and following the guidelines of Section 5 for the design, we present here the DEMETER Reference Architecture through a number of viewpoints. Firstly, we give a high-level view of the whole architecture including an instantiation example and also describe various instantiations of the DEMETER Entities (in subsection 11.1). Then the functional viewpoint is presented (in subsection 11.2) where the DEMETER systems' main components are described; these include a description of the DEMETER Enabler Hub (DEH) as well as the core (mandatory) and the advanced (optional) enablers offered for the creation of DEMETER entities using the tools offered by DEMETER. Subsequently, in subsection 11.3 the process view of the RA is presented and it is described how the components communicate to accomplish the basic functionalities; these are detailed by also providing activity and sequence diagrams for the registration and then the discovery and usage of DEMETER Enablers. Subsequently, the data viewpoint is elaborated upon in subsection 11.4, which highlights the data flows and which components are needed to manage the main data processes, such as storage architecture, data retrieval, processing, storage and security management of the date exchanged by DEMETER components and enablers. Next, the deployment viewpoint of the RA is introduced in subsection 11.5. This deals with the runtime operations and presents the topology of software components on the physical layer as well as the connections of these components to each other when applications are deployed. Finally, in subsection 11.6, the business viewpoint of the architecture is discussed, which will guide the development of the application components and support the decision making process of the stakeholders involved.

11.1 High-Level View

There are a variety of smart farming systems and platforms already deployed, employing many different communication, sensing and data processing technologies. However, building a new master system that has the ability to also incorporate other existing systems is a near impossible task, also due to the complexity/heterogeneity of the agrifood sector, when it comes issues such as scalability and governance. To tackle these, DEMETER proposes an overarching approach that integrates heterogeneous technologies, platforms and systems, while supporting fluid data exchange across the entire agrifood chain, addressing scalability and governance of ownership. As described in the previous section, these goals are delivered through the Agricultural Interoperability Space. The proposed approach enables existing Agriculture Knowledge Information Systems (AKISs) to continue their operation, but also allows those systems to make available and consume data from other cooperating systems. Additionally, newer technologies and services can be exposed that may be of interest to cooperating AKISs. This is more realistic and viable in terms of usability, market adoption, and sustainability. In order to realize this approach and address the requirements presented in Section 9, the following two core objectives need to be fulfilled by the proposed solution:

- Allow existing AKISs to offer their data to and consume data from their counterparts, providing also the means to incentivize AKISs for sharing data by ensuring data integrity and valorisation, giving them also the chance to make some profit.
- Extensive use of virtualization containers for services should be made to ensure rapid deployment, portability and scaling once required.

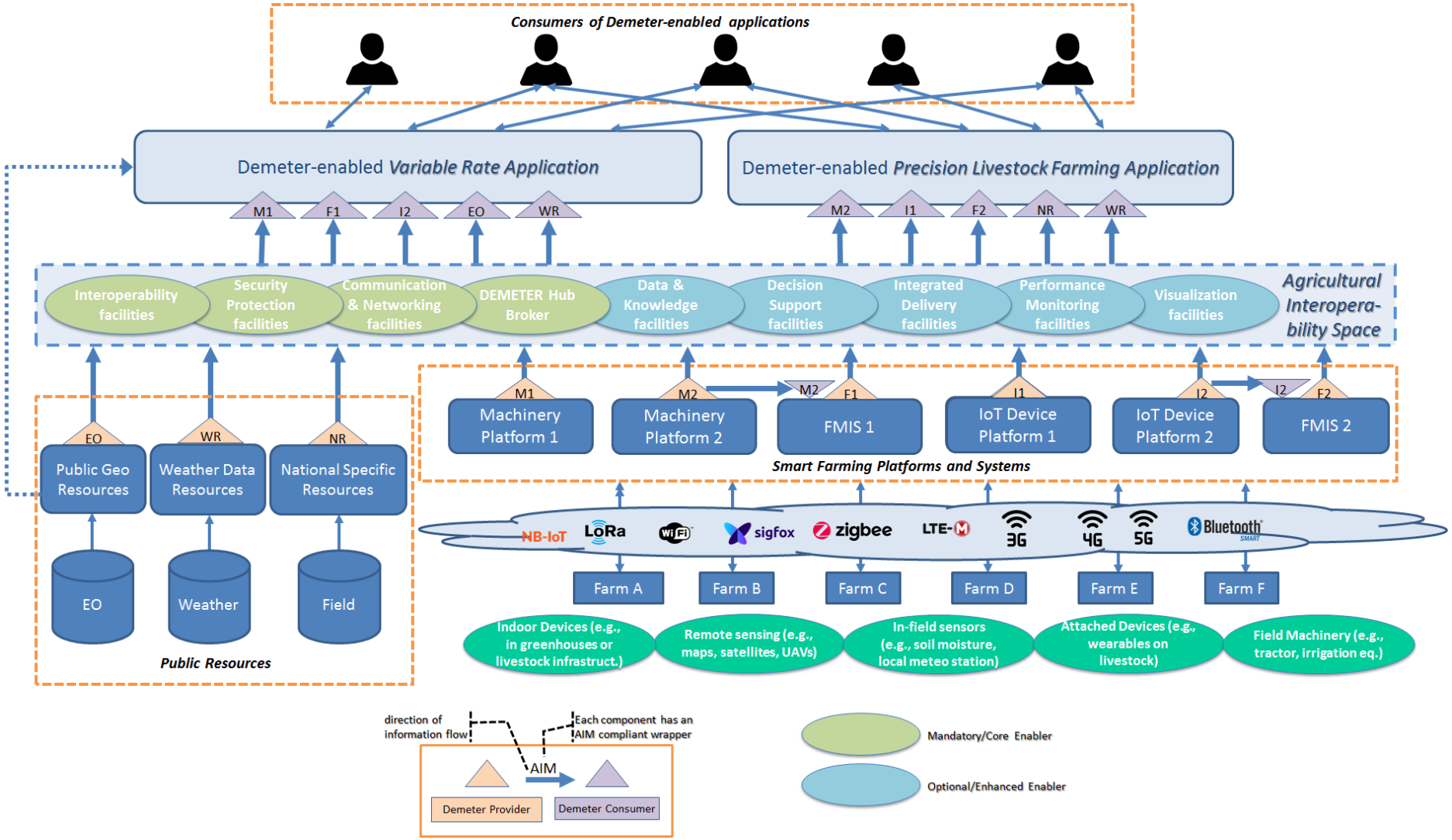


Figure 31. High-level view of DEMETER Reference Architecture instantiation example

The proposed architecture (Figure 31) consists of a DEMETER Provider and a DEMETER Consumer service based on the architecture model introduced by the Industrial Data Space (IDS)[17], then further specified by the International Data Space Association (IDSA)[18], which is the continuation of IDS. However, DEMETER Provider and DEMETER Consumer services further extend their applications by supporting AKISs to also expose and consume data. Rapid deployment and decommissioning are highly beneficial for survey services that might not require a continuous feed from a particular AKIS. Such a service would deploy and start a DEMETER Consumer service for that particular AKIS, gather necessary information, and then stop the service. The service will be packaged into a lightweight container along with all the software necessary to support self-contained deployment of the service (runtime environment, libraries for supported communication protocols, encryption techniques, etc.).

As data interoperability is of critical importance, the proposed solution provides the necessary data translation mechanisms combining the use of a semantic data model (Agriculture Information Model — AIM) developed by DEMETER, along with the respective data translation/management/inference mechanisms adopting widespread standardised solutions such as NGSI-LD[19], Saref4Agri[20], ADAPT[21], etc. In order to enable interoperability of heterogeneous data handling approaches, the DEMETER provider-consumer services, deployed on various AKISs, translate and exchange data based on the AIM common data format with the utilization of lightweight data wrappers/translators. For this conversion to be feasible, each AKIS needs to provide the specifications of the utilized data model-semantics, or it should parse returning content in the AIM format. The AIM is not built ab initio but incorporates and extends existing ontologies and vocabularies already available for this domain.

DEMETER provider-consumer services maintain the necessary mechanisms for satisfying data security and privacy concerns (cf. below). They need to be trusted to be deployed and hosted by the AKIS on their own cyber-premises (i.e., hosting environments) following the principle that moving processing capabilities is easier than moving data themselves. This is also an inherent data privacy protection feature as the owner of the data maintains the control/decision of which data are allowed to be shared with other entities. The services need to provide privacy and security functionalities, including user authentication and access authorization. Once a DEMETER-enabled application is implemented, the final version at a production level can be discovered by consumers (e.g., Farmers, Agronomists, Cooperatives, etc.) through the DEMETER Dashboard, which is also used by these stakeholders to provide their feedback regarding the perceived experience and added value.

For simplicity, Figure 31 presents only some of the platforms that can be integrated in the DEMETER Reference Architecture, thus representing a specific instantiation of the architecture deployed to serve the needs of one pilot site for example. However, apart from platforms, DEMETER service logic blocks are made available and can be used by the interested parties (e.g., data/knowledge facilities), as well as any other 3rd Party resource. All the registered resources are made available to the

[17] http://www.industrialdataspace.org/
[18] https://www.internationaldataspaces.org/
[19] ETSI. Context Information Management (CIM); NGSI-LD API. Available online: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf
[20] ETSI. SmartM2M; Extension to SAREF; Part 6: Smart Agriculture and Food Chain Domain. Available online: https://www.etsi.org/deliver/etsi_ts/103400_103499/10341006/01.01.01_60/ts_10341006v010101p.pdf
[21] ADAPT - Agricultural Data Application Programming Toolkit, Homepage URL: https://adaptframework.org/

developers through the DEMETER Enabler Hub, presented in the next subsection of this deliverable report; these are annotated with rich metadata that describe the capabilities (or constraints) of these resources thus guiding the deployment of DEMETER apps based on the adopted technologies as well as information regarding ownership of resources that are available and the restrictions that their locations might impose during this process.

As already mentioned in Section 10, each platform, thing, service or application is represented by a DEMETER-enhanced Entity and either makes itself available via the DEMETER Enabler Hub, or consumes resources available in the Hub, or both. Thus, there are various instantiations of the DEMETER-enhanced entity that are listed hereafter:

- **DEMETER-enhanced Resource**: This entity contains the resource (platform, thing, service) that registers its capabilities to the DEMETER Enabler Hub and makes them available to interested parties. It is worth mentioning that a DEMETER-enabled Resource can make use of other Enablers registered in the DEMETER Enabler Hub to enhance its features.
- **DEMETER-enabled Service**: A DEMETER-enabled Service is a 3rd Party service that is provided by a stakeholder external to the DEMETER project, which is integrated to the DEMETER ecosystem. It can both register its Service Logic to the DEMETER Enabler Hub, thus making it discoverable by interested parties, as well as discover other DEMETER Enablers via the DEMETER Enabler Hub and directly consume their exposed interfaces without any interoperability implications.
- **DEMETER-enabled Application**: The "Application Logic and User Interfaces" are DEMETER ignorant and are provided by an application provider external to DEMETER. The DEMETER-enabled application can communicate with the DEMETER eco-system and browse its Enabler Hub to discover available resources that are compatible with and registered in DEMETER. End-users directly access the user interfaces provided by these applications. Furthermore, users can consume functionality exposed by the DEMETER-enabled Resources (or resources in general, including data) only through these applications.

The following figure (Figure 32) illustrates where these entities are positioned in the DEMETER ecosystem and how they interact with stakeholders, with the DEMETER Enabler Hub and with each other. The human users (stakeholders) can have direct access to the DEMETER Dashboard, in order to use SOCS or AIS facilities, as well as to any DEMETER-enabled application. The interaction of the various entities with the DEMETER Enabler Hub are required primarily in the registration and discovery processes, so that these are made available in DEH and are discoverable by interested parties. Once the interested DEMETER consumer discovers the enablers and other resources it aims to use, these are packed for delivery along with the necessary facilities that support this development and integration process; these deployment and runtime facilities are provided by the hub. Please notice that the DEMETER-enabled application can directly interact with the DEMETER-enabled services or DEMETER-enhanced resources it aims to consume, once these are discovered in DEH. The same is valid for the DEMETER-enabled service that consumes DEMETER-enhanced resources. There is however the option, should the interested DEMETER provider wish to do so, to enable access to its resources to interested stakeholders only via the DEMETER hub and not directly with each other.

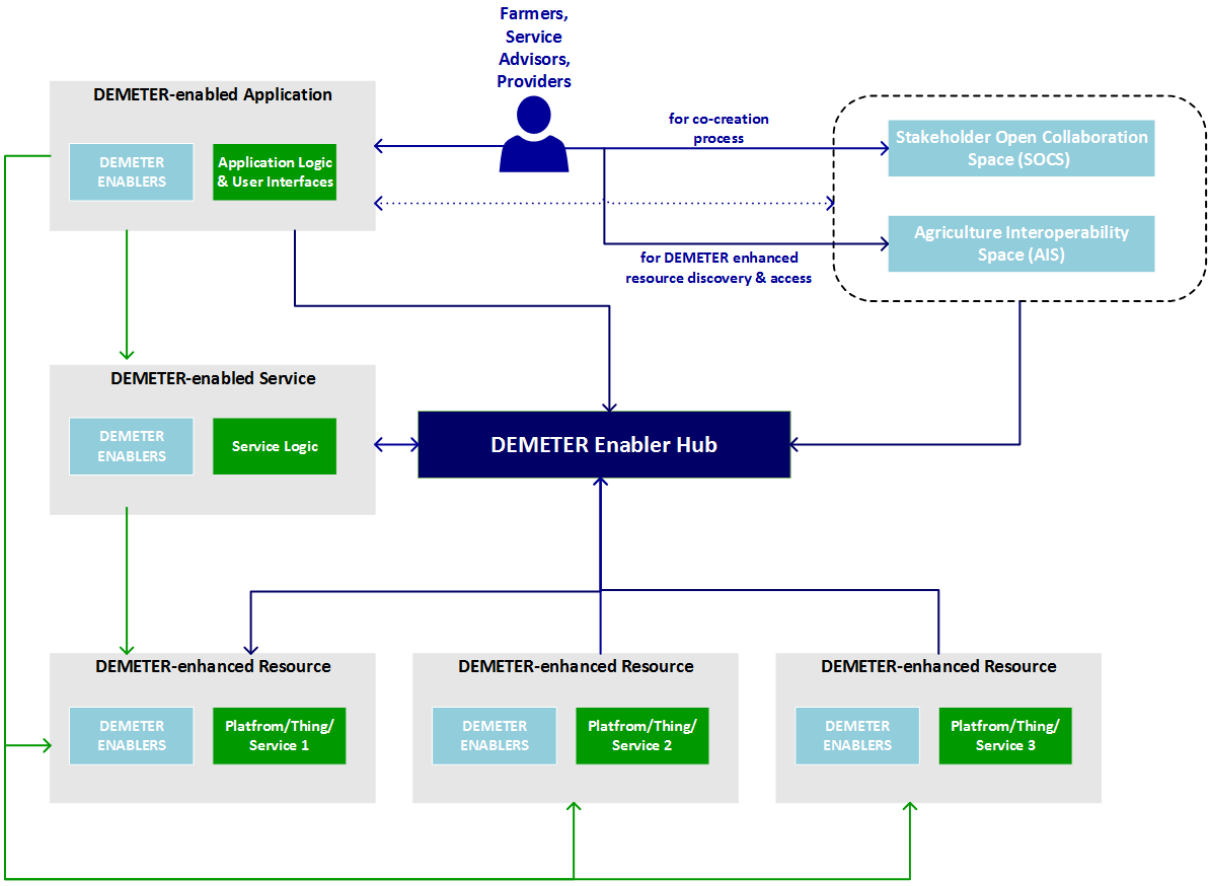


Figure 32. Positioning and interoperation of entities (i.e., applications, services, resources) enhanced/enabled and/or made available via DEMETER

11.2 Functional View

In order to implement the high-level view of the architecture and to implement the objectives and vision of it, DEMETER needs to provide several facilities/modules that interact with each other, with the various stakeholders as well as with a number of existing devices, platforms, systems and data sources. In the following figure (Figure 33), the main functional blocks of the DEMETER Reference Architecture are depicted, along with the external entities involved and their respective interactions; these constitute the **functional view** of the architecture.

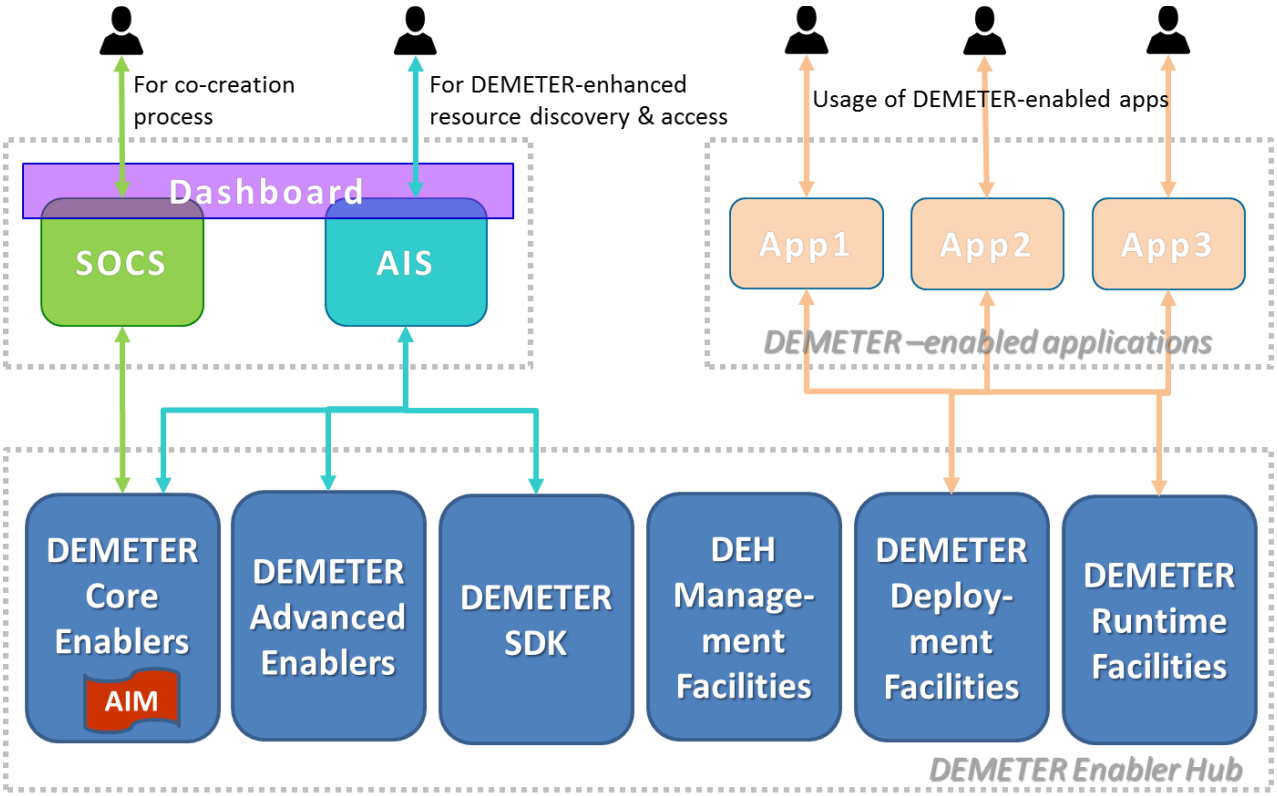


Figure 33. Main functional blocks of the DEMETER Reference Architecture

Before we describe the functional components of this architecture, we need to link them to the high-level view described in the previous section. According to this view, it is necessary to create appropriate DEMETER compliant wrappers in order to use the available (public) data resources and the external devices (things in general) and systems that we want to include in DEMETER enabled applications. On top of these resources sits the communication infrastructure responsible for handling any communication between DEMETER and these external resources. These resources are DEMETER agnostic in the sense that they do not need to comply to the DEMETER AIM and models in general. However, in order to be integrated into a DEMETER enabled application, they need to be paired with the aforementioned data wrapper/translator, which will translate their data format to/from the DEMETER AIM model. The respective facilities that deliver this are wrapped in the so-called DEMETER Provider, depicted by triangles in Figure 31 (see previous subsection). These facilities will of course be provided by DEMETER in support of all pilots, but in general, for other domains, they will be developed and offered through DEMETER by 3rd Party developers and stakeholders, who can access DEMETER in order to develop and offer these via AIS.

Now, in order to support this creation process of appropriate wrappers for the external resources (e.g., platforms, things, services, applications), DEMETER's facilities will support the development process and will ensure semantic interoperability always based on the DEMETER AIM model; to ensure that these objectives are met, an appropriate Software Development Kit (SDK) will be offered to 3rd Party developers that will facilitate the development of DEMETER-enabled applications, as well as the development of the appropriate DEMETER Consumers and Providers for the instantiation of the appropriate DEMETER-enhancing service.

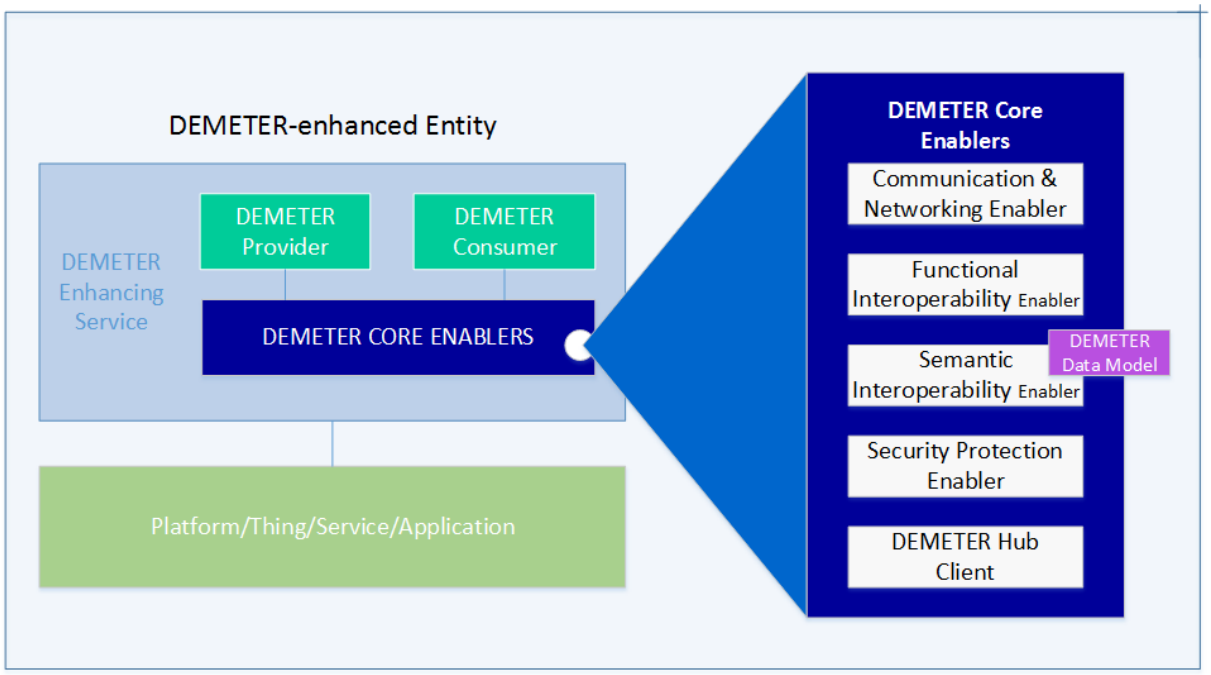


Figure 34. Functional view of DEMETER enhanced Entities with the included DEMETER Core Enablers

In addition to the SDK, DEMETER will also offer a set of **Core Enablers** needed for creating any DEMETER applications. These are listed in the figure above, which presents the architecture of a DEMETER-enhanced Entity in order to be compliant with DEMETER (and thus able to register with the DEMETER Hub). As previously discussed, on top of the existing resources, e.g., platforms, devices, services or applications, it is necessary to create a DEMETER wrapper that will allow the resource to interoperate with other DEMETER Entities. In order to facilitate this creation, DEMETER offers a list of core enablers. These enablers are mandatory for any interested stakeholder that wishes to expose or share its own resources, and provide support for: communication and networking, which will facilitate the actual communication of this resource with other DEMETER entities, semantic & functional interoperability, in order that the communication complies with the DEMETER AIM model, and security protection, e.g., for secure transfer of sensitive data or to prevent access to unauthorized entities, while they can also include a Client for the DEMETER Hub, for any information that needs to be communicated with the hub's runtime facilities.

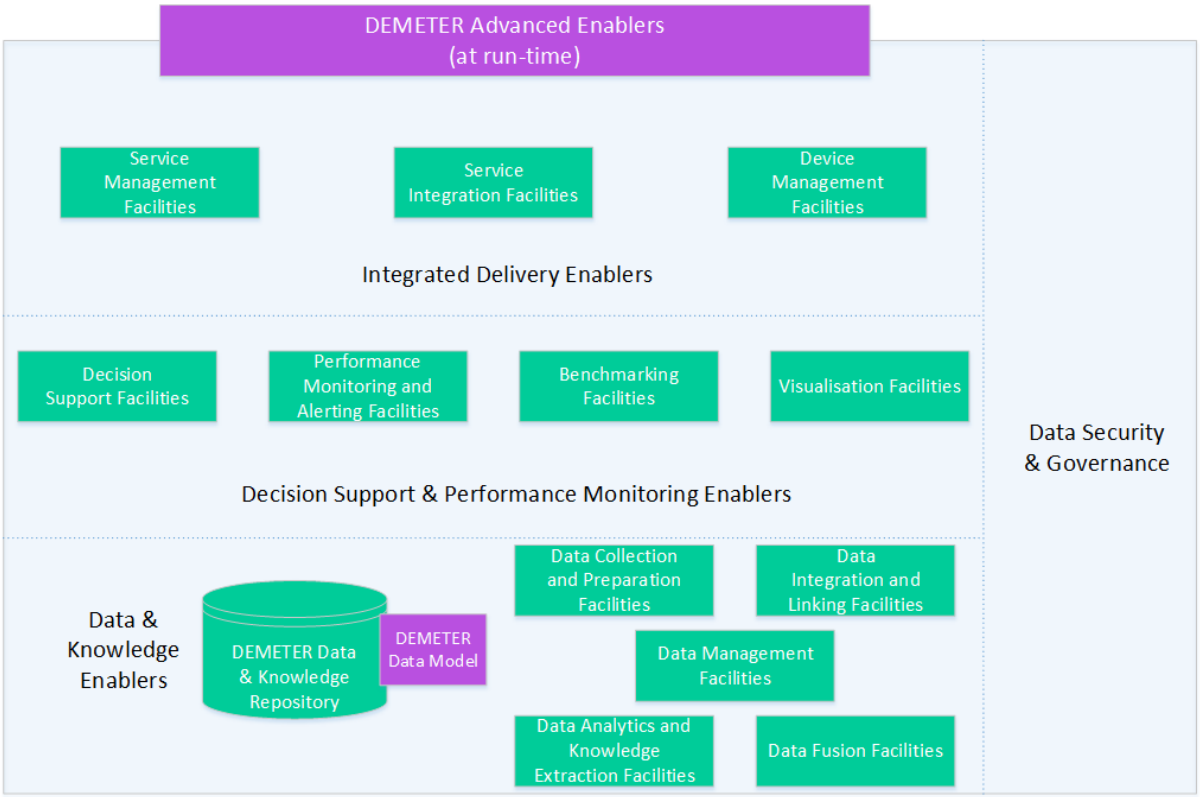


Figure 35: Advanced Enablers offered by DEMETER

DEMETER will also offer 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.

First, the *Data & Knowledge Enablers* will be responsible for Collecting and Curating data from the various sources that the DEMETER developers and stakeholders have been registered for. To elaborate, the Data Collection & Preparation enabler will collect, curate and prepare the data collected, while the Data Integration & Linking together with the Data Fusion enabler will integrate and fuse the data collected from heterogeneous sources. Furthermore, Data Management will be guaranteed according to the users' stated preferences, while a Data Analytics & Knowledge Extraction component will be available to any apps developed which will offer facilities that allow the processing of the fused data.

Second, the *Decision Support and Performance Monitoring Enablers* will provide to the DEMETER developers and stakeholders the availability to choose the Decision Support (DS) mechanisms of their interest (among the set of DS that DEMETER implements); any developer will of course have the option to use her own DS system instead of the offered ones. In addition, it will also provide Performance, Monitoring and Alerting facilities as well as Benchmarking facilities, in order to monitor at runtime the performance of DEMETER entities, such as the aforementioned decision support algorithms. Finally, visualisation enablers will complete this class of DS related enablers: they will be instrumental in conveying the information and actions taken automatically (or needed) to the final users of the DEMETER applications, such as the farmers.

Third, through the *Integrated Delivery Enablers* developers and stakeholders will be able to access the services and the devices that they desire while they will be able to include any other optional

verticals related to identification, construction, operation, positioning systems, etc. Note that these enablers will be part of the DEMETER-enabled services that will be developed specifically for the DEMETER pilots, but they can be reused in part or in their entirety for other applications by 3rd Party developers.

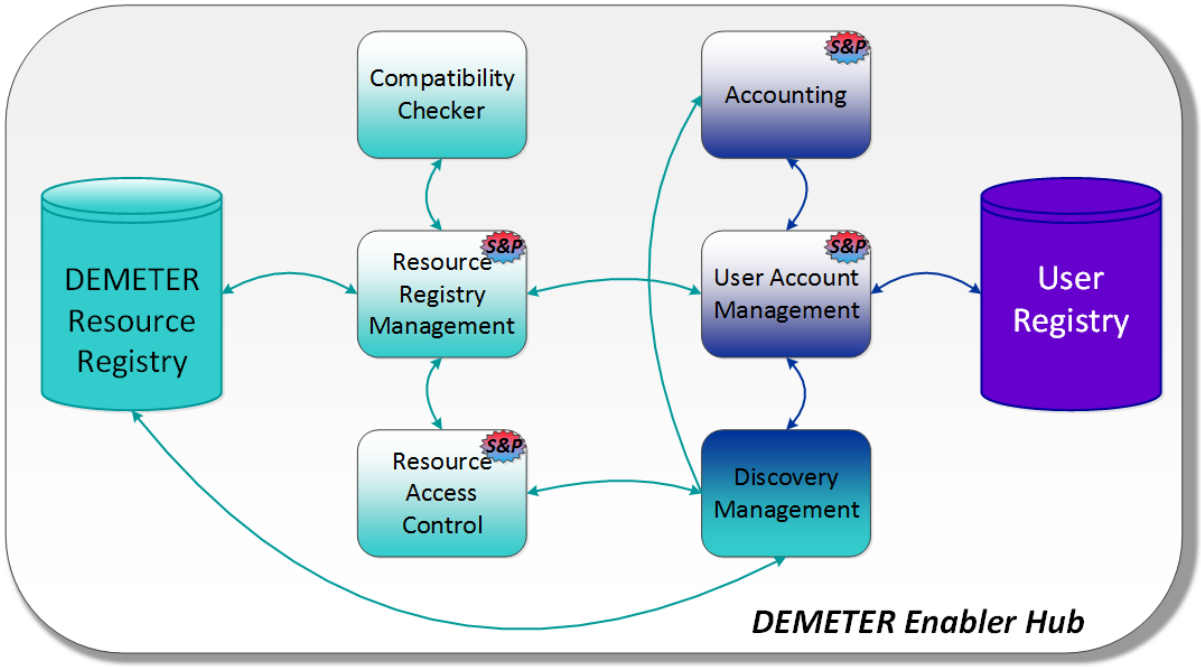


Figure 36. Internal Management Components and Registries of the DEMETER Enabler Hub (DEH)

So far, we have described the components of the DEMETER enabler hub that deal with the creation of the DEMETER entities and the entities themselves. The registration and management of these entities is the task of the DEH management facilities component. The internal subcomponents of this are presented in Figure 36. Users first need to register with the User Account Management, which updates the user registry DB before they can consume DEMETER resources or get apps and services, or register a new DEMETER enabled entity or resource that they wish to offer through the hub. To accomplish the latter, they interact with the entity registry management component. The new entity submitted for registration is first processed by the compatibility checker component in order to verify that the entity is compatible with the DEMETER AIM and any other requirement as set by the DEMETER system (the most important of which is that it uses the necessary parts of the DEMETER core enablers, or is compatible with them, and thus adheres e.g., to the security and communications standards/protocols). Then, the resource register DB is updated by the management component with the data for the new entity and, subsequently, the access control component is updated by the registry management with the information regarding the entity owner's preferences for the submitted entity, including what access rights are to be enforced. In this way the user can limit which users can discover the new entity and in which way it may be used.

Afterwards, when a user wants to discover DEMETER enabled entities, the user account management component queries the discovery management component which in turn will get the permissions on what entities to display from the access control component; afterwards, depending on the rights provided by the access control, it queries the resource registry through the entity

management component in order to get the entities to which the current user has access (rights) to use. Should an entity then be contracted, the accounting component will be informed so that the appropriate compensation be provided for using the entity. All these processes of how the hub components are detailed in the following subsection which presents the *process view* of the DEMETER Reference Architecture.

Finally, there is another component, which will permeate and be used by several of the hubs' component in this ecosystem, will deal with the privacy and security requirements that need to be addressed. For example, such concerns exist in the user and resource registry components, or in the accounting component as sensitive data (e.g., billing data) need to be protected. The respective facilities are depicted as "S&P" (i.e., Security and Privacy) modules within the appropriate components of the management facilities in this figure. [All these concerns are to be addressed by T2.4 (Data Protection, Privacy, Traceability and Governance Management) and T3.4 (Connectivity and Security Framework).]

For the users to contact the enabler hub and perform all the functions described previously, they need to use the appropriate view of the dashboard. More specifically, The **Agricultural Interoperability Space (AIS)** view should be accessed by users, such as 3rd Party developers, who want to develop DEMETER entities and offer them through the hub and those who want to deploy and offer complete solutions (apps). To this end, they will also be able to access the DEMETER semantic interoperability tools, such as the enabler creation SDK and the description of AIM by going through the AIS view of the dashboard. On the other hand, the **Stakeholders Open Collaboration Space (SOCS)** view will be used by farmers, service advisors and providers; these can select, together, the most appropriate set of tools, devices, components, data sources and available apps, all of which are registered and offered through the DEMETER Enabler Hub.

Once such tools and applications are selected, they are then made available through the DEMETER hub deployment and their execution supported by the hub runtime facilities. These facilities will be developed by T3.5 and detailed in deliverable D3.2. Their task will be to link together the various enablers composing each application, e.g., by offering a URL or URI where an enabler can be contacted, in order to get data from it, or to download a specific service; in this way they will facilitate the deployment of the final applications. The runtime facilities will be able to receive data back from the application and the individual entities they are composed of in order to gauge whether the various entities operate according to their specifications and receive ratings back that can update the information available in the registry for each resource available through it.

11.3 Process View

The **process view** deals with the dynamic aspects of a system, describes the system processes and their interactions, and focuses on the run time behaviour of the system. The process view is designed for people designing the whole system and then integrating the subsystems or the system into a system of systems. This view shows tasks and processes that the system has, interfaces to the outside world and/or between components within the system, the messages sent and received, and how performance, availability, fault-tolerance, and integrity are being addressed.

In DEMETER, for this first release of the Reference Architecture, we identified two processes that are related to DEMETER Enablers:

1. Enabler Registration
2. Enabler Discovery and Usage

For these two high-level processes we have created their corresponding high-level Activity and Sequence diagrams. These diagrams are presented in the following subsections.

11.3.1 Enabler registration

11.3.1.1 Activity diagram

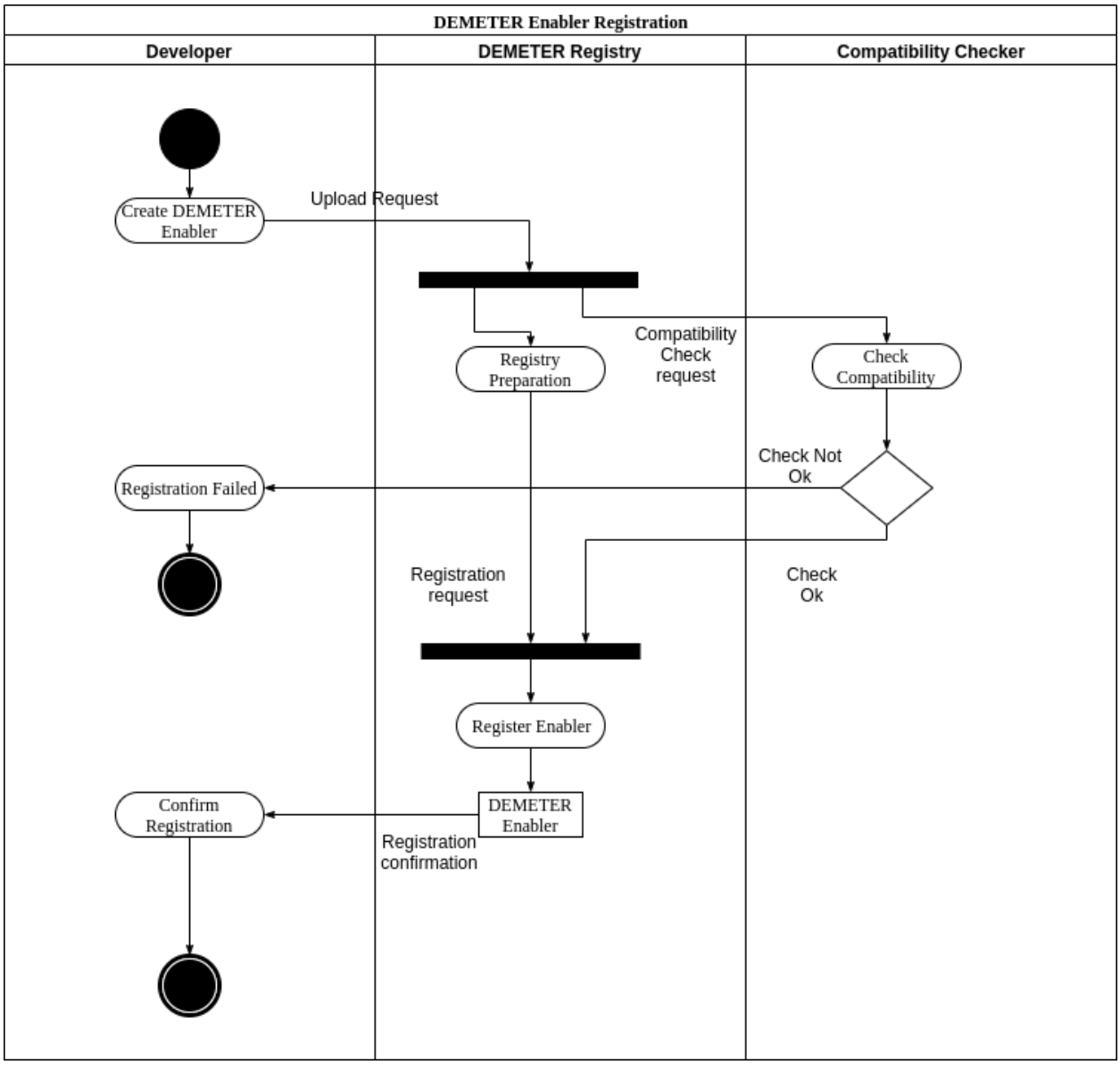


Figure 37. DEMETER Enabler Registration Activity Diagram

The diagram shows the three “systems” that are involved in this process/activity:

- Developer, the development teams that make use of DEMETER to create their own DEMETER Enablers.
- DEMETER Enabler Registry, where all DEMETER Enablers are stored.

- Compatibility checker which is the DEMETER service for validating the compatibility of an Enabler before it is stored in the registry.

The Developer implements an Enabler and issues a request for it to get registered as a DEMETER Enabler on the DEH registry. DEMETER backend services (e.g., Enabler Registry and Compatibility in Enabler HUB) prepare the Enabler and request a compatibility check so as to assess whether the Enabler will be registered as a DEMETER Enabler in the Enabler Registry or not.

11.3.1.2 Sequence diagram

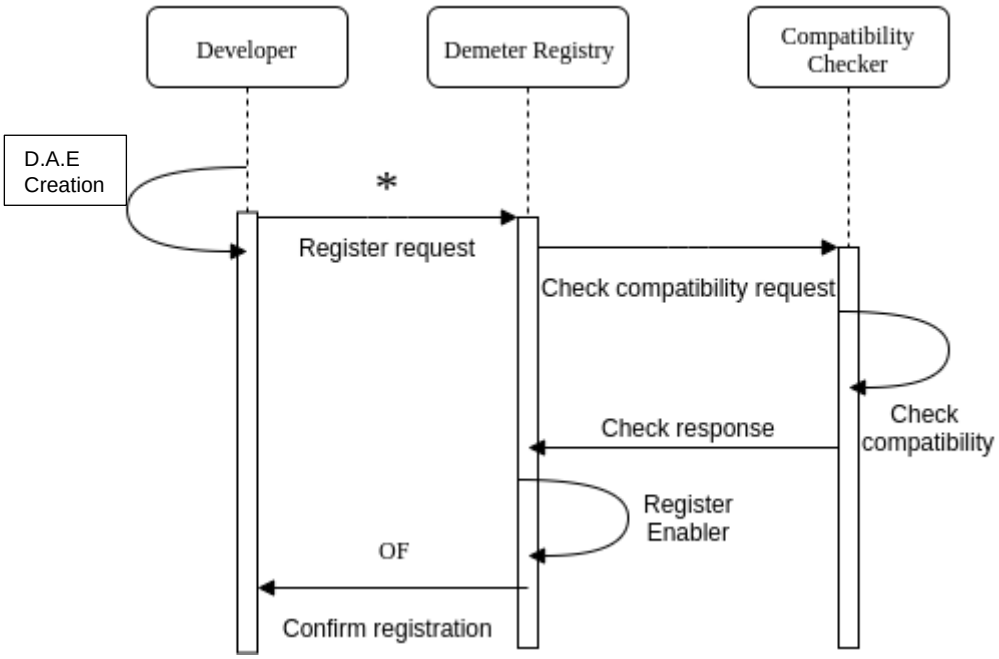


Figure 38. DEMETER Enabler Registration Sequence Diagram

The diagram depicts the sequence in the interaction between the three different systems from the moment a developer creates a resource, i.e., a DEMETER Advanced Enabler (DAE) until it gets notified that this resource has been registered to DEMETER's Enabler Registry.

11.3.2 Enabler Discovery and Usage

11.3.2.1 Activity diagram

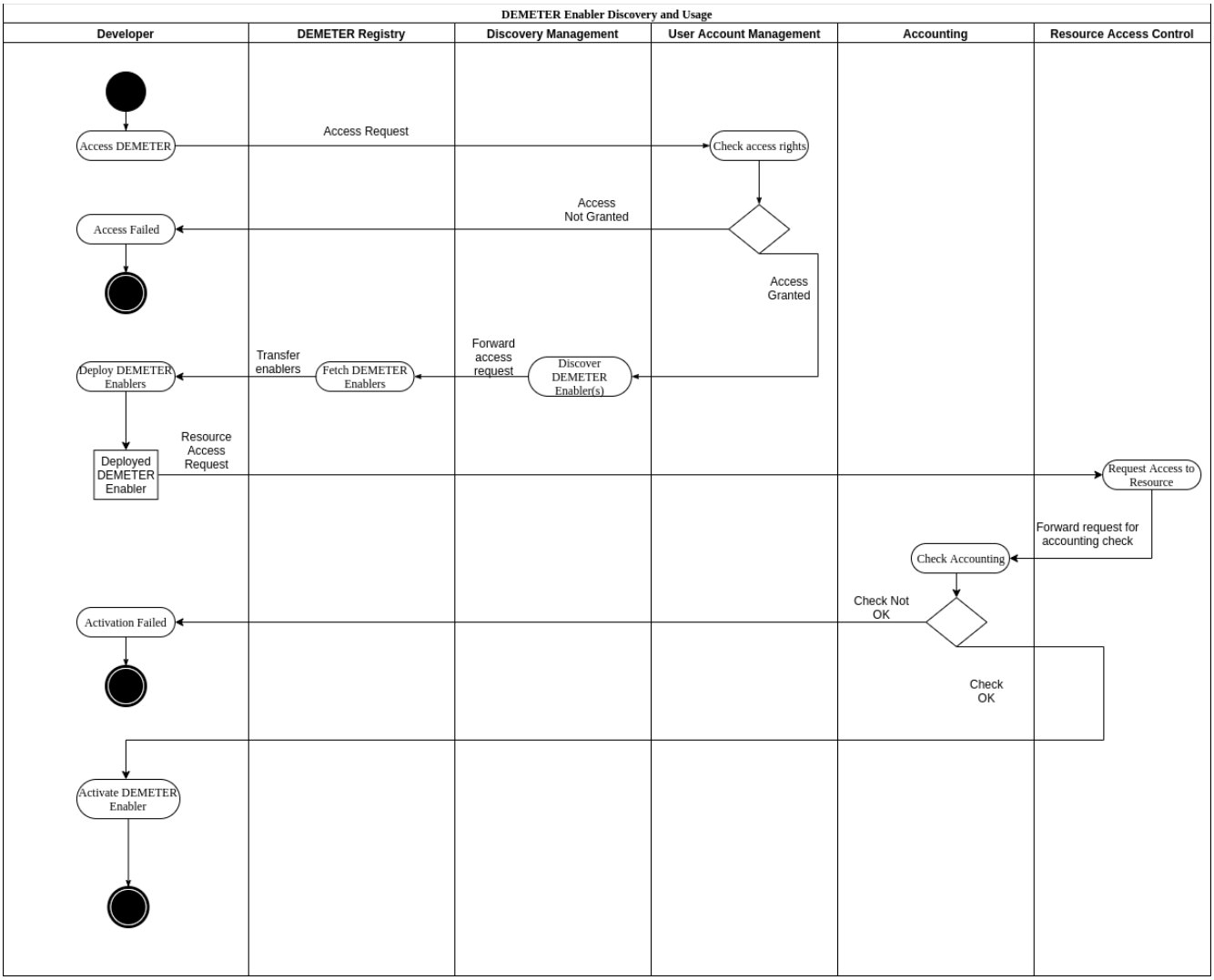


Figure 39. DEMETER Enabler Discovery and Usage Activity Diagram

The diagram shows the six "systems" that are involved in this process/activity:

- Developer, the development teams that make use of DEMETER to create their own DEMETER Enablers.
- DEMETER Enabler Registry, where all DEMETER Enablers are stored.
- Discovery Management, which offers the enabler's discovery service.
- User Account Management, which provides all the user account related services, granting different access rights to different user roles.
- Accounting, which includes functionality for (monetary) transactions between actors
- Resource Access Control, which controls users' access to specific resources (services, devices, etc)

The Developer access DEMETER's dashboard and, if allowed, searches for Enablers through the Discovery Management system. Enablers are fetched to the Developer's premises from the Registry and then a resource access request is issued to be checked and validated by the Resource access control system and the Accounting system. If the user has indeed permission to access that resource, the DEMETER Enabler is activated providing access to that particular resource.

11.3.2.2 Sequence diagram

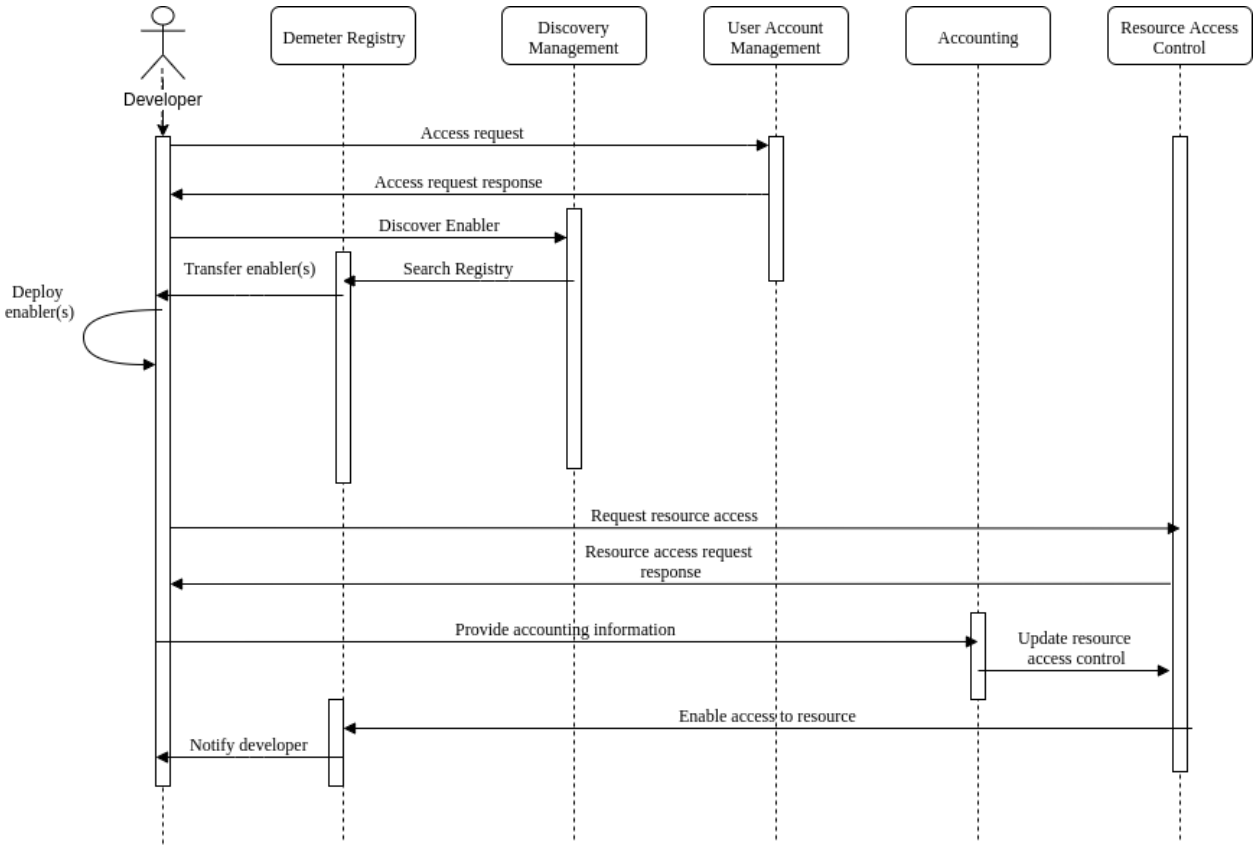


Figure 40. DEMETER Enabler Discovery and Usage Sequence Diagram

The diagram depicts the sequence in the interaction between the six different systems from the moment a developer accesses DEMETER dashboard to discover Enablers up to the moment that the developer gets access to a DEMETER enabled resource.

11.4 Data View

This section aims to provide a description of data view representation of DEMETER Reference Architecture, based on the awareness that this representation is connected to a series of concerns such as allowing interoperability between internal and external processes in DEMETER, such as making information accessible at all architectural levels (e.g., between applications and services, user and application, users and services) up to the data visualisation for the end-users through the DEMETER Dashboards. Basically, this view is about storage architecture, data retrieval, processing, storage and security management. The view highlights the data flows and which components will be needed to support and manage the main processes, such as archiving and processing.

The methodology applied for the representation of this view takes into account the following aspects:

- The definition of the scope and requirements helps to formulate the representation of DEMETER entities model. A clear understanding of needs 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 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. On the one hand the identified data spaces are a semantic database or **DEMETER Data & Knowledge Repository**, and DEMETER **User Registry & DEMETER Resource Registry**.

- D3.1 will follow an iterative approach, therefore more complete and comprehensive descriptions of the relationships among the identified entities will be refined in the future.
- The following descriptions focus on the point of view of the Stakeholders (i.e., users accessing the information via the DEMETER Dashboard), and only the main entities involved in the data storage, retrieval and processing.

The **DEMETER Data & Knowledge Repository** addresses the problem of the physical representation of semantic model, allowing a mapping between the resources that will be acquired through DEMETER's Communication & Networking components, and the AIM model used to model digital resources and therefore the entities that will be used as an integral part of the information flows between the business processes. The use of appropriate data exchange models (e.g., RDF), knowledge representation languages (e.g., SKOS, RDFS, OWL) and rule languages (e.g., SWRL or OWL-RL), which would allow semantic querying of data, will be applied in the design process to define the semantic model, the technology and tools to be used.

The **DEMETER User & Resource Registry** represents a database management system that structures data in tables with certain properties. This support will contain information relating to users, such as personal data, credentials, and so on (all aspects related to GDPR and in general to the security of personal data are better described in section 14 of this deliverable). It is immediately clear that the attention in the design of this store must be very rigorous and ensure a certain attention to the level of security of the information contained therein. Principles of encrypting of information could be used, or only those deemed most at risk for the security of personal data of a user who decides to register on the DEEMETERS' System.

The diagram below brings together the above concepts, such as entities, information and their interactions within business processes, in order to structure the data flows between them:

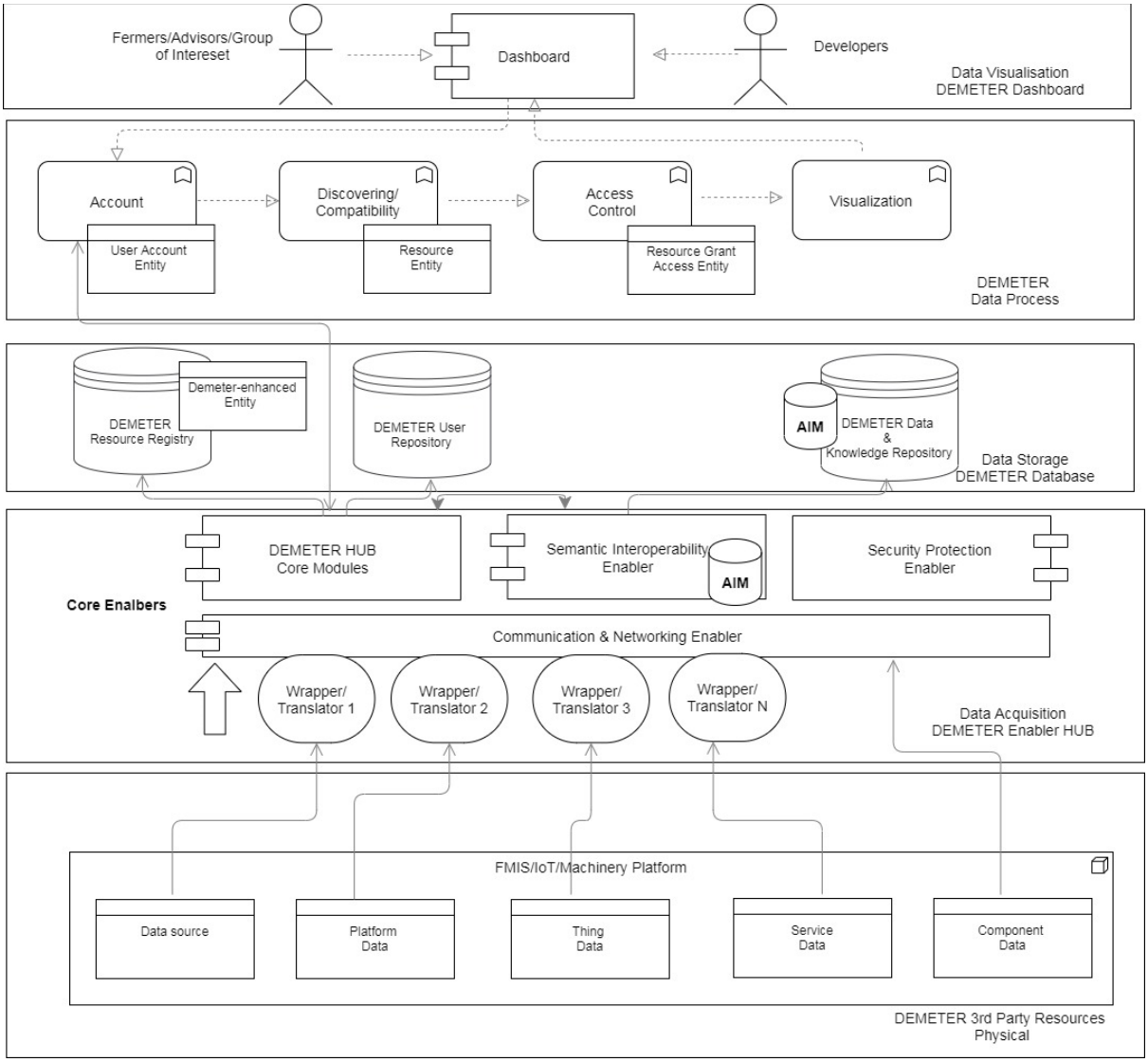


Figure 41. DEMETER Main Data Flows

Any third-party resources (e.g., Thing, Platform, Service, Component), that can feed DEMETER with its data communicates with the DEH components for data acquisition. Specific wrapper/translators will be made available in DEMETER data acquisition layer through the Communication & Networking Enabler, in order to allow translation from formats that are not compliant and therefore not aligned with the DEMETER AIM model, to be enriched and semantically aligned the data transmitted based on this model. This will allow data interoperability by combining the use of a semantic data model (AIM) with the respective data translation/management/inference mechanisms that adopt other standardized solutions. Therefore, to allow for full interoperability of heterogeneous data modelling/semantics approaches, DEMETER will provide the necessary facilities to support data translation and exchange according to the AIM format.

The DEH APIs layer responsible for data acquisition, but more generally all the components of Communication & Networking Enabler and therefore everything that interacts with the transmission and/or data acquisition from the physical layer to the digital ones, will be subject to control on data security, and on systems security policies in general that T2.4 will define for this, but also for all other levels architecture that provide for interaction of components and therefore data exchange.

The data, acquired and conforming to the model, can begin to be used by the other internal components of DEMETER including the core components of DEH. These modules or DEH Core Modules, will be responsible for data (DEMETER-enhanced entities) management, allowing their storage and recovery for instance when an end-user using the Dashboards in the upper layer will need discover the DEMETER available resources. The internal DEH processes will define all possible operations on the acquired data, interfacing visualisation Dashboard (UI) with the backend components of DEH (services layer). The data encapsulated in an 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 who will use them to power all business processes and meet all defined use case scenarios.

11.5 Deployment View

The **Deployment view** depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer as well as the physical connections between these components.

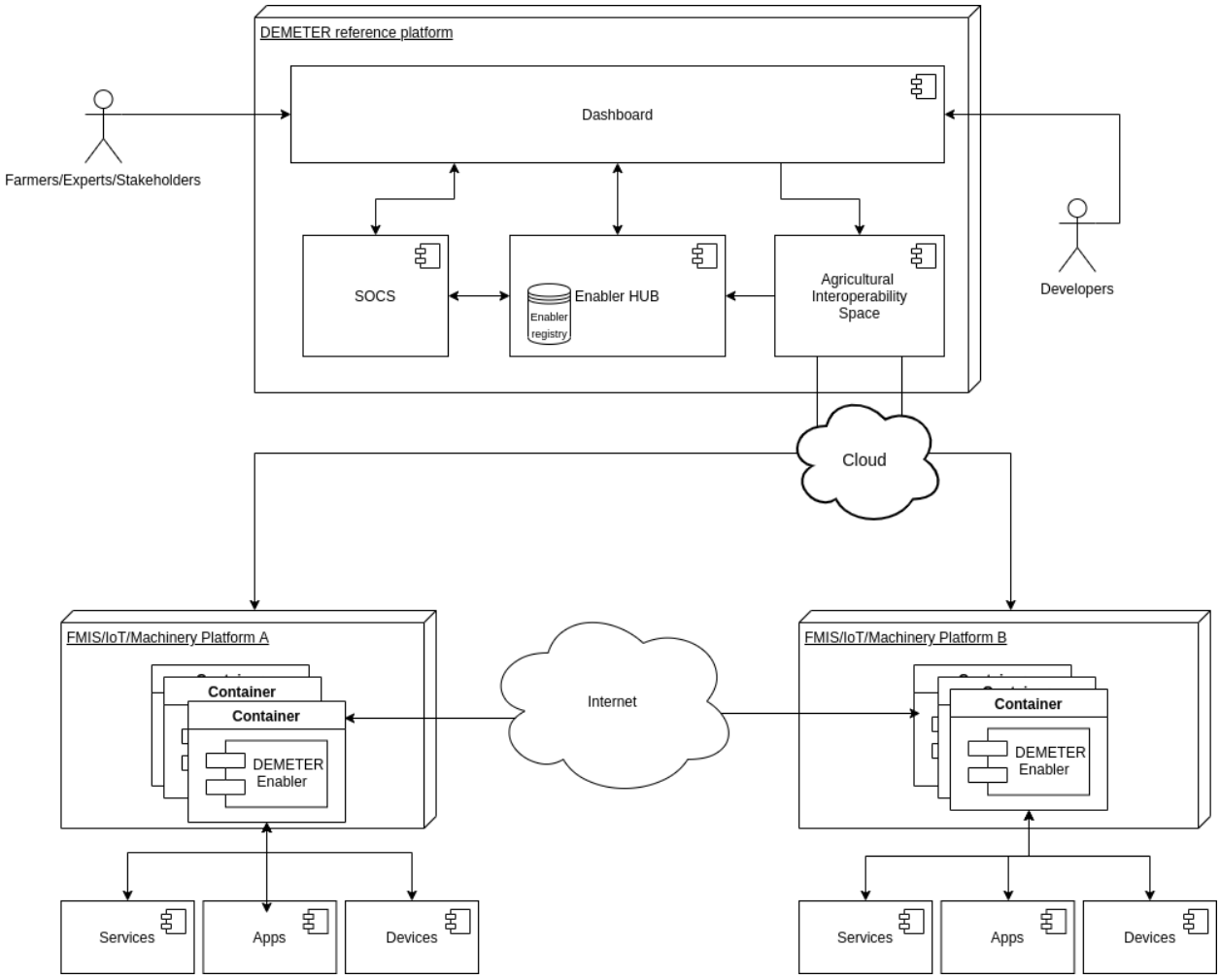


Figure 42. Deployment diagram

In DEMETER we identified the nodes that will be present during the run-time of the execution of the whole system along with the components that participate, both internal and external. In the

DEMETER's deployment diagram above, the locations where the several software artefacts reside are depicted. It is also demonstrated that the interconnections between the participating nodes are internet based.

DEMETER's reference platform include the Dashboard which provides the user interface to the users and the developers and exposes the functionality of the other software artefacts included in the reference platform, SOCS, Enabler HUB, and AIS.

FMIS/IoT/Machinery platforms can connect with the AIS so as to get access and deploy DEMETER Enablers while their already deployed Enablers (in containers) which provide access to their resources (services, apps, devices) communicate directly via the internet.

11.6 Business view

The business view, described below, focuses on the functional aspects of the DEMETER System supporting the decision-making process among the main components defined in DEMETER RA. These components, such as DEH, SOCS and AIS, will provide tools and services suitable for both business understanding and its development, implementing technological solutions specifically conceived to meet the needs of DEMETER Stakeholders such as Farmers, Advisors, Developers etc.

Furthermore, this view, contributing to the development of a business model suitable for the DEMETER context, will help to understand the main activities and interconnections between the modules or to check if all have been specified the interfaces required between the components or if all the information required for the execution of business processes is available).

Clear business design can help developers understand the system. In the process of designing the business architecture the building of the processes structure must be determined according to the needs of the user who in a business view represents the true starting point. Consequently, having relatively individual requirements associated with the individual services to be implemented helps to design a system that is not too complex.

To cover all the fundamental aspects to which the business view must respond, a whole series of aspects of a system must be considered:

- **Users**: i.e., the Human Actors involved in the system
- **Process**: that is, the user processes involved in the system
- **Function**: the functions required to support the processes
- **Information**: or the business information necessary for the proper functioning of the processes

but only a few have been taken into consideration to represent the view. One aspect among those mentioned above is to be taken into greater consideration, fundamental for the analysis and description of the existing environment and inspire all the functional but also technical solution. A process is a grouping of tasks that form an executable unit, able to realize a complex system behaviour through separate threads of control. The business processes can be described at several levels of abstraction, each level addressing different concerns. At the highest level, the process architecture can be viewed as a set of independently executing logical networks of communications, distributed across a set of resources and components, constituting the overall system.

From the relationships of these aspects follows a representative model for the business view, and a

methodological approach that has influenced and will continue to do it using an iterative approach, the logic behind which the DEMETER solution was conceived. The business model of DEMETER-based systems, for what concerns business representation has considered and covered all these aspects allow to address separately the concerns of the various

Stakeholders of DEMETER, mainly technical partners and business partners, and to handle separately the functional and non-functional requirements. The proposed approach uses a model composed of five main aspects:

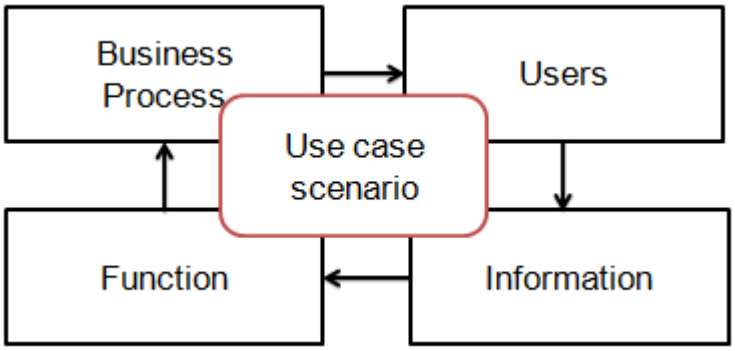


Figure 43. Aspects of the DEMETER Business Model

The model depictured in the image above uses a very simple but equally effective approach: each use case scenario, defined to meet the requirements of a system such as DEMETER, can be widely satisfied considering all the relevant business processes acting as a starting point for the development of the whole use case. The scenario in turn represents the basis and starting point for the implementation of a business process: this atomic view allows the model to be applied and therefore valid to perform all the business processes defined for the overall system. Each individual scenario will take into account the iterations among four main entities: the users (or Stakeholder), the information they need to perform functions or services within the system, and finally the outputs of a single business process that can contribute together with the other system business processes to execution of each individual use case scenario defined at the system level.

Given the above, it must be taken into account that the iteration between the DEMETER stakeholders and processes will undergo changes over time, considering a whole series of dominant factors such as sudden changes in context, constantly evolving application and technological scenarios, technologies and solutions that could change and consequently impact on this view. Reasonably therefore, the view proposed below, shows aggregate business processes extracts from the DoA and derived from the requirements of WP3, giving a rough idea of what today can be a business view of DEMETER Reference Architecture:

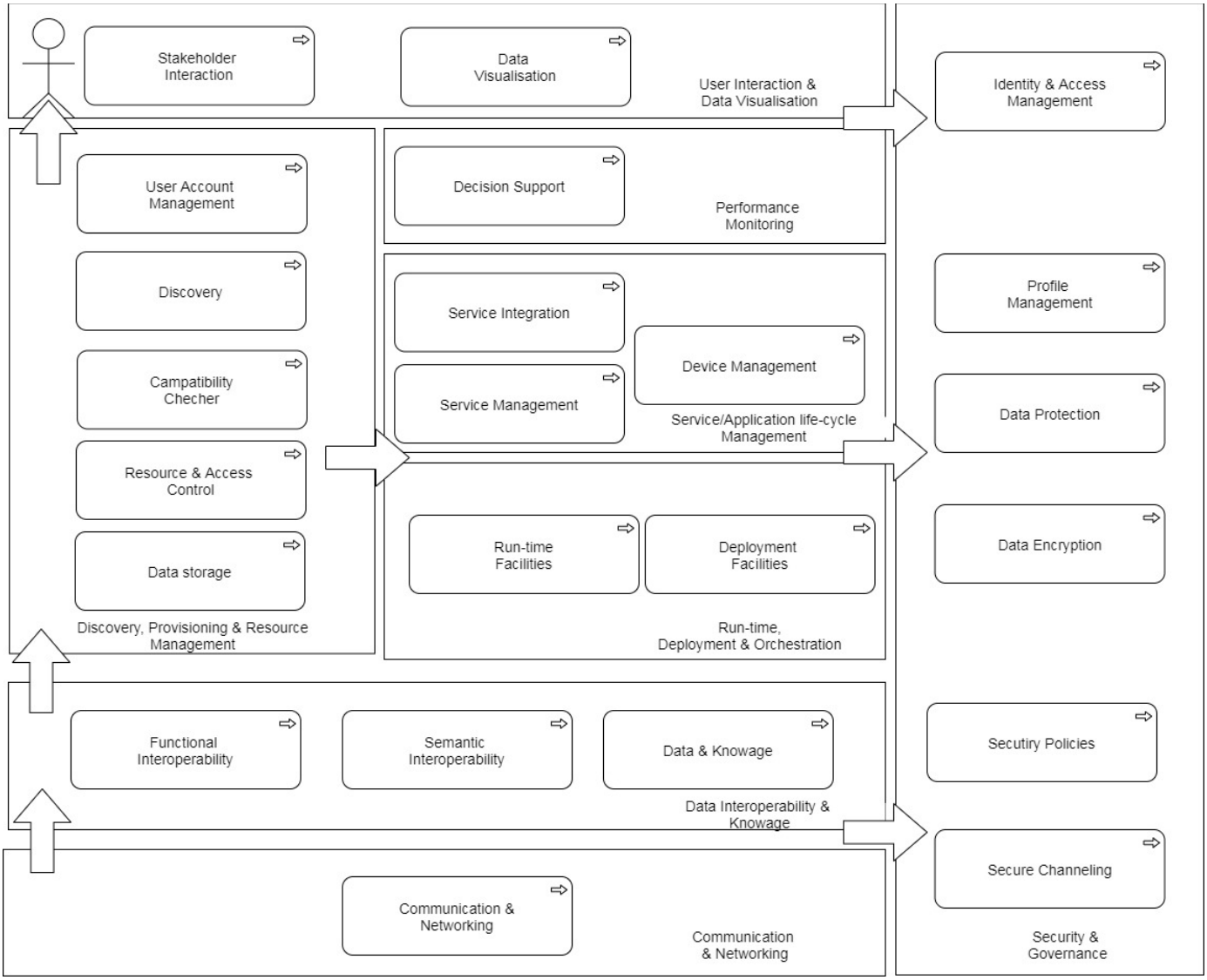


Figure 44. DEMETER Business Processes in the Reference Architecture

In this aggregate view, the processes have been grouped by domain of competence, representing the fact that every single process in DEMETER will be responsible for a specific area/context. As the image above shows, the processes are grouped by context classes:

- Communication & Networking
- Security & Governance
- Run-time Deployment & Orchestration
- Discovery, Provisioning & Resource Management
- Service/Application life-cycle Management
- Performance Monitoring
- User Interaction & Data Visualisation

The process classes can be both transversal to all the others as in the case of security processes, for example, eternal or containing data processing processes, or external or processes that interface with other processes in order, for example, to activate the semantic interoperability of data.

The **Communication & Networking** processes, refers to the transmission of digital data between

third-party applications / services and DEMETER. These processes will allow you to acquire data from multiple suppliers, allowing DEMETER to manage a network of suppliers and to exchange data with them. The physical connection between the devices and the DEMETER platform is established using internet connection and a whole series of protocols suitable for exchanging data on this type of network (e.g., LoRa, SigFox, Wifi etc.).

The **Security & Governance** processes, they will manage vertically to all processes a whole series of factors related to the security of applications, data, and infrastructure. Here we will have processes that will take care of implementing secure communication channels with the lower part of the architecture, implementing security policies to be distributed to the processes considered most at risk, acquiring data in secure mode and encrypting them in the case of user sensitive data; in addition, these processes must provide access to DEMETER resources as well as to account and user profile management services in the Frontend part of the DEMETER Platform.

The **Run-time Deployment & Orchestration** processes, they assert the distribution of new applications (software) in the DEMETER System.

The **Discovery, Provisioning & Resource Management** processes, they allow provisioning and discovery of DEMETER-enabled entities, resource management through operations such as retrieval/insert in the Resource Registry.

The **Service/Application life-cycle Management** processes, which allow the management and integration of DEMETER-enabled services, and the management of devices registered and offered through the DEH.

The **User Interaction & Data Visualisation** processes, takes care data visualisation and user-interaction through DEMETER Dashboard component, in order to provide and guarantee the right visualisation of data. These processes will support the definition of end-user functionality related to the User Interface (UI) navigation, in the understanding and discovery of in-depth data from heterogeneous sources, trying to unify and aggregate the data as much as possible, in order to deliver to each involved Stakeholder in the DEMETER System the output in line with their objectives.

12 Interfacing between main architecture components

The main objective of WP3 is to design, develop, integrate, and then deploy and test the DEMETER platform. This is meant to be a functional and modular platform which will support initially all the cross-border DEMETER pilots (in the Agrifood domain) and then other DEMETER-enabled applications. Consequently, it will provide an end-to-end solution including open source components for gathering data from the DEMETER pilots through various communication protocols (e.g., LoRa, WiFi, zigbee etc). This platform will have to cover both data-in-motion and data-at-rest and implement an open API to interoperate with existing legacy systems and other proprietary platforms in the Agri-food domain and support all possible connectors to that world. This will allow farmers, and more specifically the farm management information systems (FMIS) that they currently use to have a vertical platform providing support through all the agricultural value chain, i.e., the product design, selling, production and delivery processes. To this end, the proposed platform will not only have to ensure interoperability at a low level, but also and, most importantly, between all the software components, by implementing standard communication interfaces and also facilitating understanding through the provision of suitable models and formats.

These aforementioned aspects inspire the design phase of the platform's components, and, more specifically, lead to the requirements for and the definition of interoperable interfaces with different protocols and operating methods.

In particular, the low level components or those that deal with the communication between the DEMETER platform and the Data Providers, should often be running at the device level, enabling communication with connected devices through different types of wireless technologies, such as ZigBee, Bluetooth or Wi-Fi; or alternatively they could be integrated in the DEMETER architecture, which in this case would enable the devices to establish connections through high level protocols and specific wrappers/translators in order to allow data uniformity and enable interoperability within the DEMETER platform and among its main application components.

Any system based on DEMETER will be able to collect, publish, exchange, process and analyse large data quickly and efficiently. Data from external providers and/or legacy systems (e.g., platforms, things, services, applications, devices) or from other DEMETER components must be managed in order to activate the necessary information translation and mapping components (wrappers). The DEMETER Core Enablers components, integrated in the DEMETER platform, will define a whole series of interfaces in order to manage data. The implementation will be based on services that are each consistent with a specific purpose. Therefore, these components will be connected to the other core modules and to the DEH and will be invoked based on the type of service or request demanded.

The DEMETER Platform will expose (standard) interfaces for data entry and recovery, enabling interoperability of data from producers to consumers. The consumers will not need to know where the data is located and what is the native protocol for their recovery. To achieve this level of interoperability, it will simply communicate through a well-defined interface that specifies the data it needs.

In view of all these requirements and design decisions, we now present (Figure 45) a high-level view which shows the main interactions between the components of the Architecture and the relationships between them (but is not complete with all the necessary interfaces). Possible improvements/revisions of the design that follows may result in the second release of this document expected in February 2021 (i.e., in deliverable D3.3).

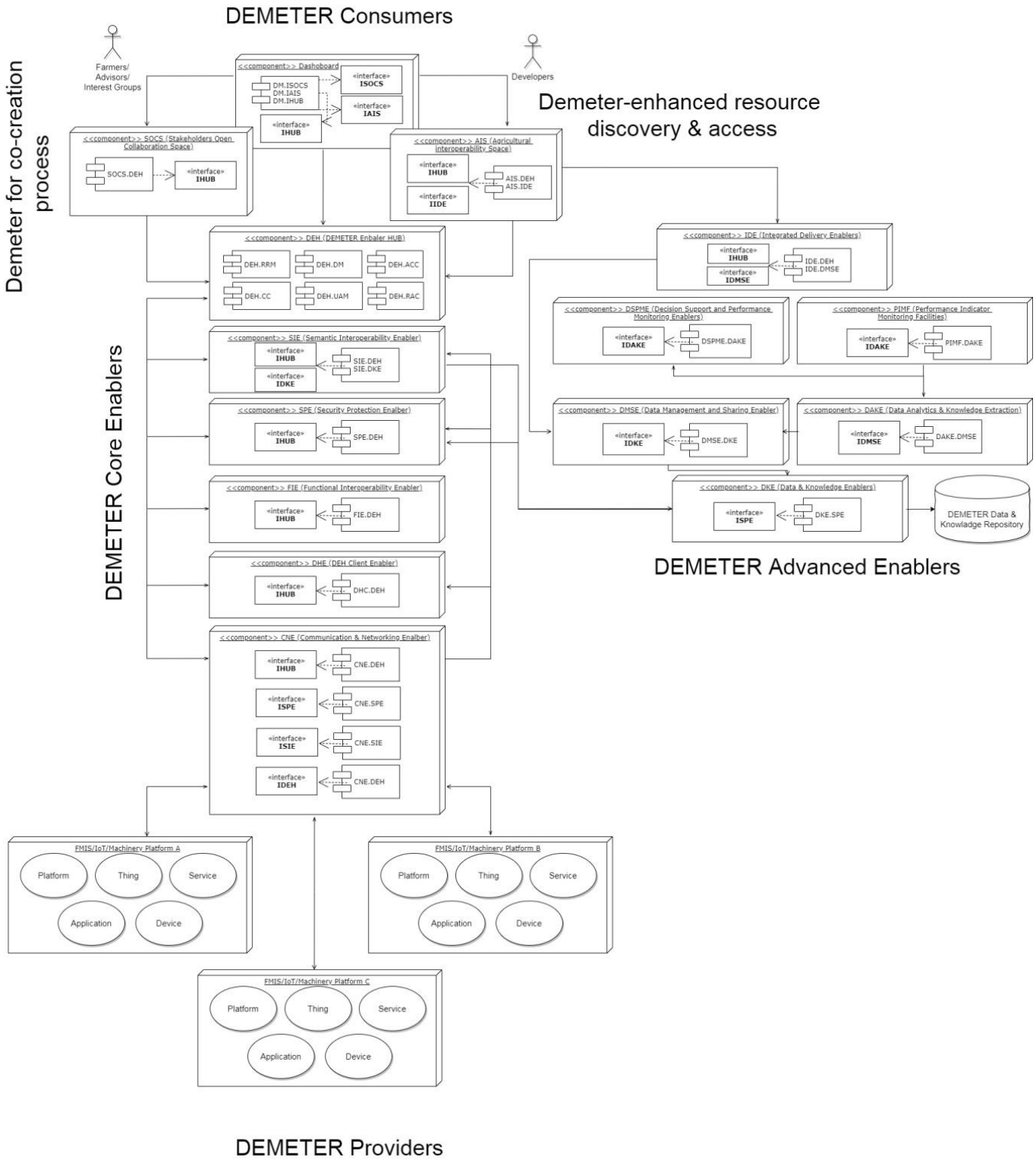


Figure 45. Main Interfaces between DEMETER's main component blocks

The component sitting at the highest level of the architecture is the Dashboard, which will implement interfaces to the main backend components of the DEMETER System: the SOCS, DEH and AIS. The implementations of these interfaces will allow a series of operations: these include the co-creation process through SOCS, the discovery of new DEMETER entities through DEH and the

possibility for developers to assemble new components through the AIS application. The Dashboard component also supports the display of data in user interfaces or web GUIs, enabling a whole range of standard data formats in web-based applications. The interfaces of the Dashboard component will have the dual task of both conforming to the formats of the backend services, but also of providing the necessary data to the web interfaces in formats useful for web interoperability. Furthermore, each component will have specific web user interfaces that will be used to perform the activity of its competence, meaning it will execute the activity it is tasked to perform as per the following description:

- The SOCS component will support the definition of paradigms and methodologies aimed at promoting the co-creation of resources, using specific online tools for this need and open cooperative innovation. The cornerstone of this approach will be the development of the Idea Management Platform, exploiting cloud-based knowledge management and consequently controlling the huge amount of data from the data Providers up to the business level of the increased collaboration itself. The platform will provide tools and services specifically designed for the innovation solutions developed during the project. The collaborative platform will also be built on open source technologies and exploiting the relevant results from research fields such as Co-Creation, Open Innovation and Collective Intelligence.
- The DEMETER Enabler Hub (DEH) will take charge, through its interfaces, the processes of the registration and discovery of DEMETER enabled entities, the management of user accounts and the authorizations regarding which entities can be viewed using the access control module; depending on the rights provided by the access control, the interfaces provided by the entity management module will produce the resources enabled for discovery (and viewing) by the account that requested them. Finally, the accounting module interfaces will be used to provide all appropriate compensation (and accounting) services for the using the entity.
- The AIS component will support a whole series of interfaces that allow DEMETER developers to build and put together new DEMETER-enhanced entities starting from the enablers provided by DEMETER (both the DEMETER core and the advanced enablers). To accomplish this task, it will need to interface with the DEH, in order to discover the resources already present in DEMETER, the access rights and what is available from the services of the DEH. Finally, it will produce interfaces capable of managing the delivery of applications, their life cycle (from development to production) and their instantiation within the context of DEMETER-enabled applications.

The medium part of the architecture presented in Figure 45 will be populated by the core and the advanced Enablers, which in turn communicate to the lowest level (the devices, things, platforms) and with each other through the facilities provided by the Communications and Networking Enabler. Now, these enabler components, in most cases at least, should interface with the DEH to be aligned with the resources present in DEMETER, as well as to be informed regarding the access rights to vary components (e.g., data sources) and finally to allow the transactions between them to be always authenticated e.g., by indicating a user account qualified to perform operations in the DEMETER context. The Core Enablers will therefore provide common and mandatory components for any stakeholder interested in sharing their resources by creating a DEMETER entity registered with the hub, while the advanced ones can be considered as optional components offered to assist in the

creation of the DEMETER entities. It should be stressed that while the usage of the advanced enablers is optional, all of the entities created will be able and in fact required to use the facilities and services provided by the core enablers/components, such as the one that defines the rules on security at all levels, or the semantic interoperability between DEMETER entities, as well as the communication and networking enabler which enables the interactions.

13 Architecture instantiations for the DEMETER pilots

This section describes the instantiation diagram of the DEMETER Reference Architecture presented in section 6 for each of the 20 pilots of DEMETER, illustrating the stakeholders, technologies, solutions and DEMETER enablers/tools used by each pilot. As many of the components of each instantiation are used by all pilots (i.e., mandatory, core enablers offered by DEMETER), it has been decided not to list them explicitly in the diagrams. These components are represented in the diagrams as the green ovals of the Agricultural Interoperability Space layer and will not be explicitly listed, unless there is a very specific requirement to be addressed (e.g., support for a given data encryption algorithm).

For each of the architecture instantiations presented in this section the specific RA elements used by the pilots are identified and recorded for all element categories below:

1. Pilot stakeholders/application end users;
2. Pilot applications;
3. Pilot-specific or optional DEMETER enablers that are required for the development of the pilot application (included in the diagrams as the blue oval shapes);
4. All existing platforms/systems/services to be engaged in the pilot. In case standardized data-models/semantics are used it should be recorded as a comment at the bottom of the diagram. If nothing is mentioned, it is assumed that vendor-specific data formats are used;
5. External (can be public or open) platforms/repositories used by the apps;
6. All networking/communication protocols to be used in the pilot;
7. The farms to be engaged in the pilots;
8. All HW, equipment, devices, machinery, etc. to be used in the farms.

The following subsections present the architecture instances for each of the DEMETER pilots.

13.1 Pilot 1.1 & 1.2: Water Savings in Irrigated Crops & Smart Energy Management in Irrigated and Arable Crops

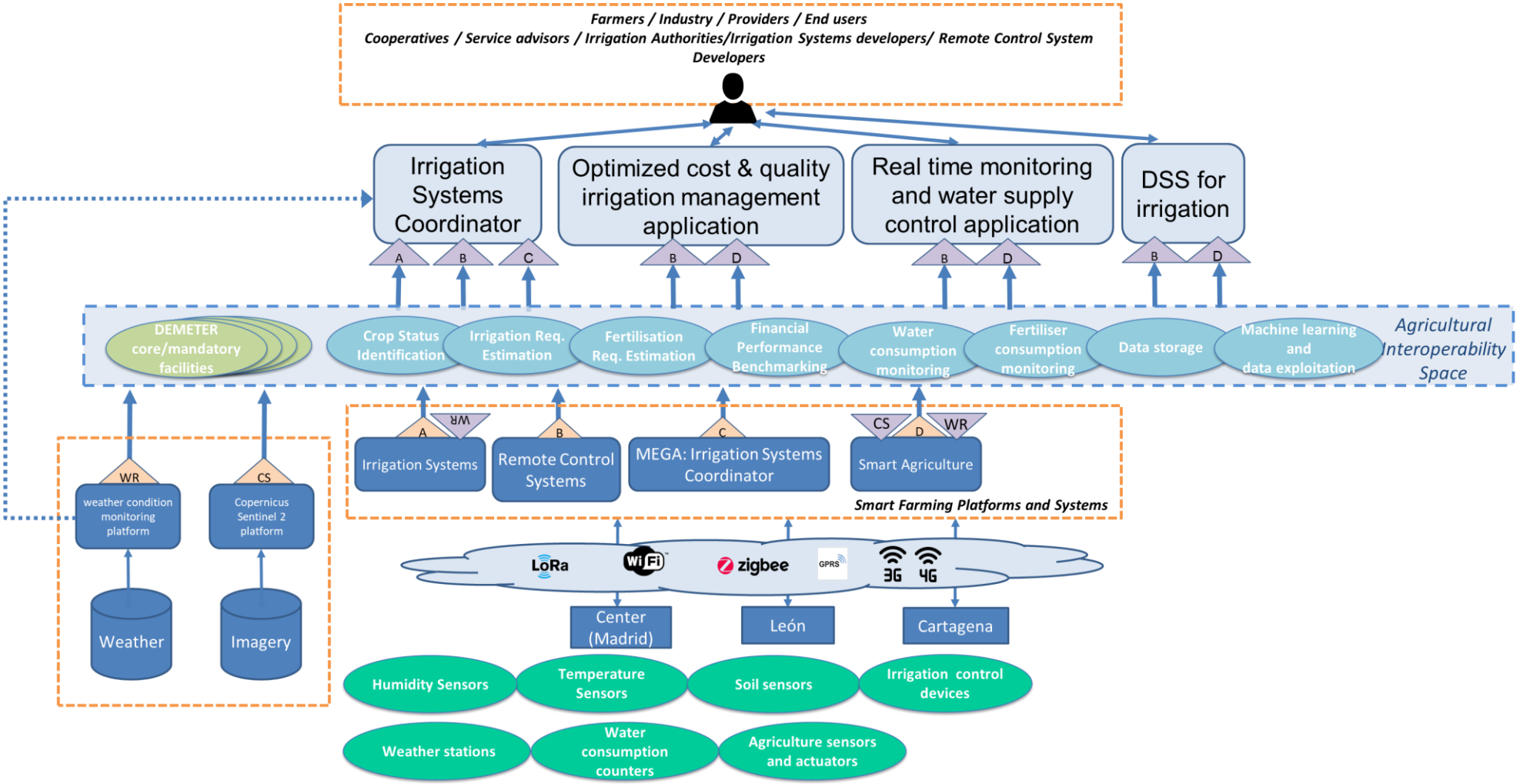


Figure 46. Pilots 1.1 & 1.2– DEMETER Reference Architecture instantiation

13.2 Pilot 1.3: Smart Irrigation Service in Rice & Maize Cultivation

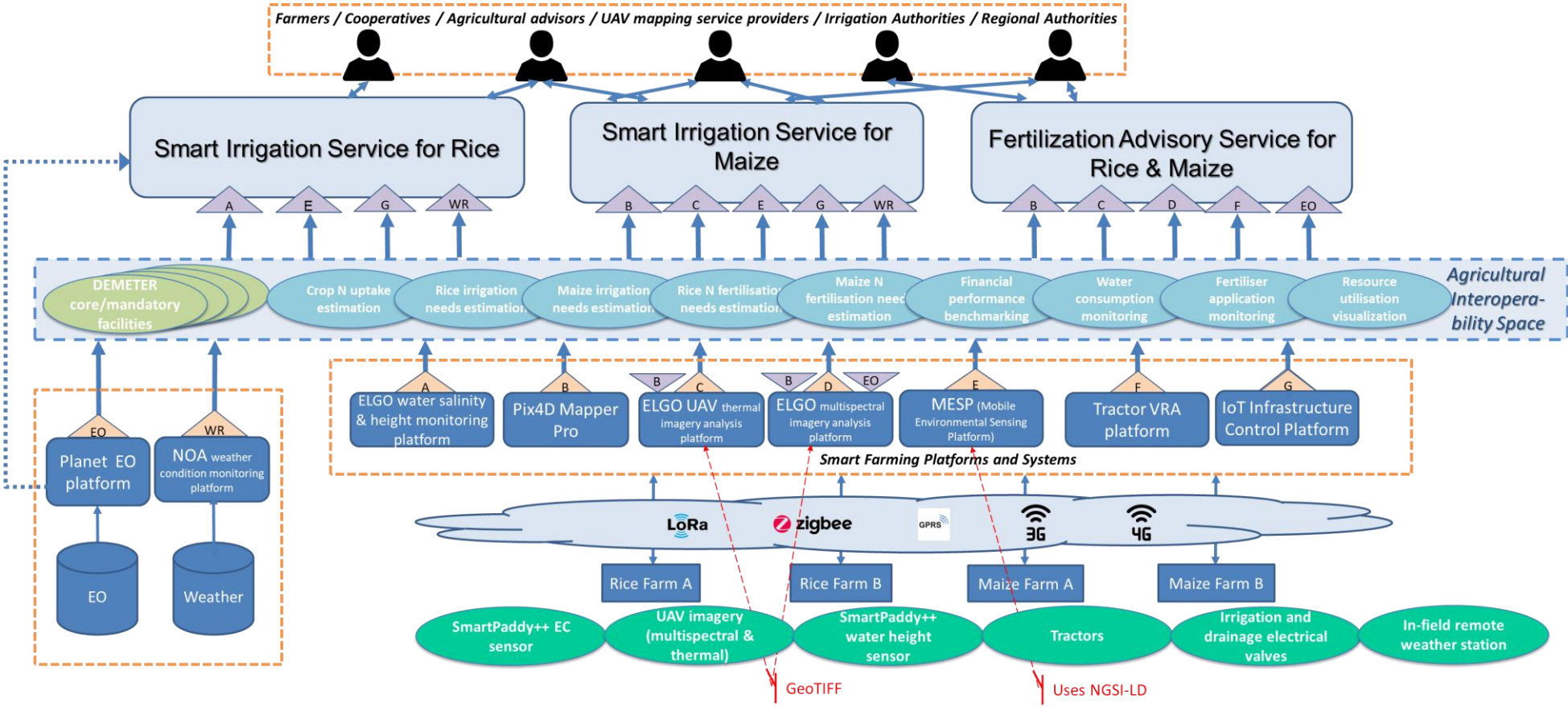


Figure 47. Pilot 1.3 – DEMETER Reference Architecture instantiation

13.3 Pilot 1.4: IoT Corn Management & Decision Support Platform

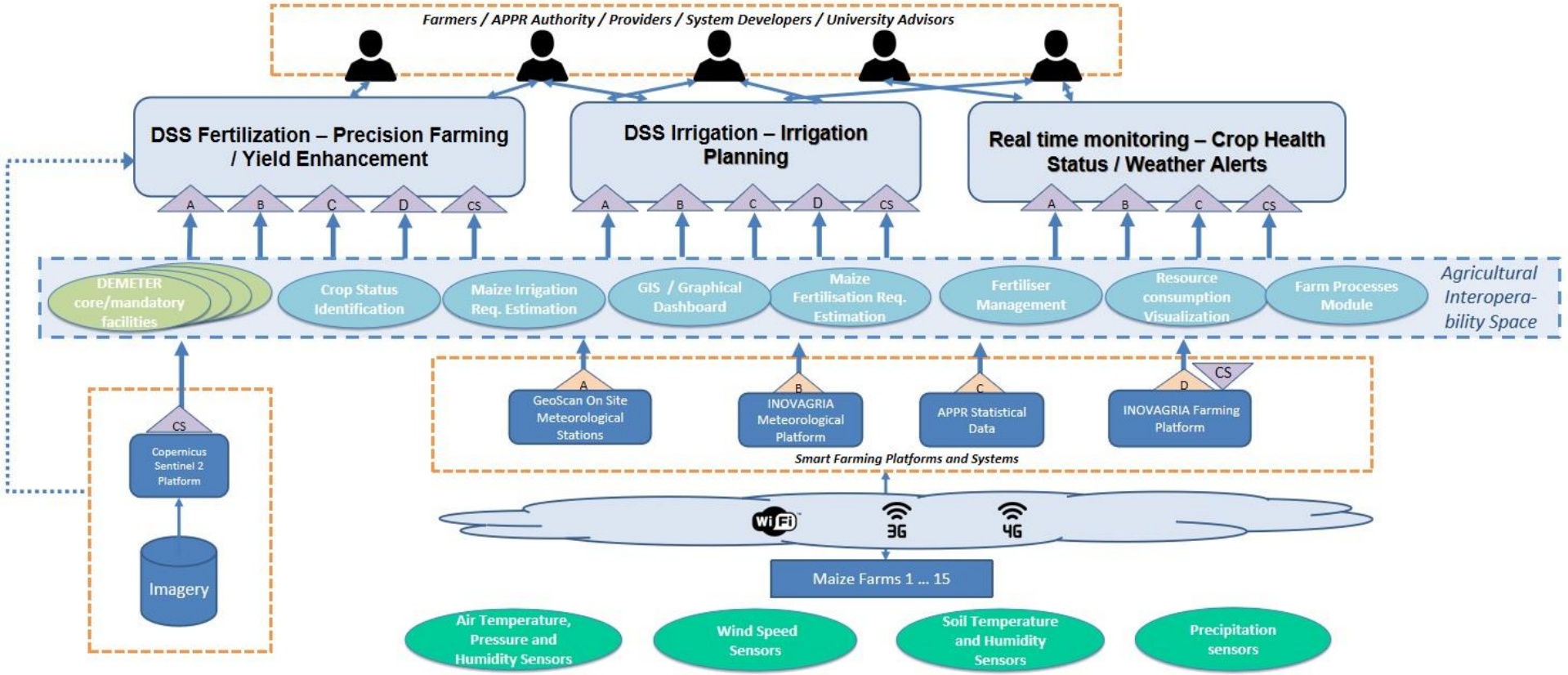


Figure 48. Pilot 1.4 – DEMETER Reference Architecture instantiation

13.4 Pilot 2.1: In-Service Condition Monitoring of Agricultural Machinery

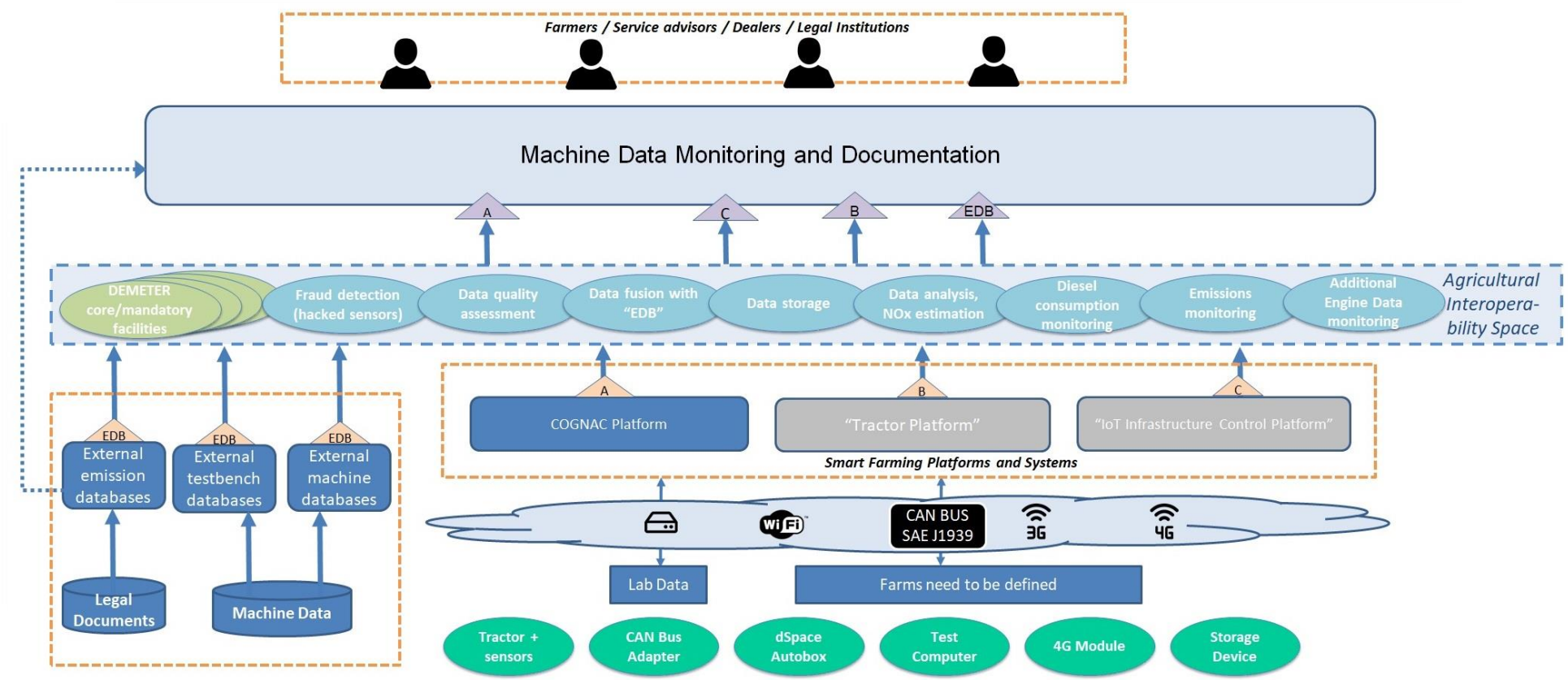


Figure 49. Pilot 2.1 – DEMETER Reference Architecture instantiation

13.5 Pilot 2.2: Automated Documentation of Arable Crop Farming Processes

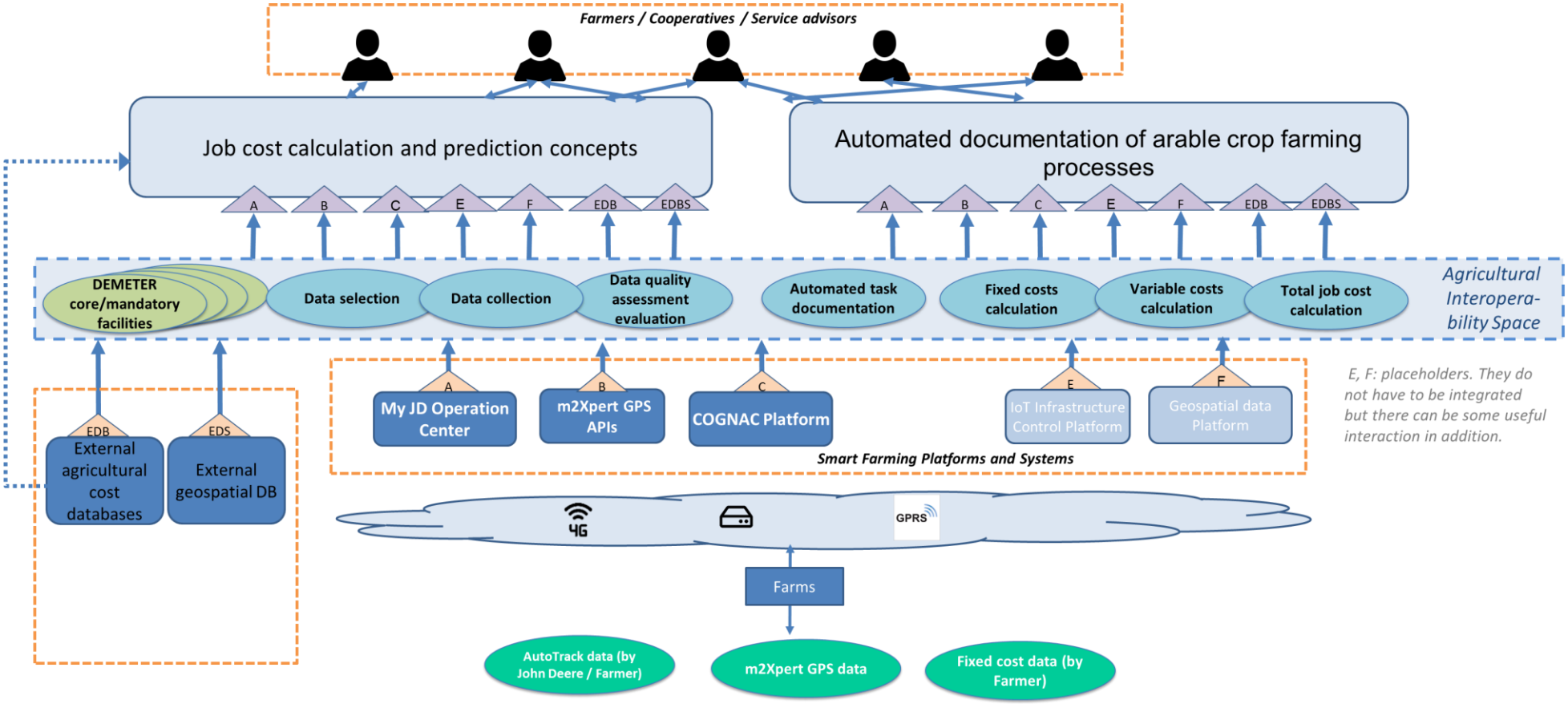


Figure 50. Pilot 2.2 – DEMETER Reference Architecture instantiation

13.6 Pilot 2.3: Data Brokerage Service and Decision Support System for Farm Management

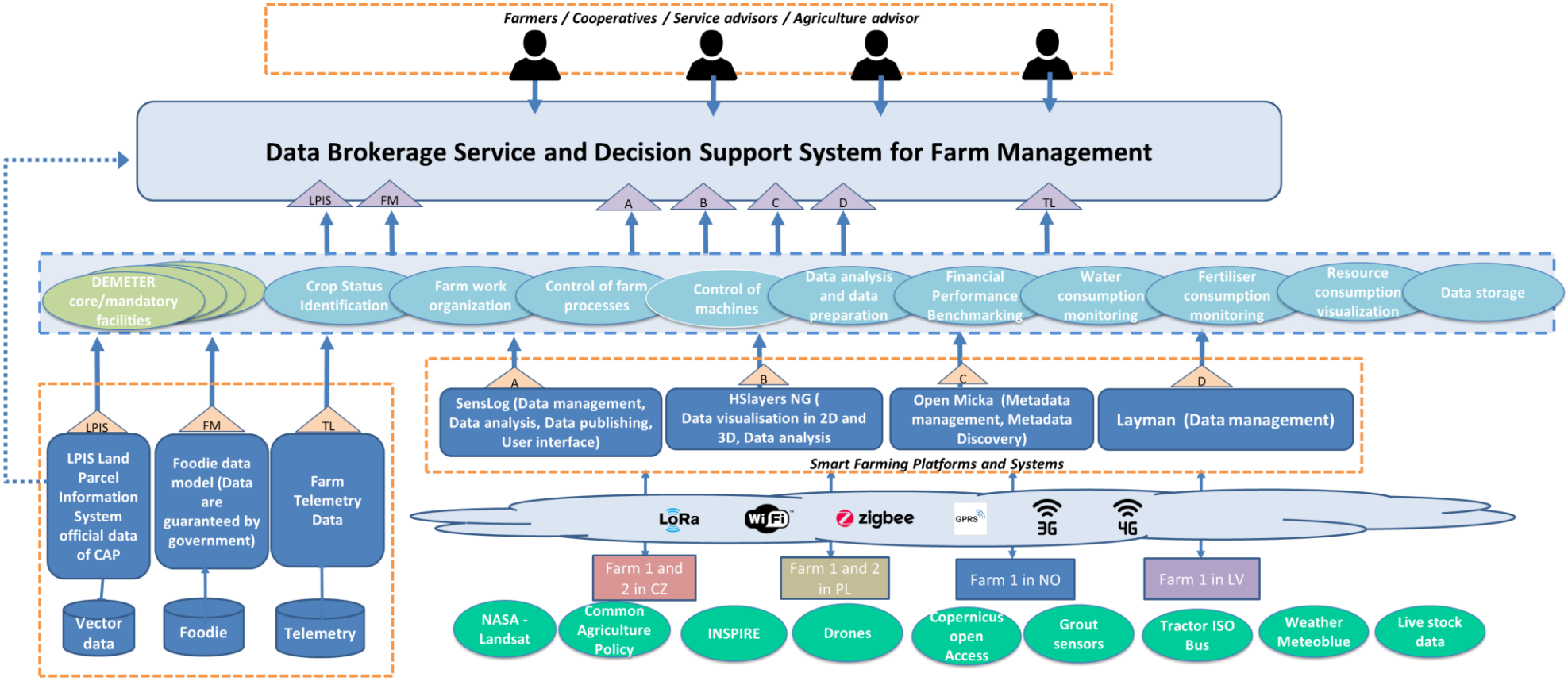


Figure 51. Pilot 2.3 – DEMETER Reference Architecture instantiation

13.7 Pilot 2.4: Benchmarking at Farm Level Decision Support System

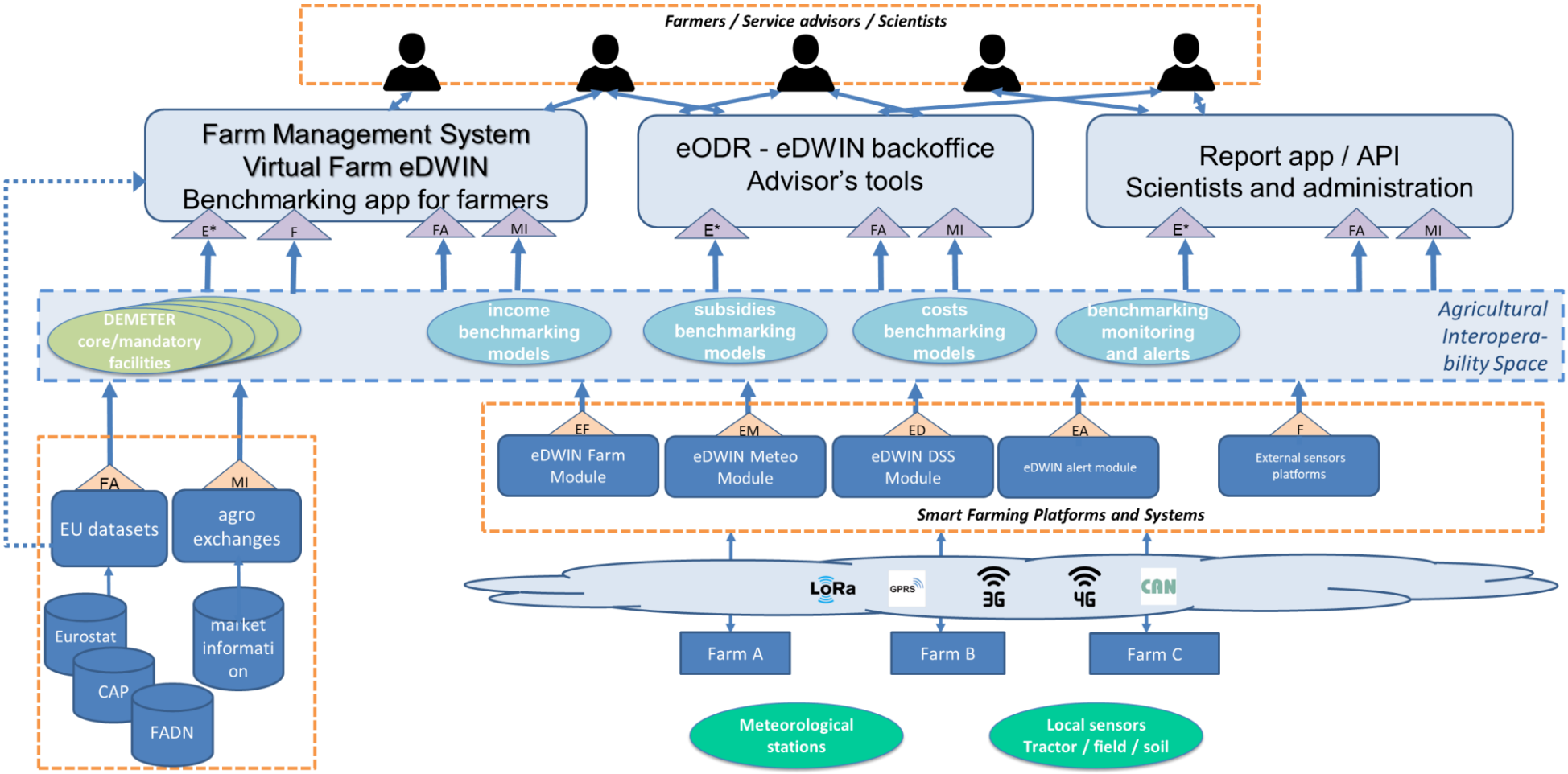


Figure 52. Pilot 2.4 – DEMETER Reference Architecture instantiation

13.8 Pilot 3.1: Decision Support System to Support Olive Growers

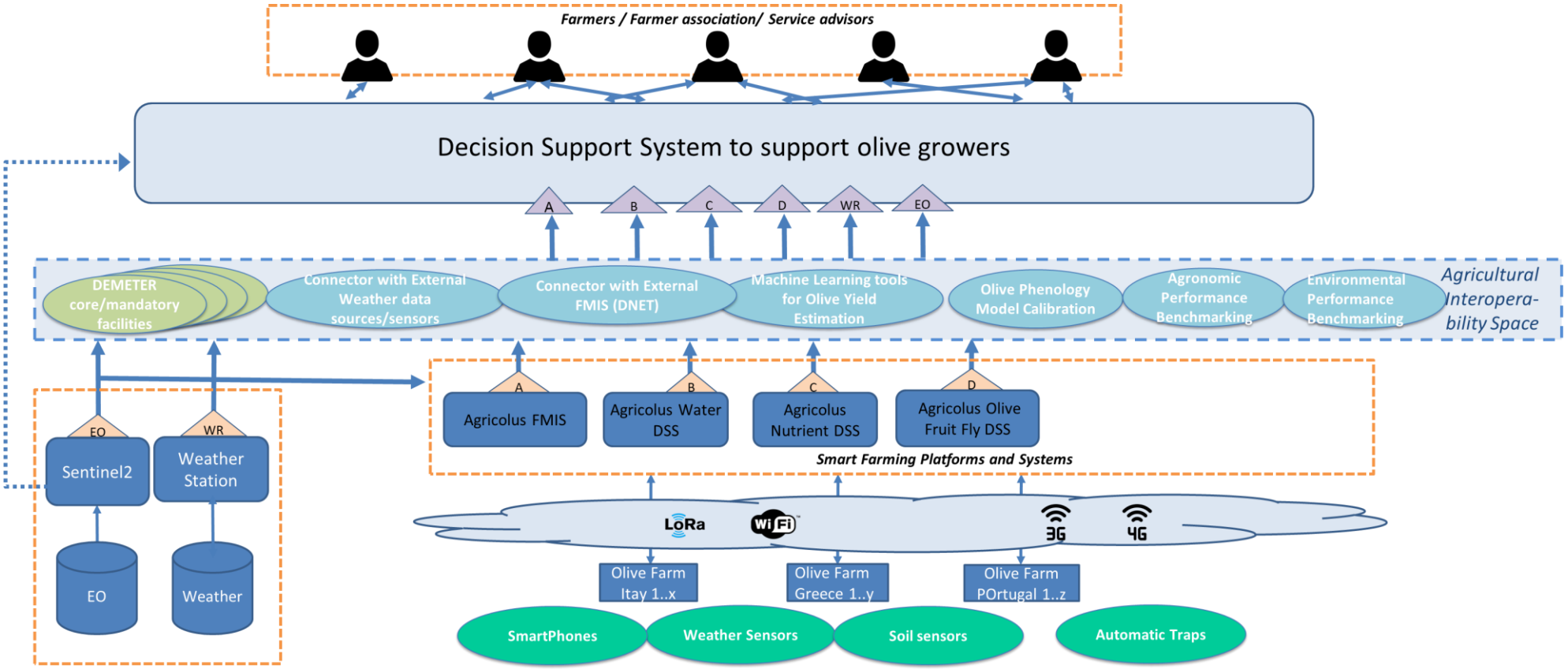


Figure 53. Pilot 3.1 – DEMETER Reference Architecture instantiation

13.9 Pilot 3.2: Precision Farming for Mediterranean Woody Crops

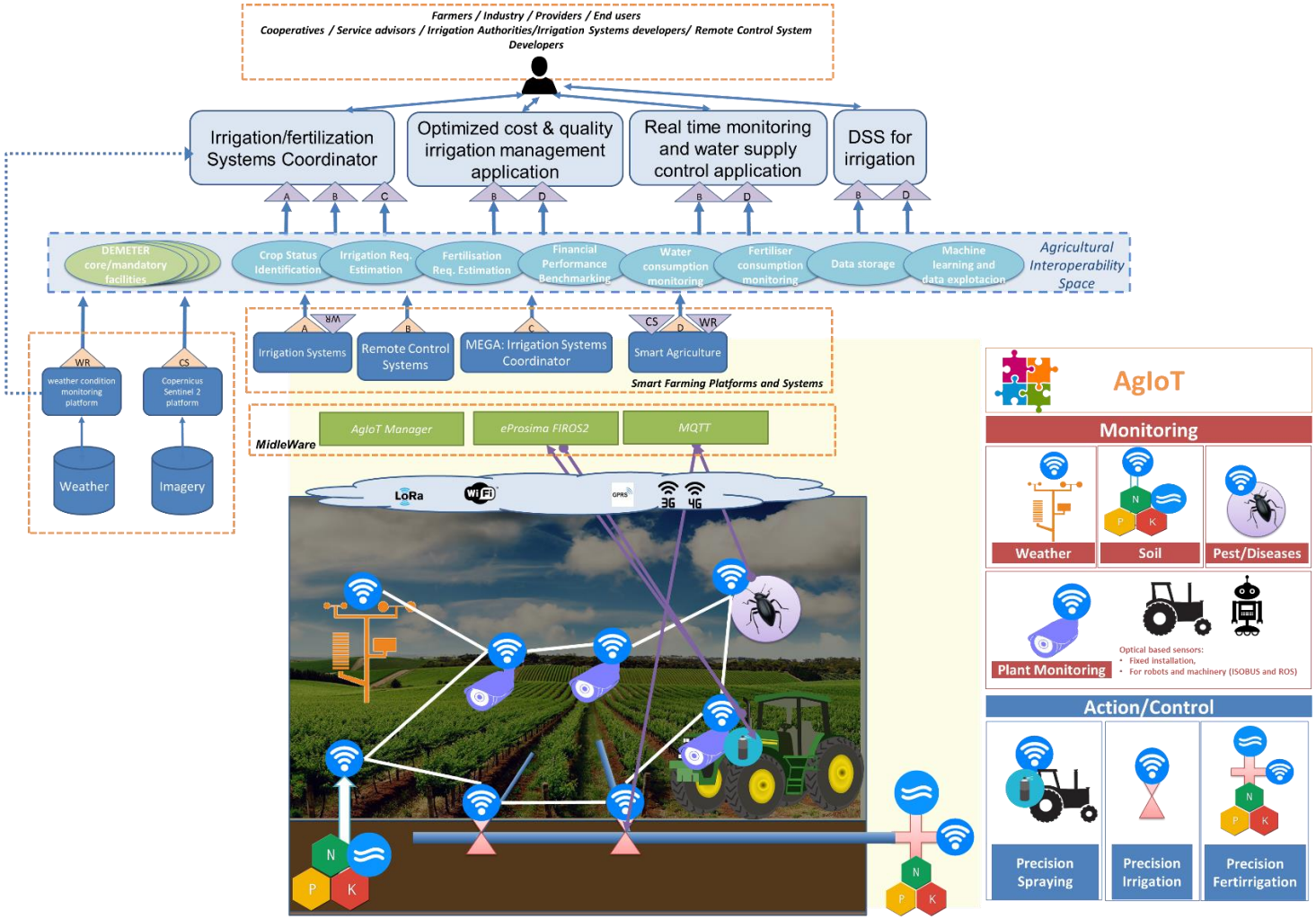
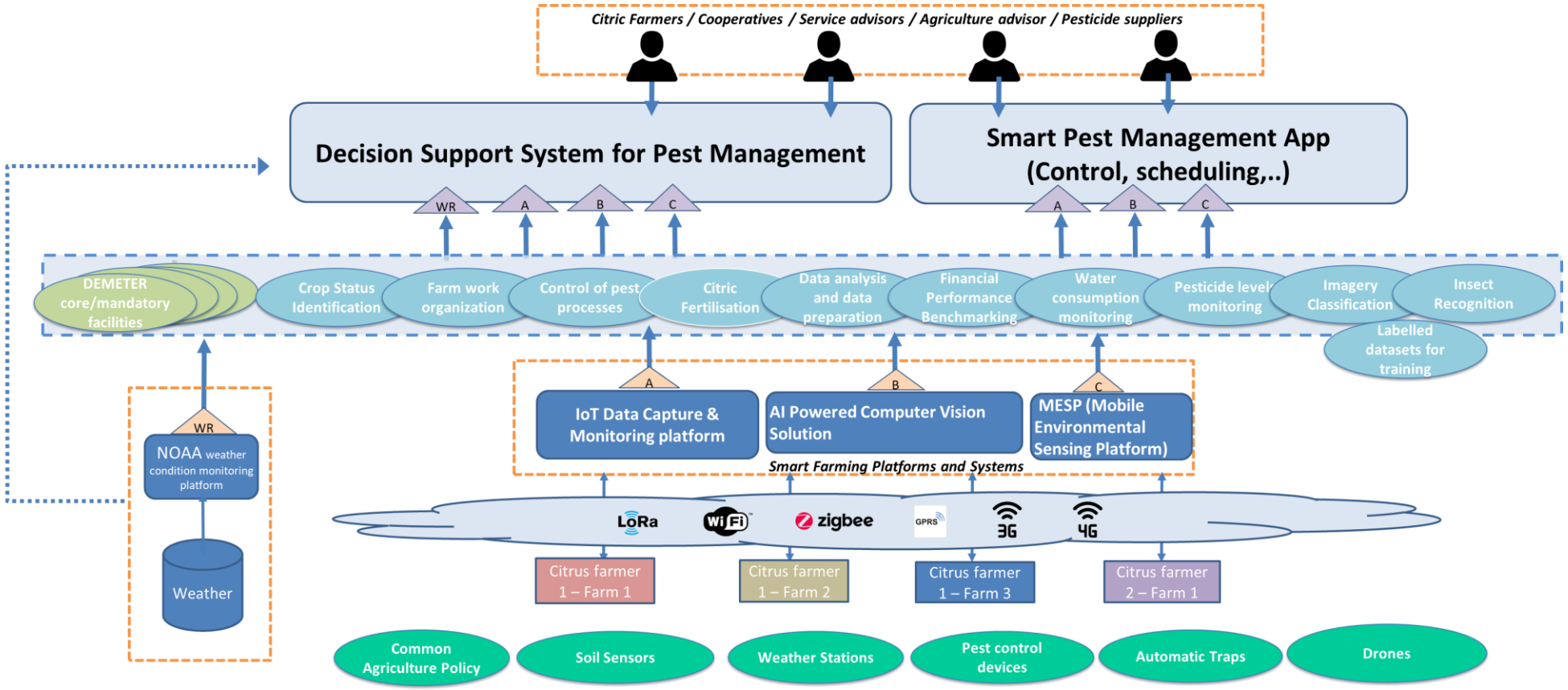


Figure 54. Pilot 3.2 – DEMETER Reference Architecture instantiation

13.10 Pilot 3.3: Pest Management Control on Fruit Fly



Possible Additional Capabilities:
- Combine information about Weather and number of flies captured
- Combine information about soil moisture and number of flies captured

Figure 55. Pilot 3.3 – DEMETER Reference Architecture instantiation

13.11 Pilot 3.4: Open Platform for Improved Crop Monitoring in Potato Farms

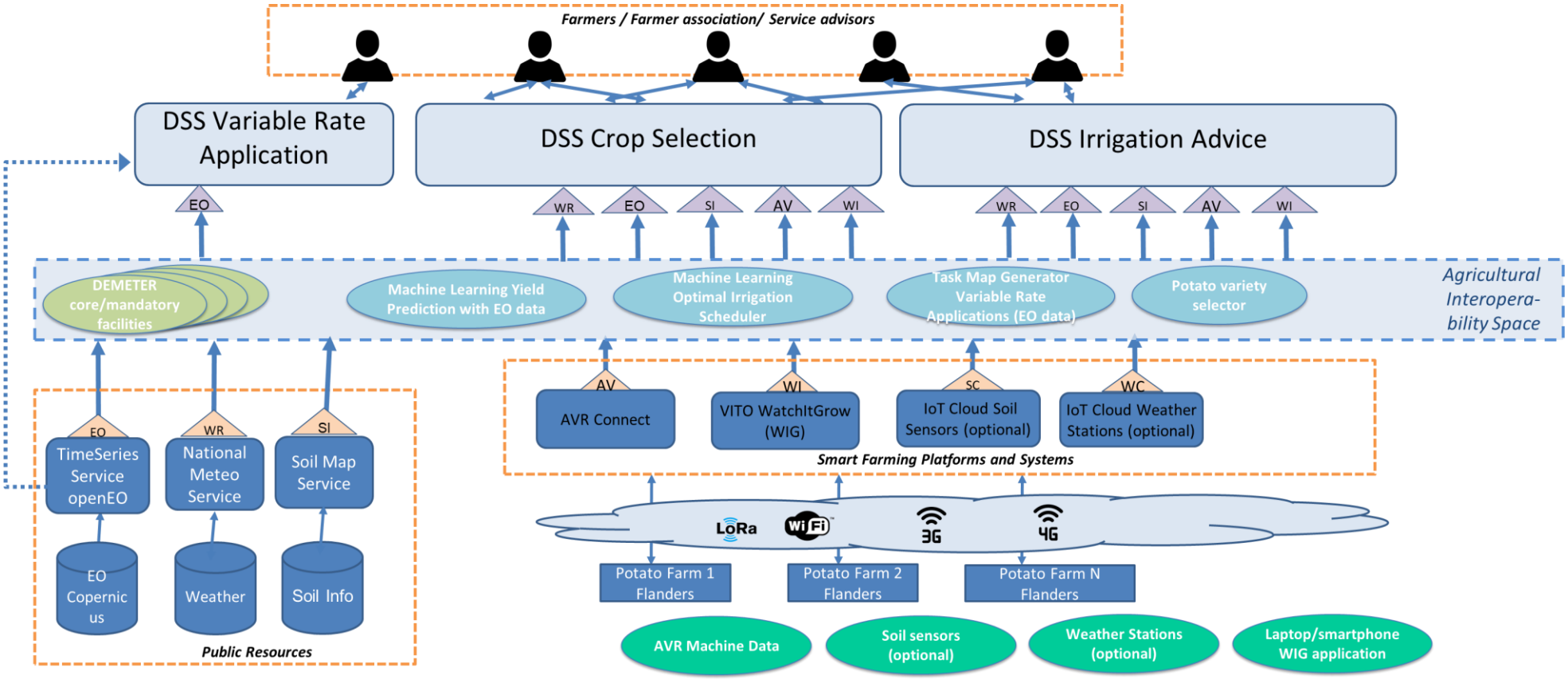


Figure 56. Pilot 3.4 – DEMETER Reference Architecture instantiation

13.12 Pilot 4.1: Dairy Farmers Dashboard for the Entire Milk and Meat Production Value Chain

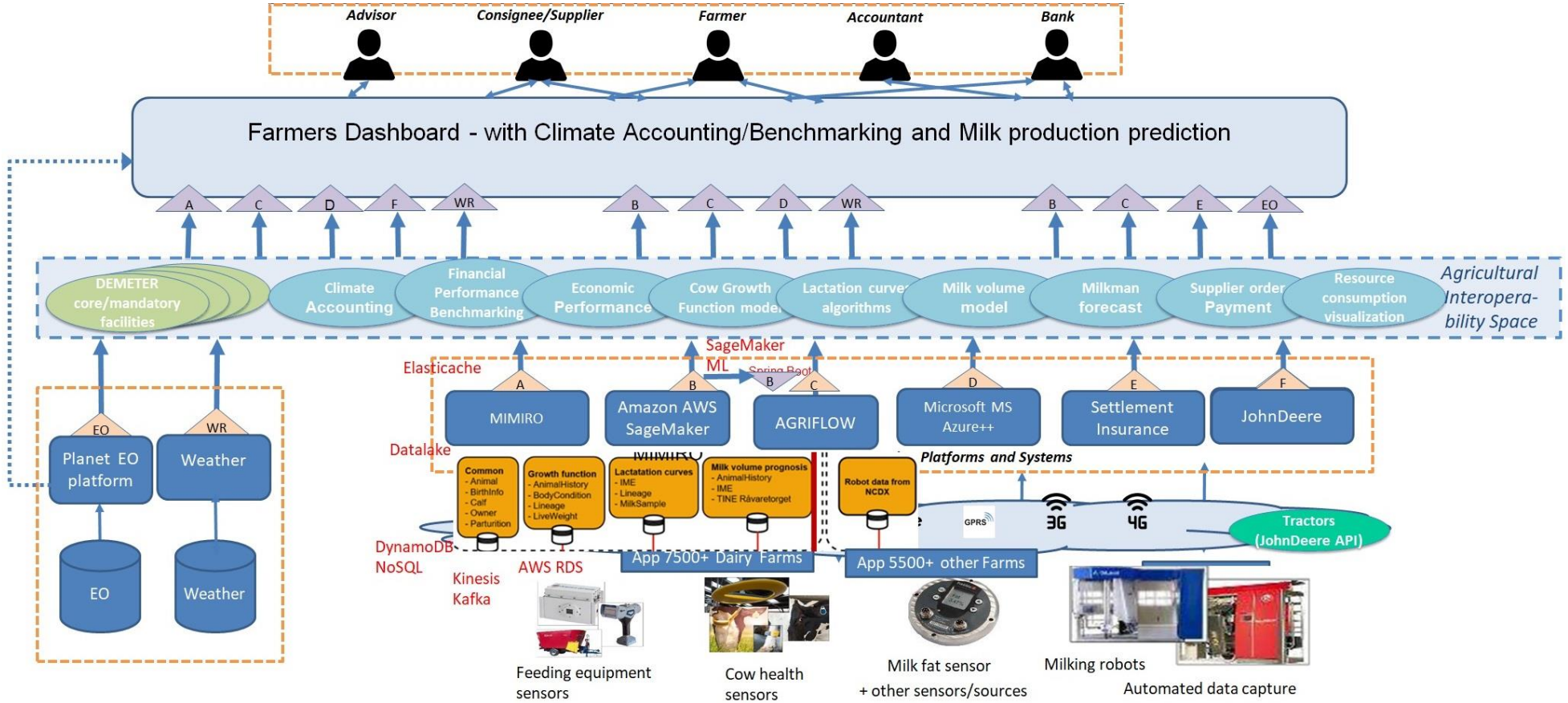


Figure 57. Pilot 4.1 – DEMETER Reference Architecture instantiation

13.13 Pilot 4.2: Consumer Awareness: Milk Quality and Animal Welfare Tracking

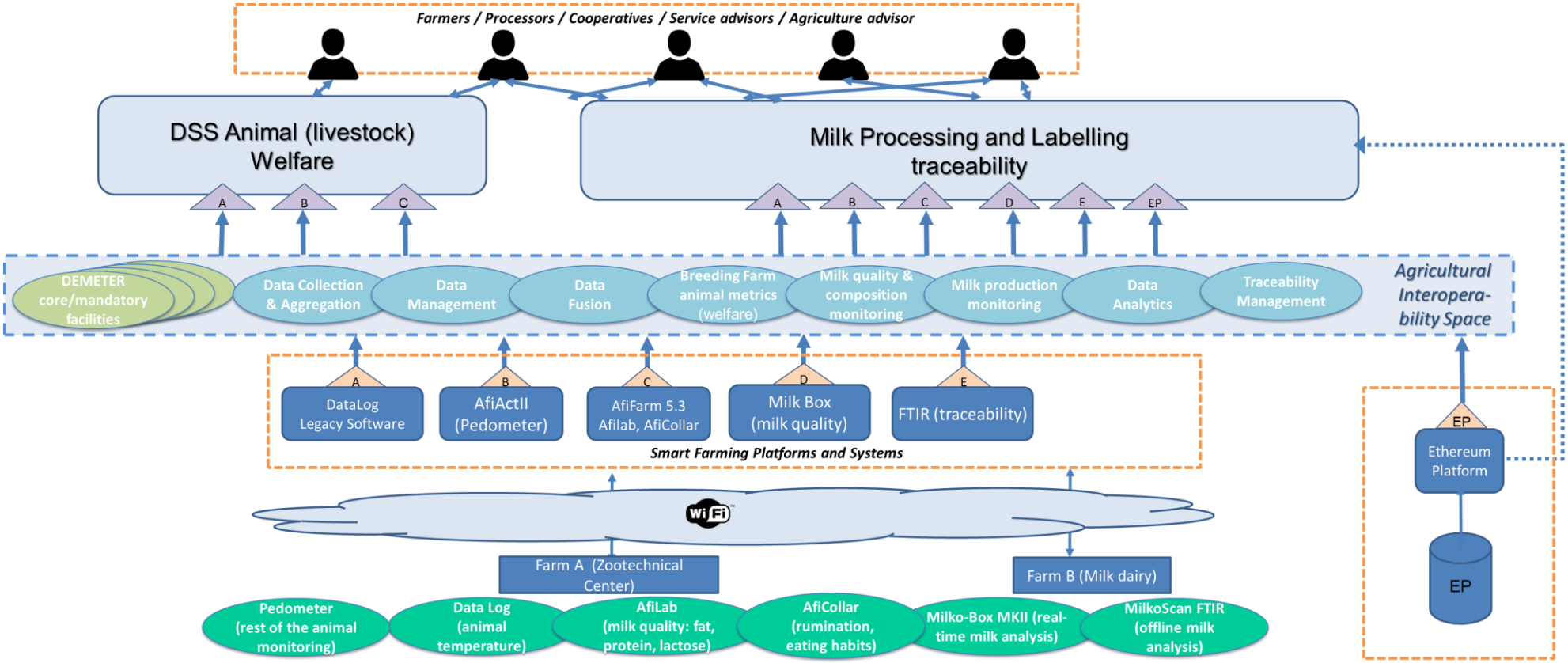


Figure 58. Pilot 4.2 – DEMETER Reference Architecture instantiation

13.14 Pilot 4.3: Proactive Milk Quality Control

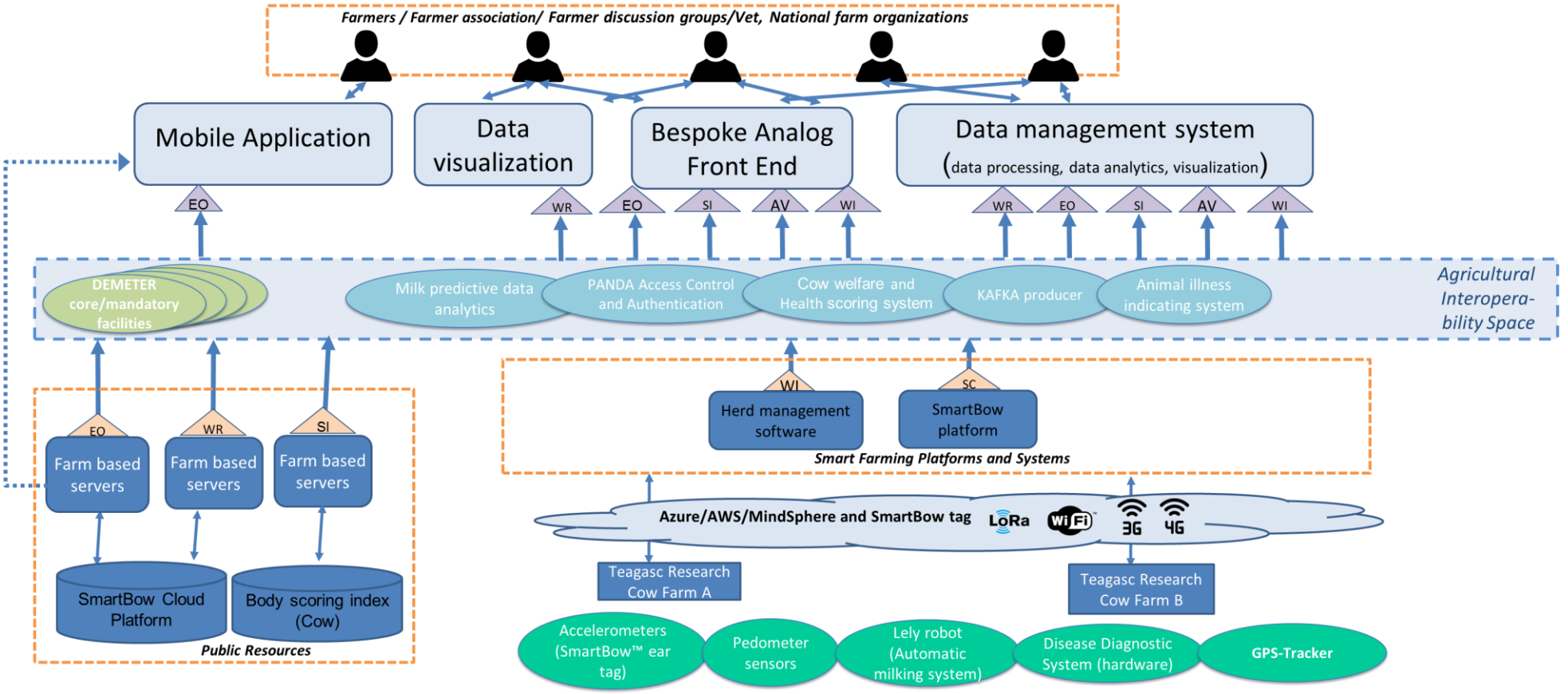


Figure 59. Pilot 4.3 – DEMETER Reference Architecture instantiation

13.15 Pilot 4.4: Optimal Chicken Farm Management

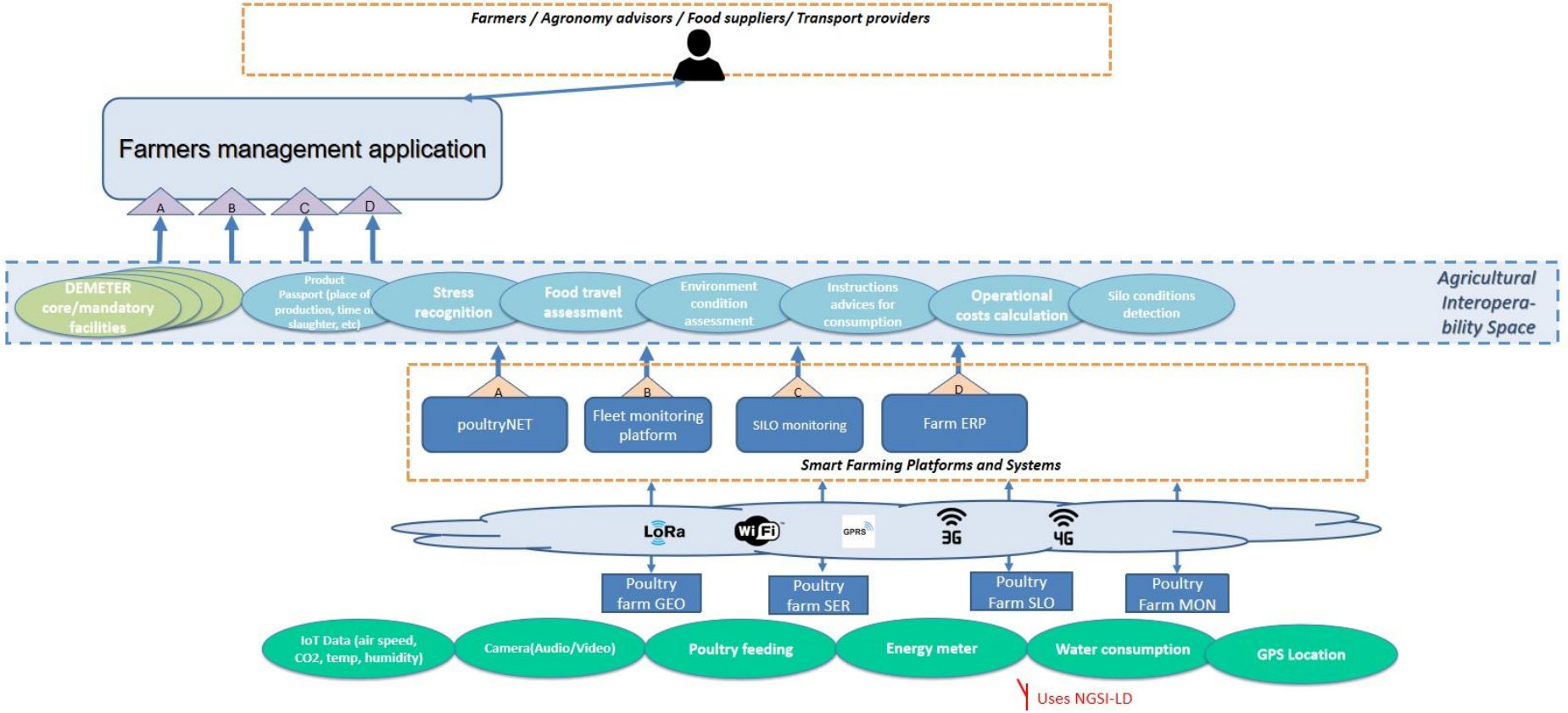


Figure 60. Pilot 4.4 – DEMETER Reference Architecture instantiation

13.16 Pilot 5.1: Disease Prediction and Supply Chain Transparency for Orchards/Vineyards

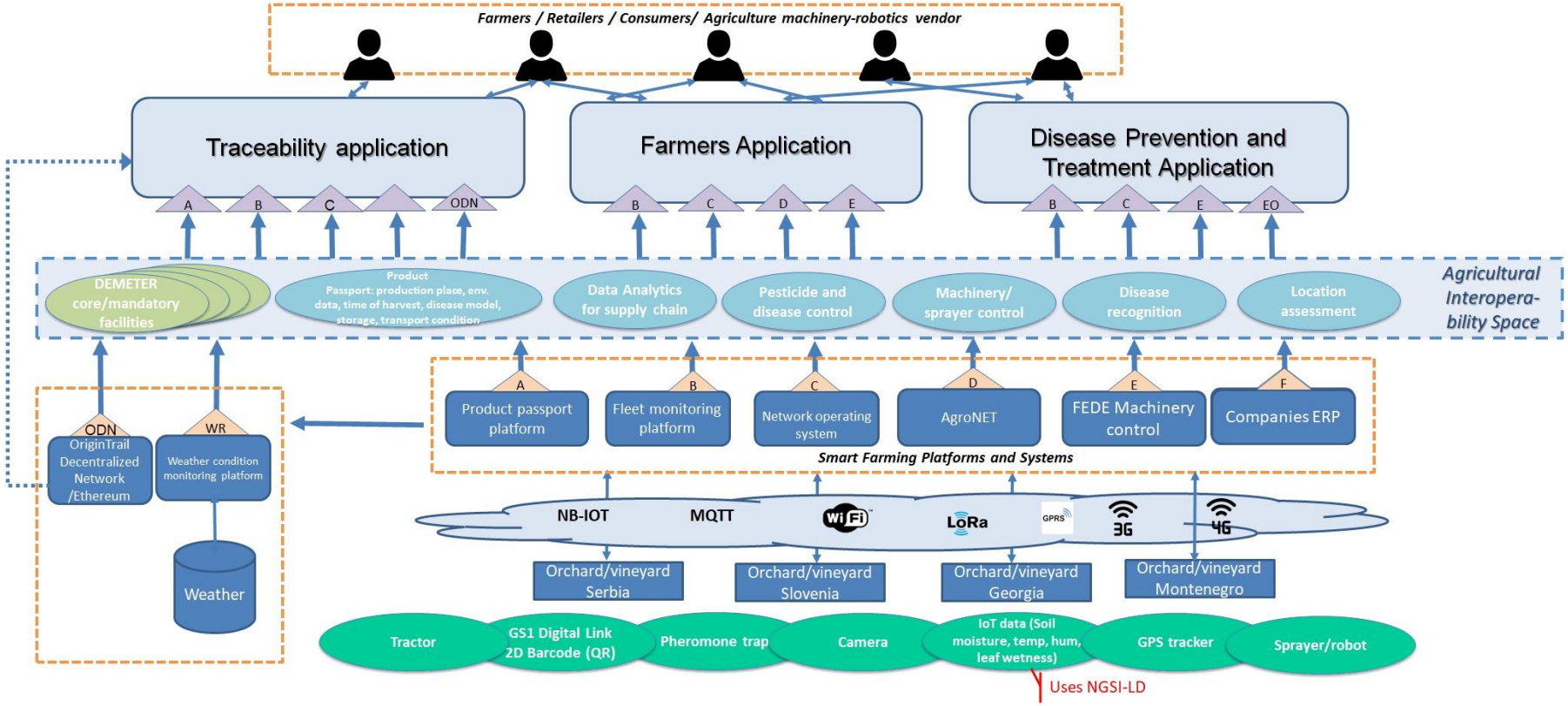


Figure 61. Pilot 5.1 – DEMETER Reference Architecture instantiation

13.17 Pilot 5.2: Farm of Things in Extensive Cattle Holdings

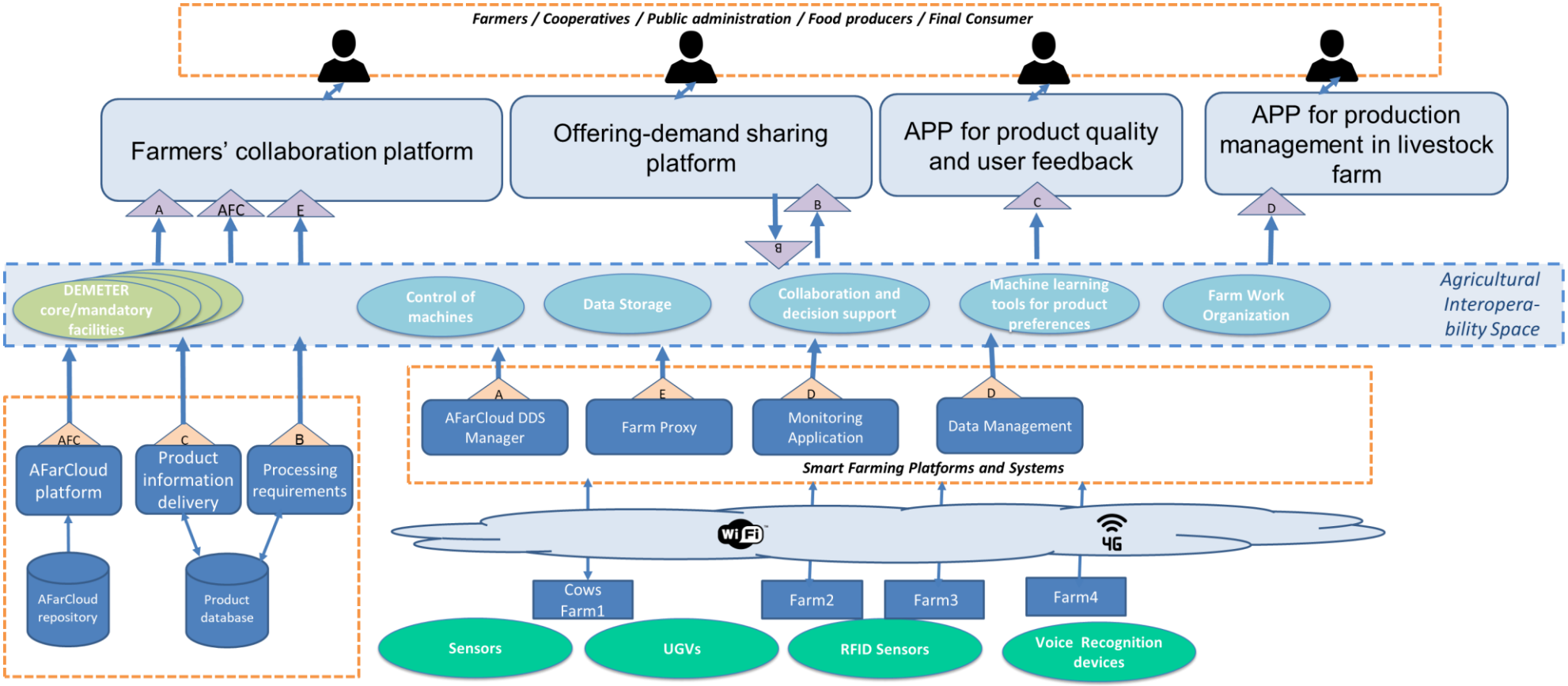


Figure 62. Pilot 5.2 – DEMETER Reference Architecture instantiation

13.18 Pilot 5.3: Pollination Optimisation in Apiculture

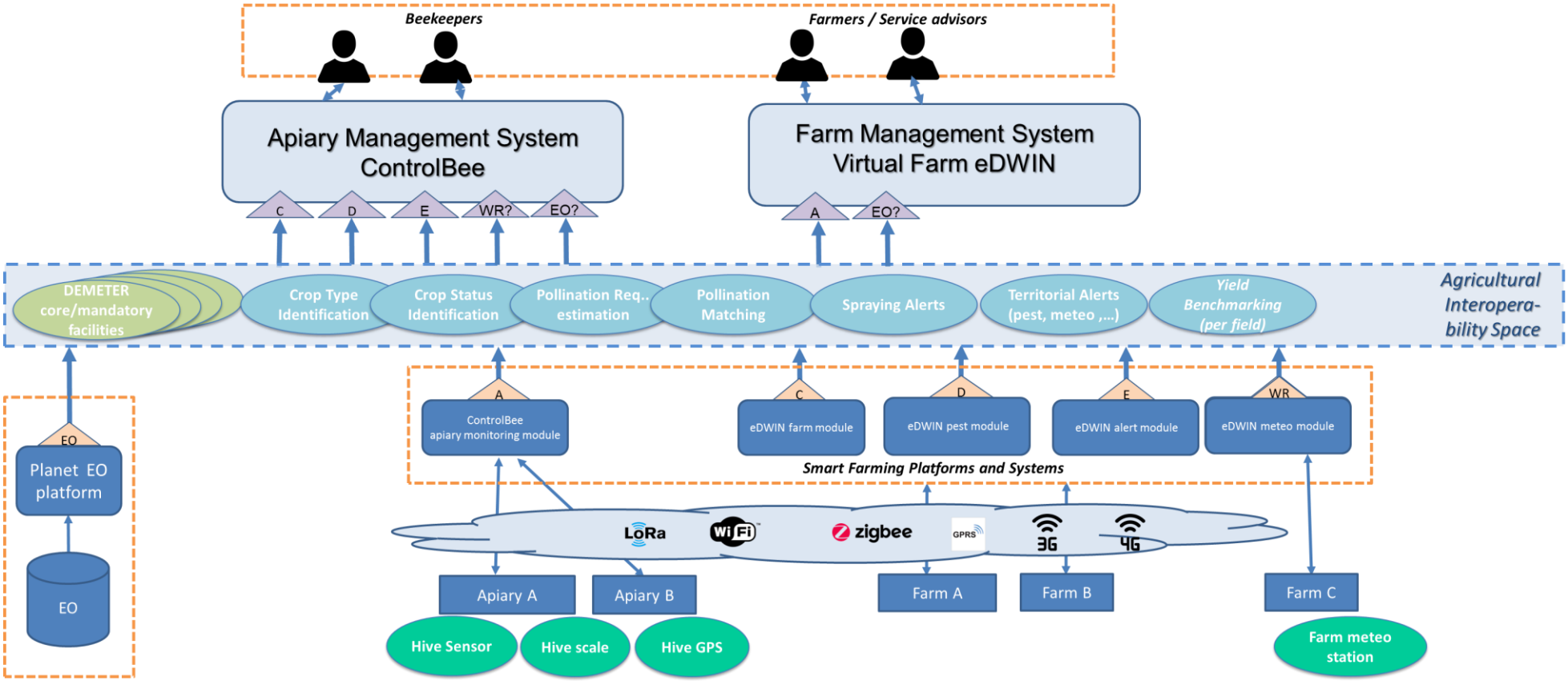


Figure 63. Pilot 5.3 – DEMETER Reference Architecture instantiation

13.19 Pilot 5.4: Transparent Supply Chain in Poultry Industry

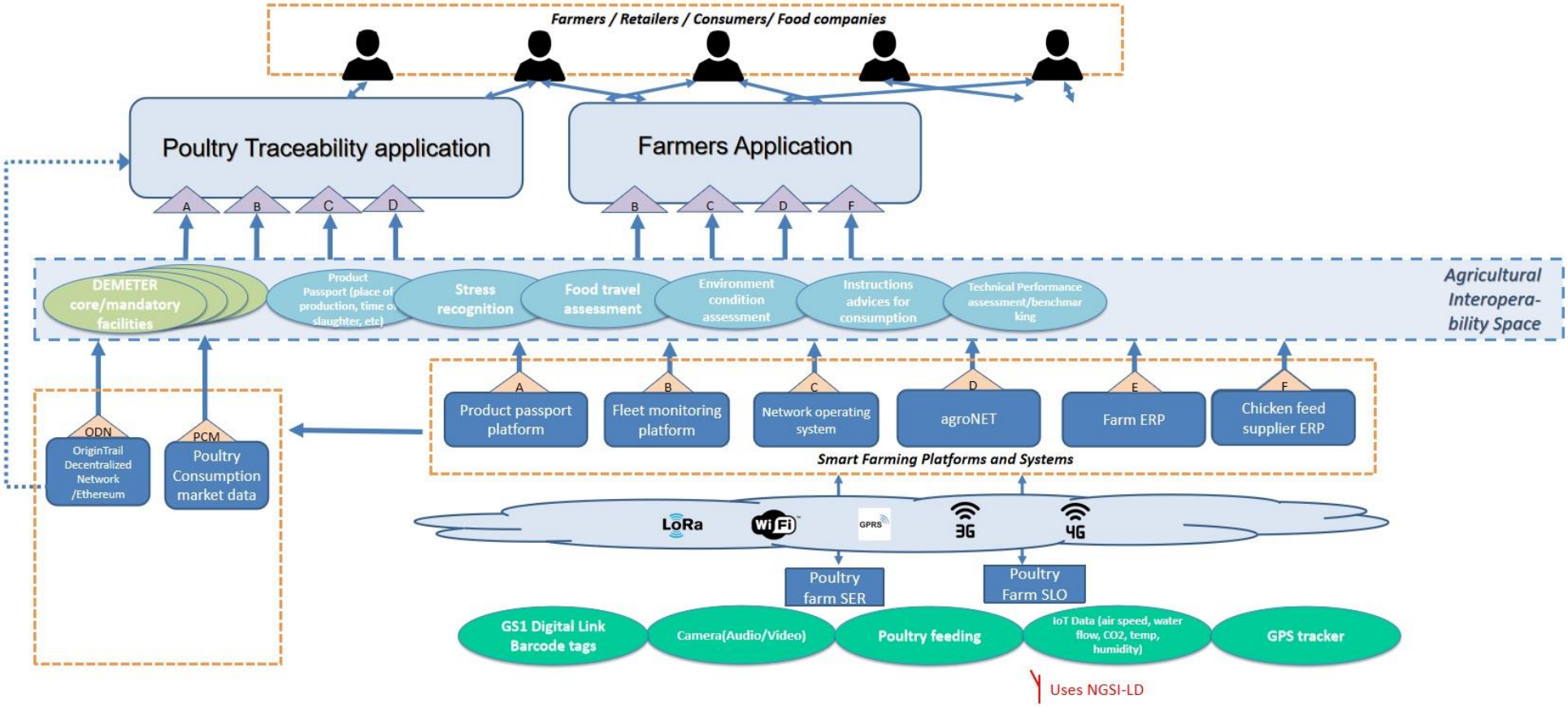


Figure 64. Pilot 5.4 – DEMETER Reference Architecture instantiation

14 GDPR considerations

14.1 GDPR measures overview

During the DEMETER project, we will engage with farmers from across Europe, deploying technology on their farms and working with them to evaluate the results. Thus, there exist several considerations that need to be taken regarding GDPR's technical aspect, such as data integrity and confidentiality, security, access control, traceability, ciphering algorithms and data provenance. Other GDPR management considerations (dataset review, GDPR guides, counsel, etc.) are out of scope. The following measures must be implemented to act in compliance:

- Data access control: the access control system must ensure that only authorized people or entities use the automated processing and decision-making system from the DEMETER Data & Knowledge Enablers and limit the access to personal data accordingly. This will be done by authorization and authentication mechanisms. The data analysis will be carried out in an anonymized way and the identification of individual farmers will not be possible, in accordance with the Directive 95/46/EC of the European Parliament on the protection of personal data. Personalized feedback will be provided to each farmer regarding data captured from their farms. For the IoT-based advisory services, the farmer can provide access to their own data to their advisor. At any given moment, the farmer will have the ability to restrict access to the data to any other individual or entity.
- Equipment access control: the access control system must deny unauthorized people or entities access to processing equipment, regardless of the specific implementation (in-house services or cloud services).
- Storage control: the unauthorized input of personal data and the unauthorized inspection, modification or deletion of stored personal data must be prevented by the Data Security & Governance DEMETER Advanced Enablers. To ensure an effective control of the data generated in DEMETER, it should be saved in an unalterable and traceable manner. The DEMETER Security Protection Enabler must implement enforceable user restrictions and log all data handling. Furthermore, any processing of personal data will be subject to appropriate technical and organizational security measures against unauthorized access and modifications, taking into account the nature, scope, and context. The recruitment of the farmers will be done on a voluntary basis (and a signed informed consent in the language of the participant will be required), while each farmer will have the ability to opt out of the measurements at any given moment. On request of the farmer, their data can also be removed from the database and omitted from any analysis.
- Communication control: ensure that it is possible to verify and establish the bodies to which personal data have been or may be transmitted or made available to. The sharing of data between DEMETER stakeholders falls under the scope of their specific legislation. The DEMETER Enablers will not be able to effectively prevent wrongful sharing of data. In this regard, the data export from the Enablers could be limited to specific users or entities as part of the access control system. The DEMETER Communication and Networking Enabler will ensure a secure communication layer between all entities.
- Input control: ensure that it is possible to verify and establish which personal data have been introduced in automated processing systems and when and by whom. DEMETER Enablers will ensure input control by logging of the usage of the tools.

- Transport Control: those people in charge of managing data must prevent the unauthorized reading, copying, modification or deletion of personal data during transfers of personal data. DEMETER will implement measures such as encryption (and lightweight for constrained devices) to avoid the unauthorized use of data.
- Recovery: Ensure that installed systems may, in the case of interruption, be restored. DEMETER Security Protection Enabler will be implemented in a way to backup stored data in an appropriate way.
- Reliability: The software must ensure that the functions of the system perform correctly and that the appearance of faults in the functions is reported. DEMETER Enablers will be subject to logging and traceability, thereby making the analysis of possible technical issues possible.
- Integrity: ensure that stored personal data cannot be corrupted by means of a malfunctioning of the system. The integrity of data stored by DEMETER Enablers can secured by the implementation backup routines and access control.

The above stated principles should be appropriately addressed in the development and operational use of Enablers. While some principles go beyond what is possible to guarantee in the development phase of the Enablers, the project aims to develop tools that support data controllers to act in accordance to these principles. That said, the misuse of data is a problem which cannot be prevented with enough certainty.

14.2 Other technical measures

To ensure the points laid out in the previous subsection, the following technical aspects should be considered:

14.2.1 Access control

Authentication ensures that an identity of a subject (user or smart object) is valid, i.e., that the subject is indeed who or what s/he/it 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 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. This will ensure that the data and equipment from DEMETER, along with transport and input control, will be controlled in terms of who can access what.

14.2.2 Traceability

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. In the case of DEMETER, it is crucial to count on it in the smart farm and agrifood 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 traceability services, traceability is crucial in DEMETER in relation to access control, to keep track of who enters what and when. All stakeholders will need to share information and analyses, and that requires saving a log for possible audit logs or security breaches handling.

Regarding DLTs (Digital Ledger Technologies), many approaches have been studied for traceability in agrifood 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.

14.2.3 Lightweight Cryptography

During the last years, the number of IoT devices has grown in a considerable manner. The different types of applications, in which these gadgets are employed, involve in many cases the choice of resource-constrained devices. Lightweight cryptography primitives have been proposed and used on these kinds of devices. Agencies and organizations working on technology innovations outline several techniques which can be used for lightweight cryptography and which could be useful in IoT devices. WSN, RFID or, generally speaking, sensor networks have limited numbers of gates available for security and are often highly constrained with the power drain on the device. Therefore, Elliptic Curve Cryptography is a matching solution with the device limitations. Elliptic Curves allow an approach to public key cryptography by offering a relatively short key size. These short keys are faster and requires less computing power and storage capability than other first-generation encryption public key algorithms. For example, a 256-bit ECC encryption key provides the same security level as a 3072-bit RSA encryption key. Although ECC algorithms are more complex and more difficult to implement than the traditional algorithms, they are more complex to break by a malicious 3rd Party. These features have made the Elliptic Curve Cryptography a very interesting solution for the embedded systems world.

Regarding key exchange, the ECDH and ECMQV schemes are valid implementations of key agreement protocols for elliptic curves; both are based on Diffie-Hellman method for securely exchanging cryptographic keys over a public and insecure channel. The basic idea of a key agreement protocol is to generate a shared secret value calculated by all parties, equal in both cases, where the only cryptographic component exposed are the public keys of the entities. The security of this type of protocols is implicit in the generation of the shared secret, since both influence the outcome; so, when properly done, these schemes preclude undesired third parties from forcing a key choice on the agreeing entities.

The ECDH protocol can be forced with a man-in-the-middle attack, so it is possible to use a combination of ECDH and ECDSA schemes or, alternatively, the ECMQV. The latter is an evolution of ECDH, in which a previous public key exchange is necessary. After performing the algorithm, the two

parties have the same shared secret value, as in ECDH protocol, but with the assurance of sharing the secret value only with the other entity.

The chosen cryptography type will take care of ensuring that the transport control security needs of DEMETER are met.

14.2.4 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 to 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.

Based on the ReliAble euRopean Identity EcoSystem (ARIES)[22] 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 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. These approaches will ensure that there is a data transfer and storage control along DEMETER Enablers, as laid out on the previous section.

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

[22] https://www.aries-project.eu/

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.

14.2.5 Privacy and Security By-Design Technologies

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

14.2.5.1 Identity Management (IdM)

For identity management, traditional solutions lack proper features to manage the privacy-preserving and the minimal disclosure in 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 so as protecting 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 have interfaces to enable the modification of entities information from system administrators. For IoT environment, the IdM system enables distributed and scalable deployment to achieve high-performance with vast number of devices and identity data. Moreover, the IdM system must support limited computing resources to allow its deployment in constrained IoT gateways.

14.2.5.2 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 link. 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 amount 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 certifies. Both SKC and PKC enables that a data producer can encrypt information to be shared only by a unique consumer.

The ubiquitous and distributed nature of IoT environment requires privacy group sharing techniques to allow that a data producer can encrypt information to be accessible by a group of consumers or unknown receivers.

As alternative to PKC certificates, Identity-Based Encryption (IBE) [28] was developed to enable the

data sharing with a group of consumers based on an identity string. In that sense, Attribute-Based Encryption (ABE) [29] 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 Cyphertext-policy Attribute-based Encryption (CP-ABE) [30] 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 have not to control the dissemination of the encrypted information to consumers, while consumer can access to the information according to credentials based on its 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).

15 Conclusions / Next Steps

This deliverable describes in detail the DEMETER Reference Architecture and all related concepts. It initially presents the basis for the architecture design methodology employed and then provides an overview of the complex Digital Platforms and Reference Architecture landscape that is currently in place, along with the identified considerations for platforms selection and usage in DEMETER. A thorough state of the art review is then provided, addressing the dominant IoT Reference Models, Big Data Frameworks, Interoperability Platforms, Sector-specific Reference Architectures and existing LSP Reference Architectures in Smart Agriculture, as well as the OPEN DEI CSA. Subsequently the DEMETER Architectural Framework and alignment process specifics are provided, followed by a high-level overview of the DEMETER technical requirements, along with an introduction of the main DEMETER concepts and the respective terminology. The core of this document is a detailed elaboration on the designed DEMETER Reference Architecture, where six architecture views are thoroughly presented (i.e., high-level view, functional view, process view, data view, deployment view and business view. Next, the high-level interfaces between the main DEMETER architecture blocks are discussed, while DEMETER is positioned in the smart agri ecosystem/landscape. Subsequently, instantiations of the DEMETER Reference Architecture for all 20 DEMETER pilots are presented. Finally, the main GDPR considerations that are in place are presented, while the document concludes with a summary of the content discussed and the future plans that are in place.

The content of this deliverable is the result of collaborative work of partners not only in Task 3.1 (that is responsible for this Task), but also in WPs 2, 3, 4 and 5. More specifically, Section 9 has been prepared based on input by the entire WPs 2, 3 and 4, while section 13 has been contributed by the pilot leaders in WP5.

This deliverable contributes to the achievement of Milestone 2 (DEMETER Enablers, Hub, Spaces and Applications Release 1) planned for June 2020.

As already mentioned, the Reference Architecture elaborated upon in this deliverable will be complemented by five more deliverables expected in the next four months that will carry more technical and implementation details, i.e.:

- D2.1 DEMETER data models and semantic interoperability mechanisms (April 2020)
- D2.2 DEMETER data and knowledge extraction tools (May 2020)
- D3.2 DEMETER technology integration tools (June 2020)
- D4.1 Decision Support, Benchmarking and Performance Indicator Monitoring Tools – Release 1 (May 2020)
- D4.2 Decision Enablers, Advisory Support Tools and DEMETER Stakeholder Open Collaboration Space (June 2020)

These deliverables will provide additional information on the specific models developed, on the interfaces in place, on the concrete technologies and existing solutions exploited, etc., which are not discussed herewith.

Finally, in a year from now, the revised version of the DEMETER Reference Architecture is planned for release that will be presented in D3.3 to be delivered in February 2021.

16 References

AFarCloud D2.2 "Architecture requirements and definition (v1)",. (2019). Retrieved from http://www.afarcloud.eu/wp-content/uploads/2019/09/D2.2-Architecture-Requirements-and-Definition-1.0.pdf

AIOTI. (2018). *AIOTI - High Level Architecture (HLA), Release 4.0, AIOTI WG03 – IoT Standardisation,.* Retrieved from https://aioti.eu/wp-content/uploads/2018/06/AIOTI-HLA-R4.0.7.1-Final.pdf

Big Data (Wikipedia). (n.d.). Retrieved from https://en.wikipedia.org/wiki/Big_data

Carrez, F. (2013). *IoT-A. D1.5 – Final architectural reference model for the IoT v3.0.*

Data-driven Artificial Intelligence For European Economic Competitiveness and Societal Progress. BDVA Position Statement. (2018, November). Retrieved from http://www.bdva.eu/sites/default/files/AI-Position-Statement-BDVA-Final-12112018.pdf

Dumitrache, I., Sacala, I. S., Moisescu, M. A., & Caramihai, S. I. (2017). A conceptual framework for modeling and design of Cyber-Physical Systems. *Studies in Informatics and Control 26, no. 3*, 325-334.

European BDVA Strategic Research and Innovation Agenda v4.0. (2017, October). Retrieved from http://www.bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf

FIWARE developers catalogue. (n.d.). Retrieved from https://www.fiware.org/developers/catalogue/

FIWARE developers page. (n.d.). Retrieved from https://www.fiware.org/developers/

IDS Reference Architecture Model, Industrial Data Space, Version 2.0. (n.d.). Retrieved from https://www.internationaldataspaces.org/en/publications/ids-ram2-0/

IEC 62264-3:2016: Enterprise-control system integration - Part 3: Activity models of manufacturing operations management. (2016, December). Retrieved from https://www.iso.org/standard/67480.html

Industrial Internet of Things Volume G4: Security Framework - IIC:PUB:G4:V 1.0:PB:20160926. (2016). Retrieved from https://www.iiconsortium.org/pdf/IIC_PUB_G4_V1.00_PB-3.pdf

Industry Specification Group (ISG) cross cutting Context Information Management (CIM). (n.d.). (ETSI) Retrieved from https://www.etsi.org/committee/1422-cim

IoF2020 Deliverable 3.2 "The IoF2020 Use Case Architectures and overview of the related IoT Systems". (n.d.). Retrieved from https://www.iof2020.eu/deliverables/d3.2-uc-architectures-v2-final-1.3.pdf

ISA95 Enterprise-Control System Integration. (n.d.). Retrieved from https://www.isa.org/isa95/

Joint Vision Paper for an AI Public Private Partnership (AI PPP). Brussels: BDVA –euRobotics. (2019). Retrieved from http://www.bdva.eu/sites/default/files/VISION%20AI-PPP%20euRobotics-BDVA-Final.pdf

NIST Special Publication 1500-6. NIST Big Data Interoperability Framework: Volume 6, Reference Architecture. (2015). Retrieved from https://bigdatawg.nist.gov/_uploadfiles/NIST.SP.1500-6.pdf

OpenFog Reference Architecturefor Fog Computing. (2017, February). Retrieved from https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf

Reference Architecture Model Industrie 4.0 (RAMI4.0) - DIN SPEC 91345:2016-04. (2016). Retrieved from https://www.din.de/en/wdc-beuth:din21:250940128

Smart Agrifood - FIWARE Foundation Open Source Platform. (n.d.). Retrieved from https://www.fiware.org/community/smart-agrifood/

The Industrial Internet of Things Volume G1: Reference Architecture Version 1.9. (2019, June 19). Retrieved from https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf